Network Capability Exposure in 5G Mobile Networks

Zoujiamin Chen

TU Delft

Network Capability Exposure in 5G Mobile Networks

Thesis report

by

Zoujiamin Chen

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on November 29, 2024 at 09:00

Thesis committee:

Chair:	Dr. Ir. Eric Smeitink
Supervisors:	Ir. Rogier Noldus
External examiner:	Dr. Ir. Qing Wang
Place:	Faculty of Electrical Engineering, Delft
Project Duration:	October 2023 - November 2024
Student number:	5514924

Faculty of Electrical Engineering · Delft University of Technology

Acknowledgements

This thesis marks the end of my studies at Delft University of Technology, for the Master of Science in Electrical Engineering in the specialization track of Wireless Communication and Sensing. The journey has been both challenging and rewarding, providing valuable opportunities for growth and learning. I would like to take this opportunity to express my gratitude to everyone who has supported and guided me throughout this journey.

First and foremost, I would like to thank my daily supervisor, Ir. Rogier Noldus, for his guidance throughout the thesis process. His valuable feedback, insightful suggestions, and dedication were very important in shaping the outcome of this work. I am also grateful to Dr. Ir. Eric Smeitink and Dr. Ir. Qing Wang for kindly agreeing to be part of my thesis committee.

Finally, I would like to express my appreciation to my family and friends for their encouragement and support throughout my studies. Their belief in me kept me motivated and helped me persevere through challenging times.

Thank you all for being part of this journey.

Abstract

Network capability exposure (NCE) in 5G allows service providers to make network functionalities—such as data, connectivity services, and traffic management—accessible to developers and enterprises through APIs. This is essential for creating programmable networks that support diverse and complex 5G use cases, including gaming, drones, smart manufacturing, and autonomous vehicles. By leveraging these APIs, developers can access advanced 5G capabilities to design innovative applications, while service providers and enterprises unlock new revenue streams. For instance, APIs can enable mobile devices to dynamically activate high-speed connectivity tiers for specific applications, showcasing the flexibility and potential of NCE in 5G.

This thesis is divided into two parts. The first part investigates NCE in 5G mobile networks, focusing on the architecture, functionalities, and applications of the Network Exposure Function (NEF). It examines the NEF's role in securely exposing network services, its integration within the 5G ecosystem, and its implementation. Furthermore, the thesis evaluates capability exposure across industry standards, including 3GPP, O-RAN, the Operator Platform, and the CAMARA Project. The second part explores two use cases: (1) augmented reality (AR)-enhanced communications, where additional network capability exposure can enrich voice calling with AR features, and (2) drone operations, emphasizing collision avoidance for Beyond Visual Line of Sight (BVLOS) scenarios. Based on these analyses, the thesis proposes new exposure capabilities, including call control capability exposure and a Collision Avoidance API, to address identified gaps.

Contents

Nomenclature

List of Abbreviations

3GPP Third Generation Partnership Project

5G Fifth Generation Mobile Networks

5G-A Fifth Generation-Advanced

5GC Fifth Generation Core Networks

A-BGF Access Border Gateway Function

AF Application Function

AMF Access and Mobility Management Function

API Application Programming Interface

AR Augmented Reality

AS Application Server

AUSF Authentication Server Function

CDN Content Delivery Network

CER Capabilities Exposure Role

CP Control Plane

CSCF Call Session Control Function

CSP Communication Service Provider

CU Central Unit

DC Data Channel

DU Distributed Unit

eMBB Enhanced Mobile Broadband

ETSI European Telecommunications Standards Institute

GMLC Gateway Mobile Location Center

gNB Next Generation Node B

GPSI Generic Public Subscription Identifier

GSMA Global System for Mobile Communications Association

HPLMN Home Public Land Mobile Network

HSS Home Subscriber Server

I-CSCF Interrogating Call Session Control Function

IETF Internet Engineering Task Force

IMS IP Multimedia Subsystem

IMSI International Mobile Subscriber Identity

IoT Internet of Things

IP Internet Protocol

ISP Internet Service Providers

ITU International Telecommunication Union

JCAS Joint Communication and Sensing

LAN Local Area Network

LCS Location Services

LDR Location Determining Request

LMF Location Management Function

LRR Long-Range Radar

LTE Long-Term Evolution

MAC Medium Access Control

MEC Multi access Edge Computing

mMTC Massive Machine Type Communications

mmWave millimeter-Wave

MO-LR Mobile Originated Location Request

MRF Media Resource Function

MT-LR Mobile Terminated Location Request

NaaS Network as a Service

NE Network Entity

Near-RT RIC Near-Real-Time RAN Intelligent Controller

NEF Network Exposure Function

NF Network Function

NFV Network Function Virtualization

NG-RAN Next Generation Radio Access Network

NI-LR Network Induced Location Request

Non-RT RIC Non-Real-Time RAN Intelligent Controller

NR New Radio

NRF Network Repository Function

NSSAI Network Slice Selection Assistance Information

NSSF Network Slice Selection Function

NWDAF Network Data Analytics Function

O-RAN Open Radio Access Network

OSI Open Systems Interconnection

P-CSCF Proxy Call Session Control Function

PCC Policy Control and Charging

PCEF Policy and Charging Enforcement Function

PCF Policy Control Function

PCRF Policy and Charging Rules Function

PDCP Packet Data Convergence Protocol

PDN Packet Data Network

PDU Protocol Data Unit

PFD Packet Flow Description

PFDF Packet Flow Description Function

PHY PHysical Interface

PLMN Public Land Mobile Network

QoE Quality of Experience

QoS Quality of Service

RAI RAN Analytics Information

RAN Radio Access Network

RIC RAN Intelligent Controller

RLC Radio Link Contro

RRC Radio Resource Control

RTP Real-time Transport Protocol

S-CSCF Serving Call Session Control Function

S-NSSAI Single-Network Slice Selection Assistance Information

SBA Service Based Architecture

SCEF Service Capability Exposure Function

SCS Service Capability Server

SDAP Service Data Adaptation Protocol

SDK Software Development Kit

SDP Session Description Protocol

SIM Subscriber Identification Module

SIP Session Initiation Protocol

SLA Service Level Agreement

SLF Subscription Locator Function

SMF Session Management Function

SMO Service Management and Orchestration

SUPI Subscription Permanent Identifiers

TrGw Transition Gateway

UAV Unmanned Aerial Photography

UDM Unified Data Management

UDR Unified Data Repository

UE User Equipment

UPF User Plane Function

URLLC Ultra Reliable Low Latency Communications

UTM Unmanned Aircraft System Traffic Management

V2X Vehicle to Everything

ViLTE Video over Long-Term Evolution

Vo5GS Voice over 5G System

VoIP Voice over Internet Protocol

VoLTE Voice over LTE

VoNR Voice over New Radio

VPLMN Visited Public Land Mobile Network

VR Virtual Reality

W3C World Wide Web Consortium

XR Extended Reality

List of Figures

List of Tables

1 Introduction

In this chapter, the thesis topics are briefly introduced in Sections 1.1 and 1.2. Section 1.3 outlines the research objectives and research questions, while Section 1.4 presents the structure of the document.

1.1. Evolution of Mobile Telecommunications: From 1G to 5G

Telecommunications has undergone significant evolution over the past few decades, marked by the development and deployment of several generations of mobile networks. These generations, commonly referred to as 1G through 5G, represent distinct phases of advancement in mobile communication technology.

1G, or the first generation, introduced analog cellular networks in the 1980s. These networks enabled basic voice calls and employed analog modulation techniques for signal transmission.

The transition to 2G, or second-generation network, marked a shift to digital technology. This allowed for more efficient use of the radio spectrum, improved voice quality, and the introduction of basic data services such as SMS (Short Message Service).

With the advent of 3G, or third-generation network, in the early 2000s, mobile communication entered the era of broadband data services. 3G networks provided faster data transmission speeds, enabling services such as video calling, mobile internet access, and multimedia messaging.

The evolution continued with the rollout of 4G, or fourth-generation network, which represented a significant leap in data speeds and network capacity. 4G networks introduced technologies such as LTE (Long-Term Evolution), delivering even faster data rates and enabling advanced services such as high-definition video streaming, online gaming, and mobile commerce. Unlike its predecessors, 4G does not support circuit-switched (CS) technology for voice calls, relying instead on packet-switched methods like Voice over LTE (VoLTE) for handling voice communication.

Each generation of mobile networks has brought about transformative changes in how we communicate and interact with technology, laying the foundation for the increasingly connected world we inhabit today.

Building upon the advancements of previous generations, Fifth Generation Mobile Network (5G) represents the latest phase in the evolution of mobile telecommunications. Developed with characteristics such as increased spectral efficiency, higher data rates, low latency, and superior user experience, 5G aims to deliver performance close to that of fixed networks while offering full mobility and coverage.

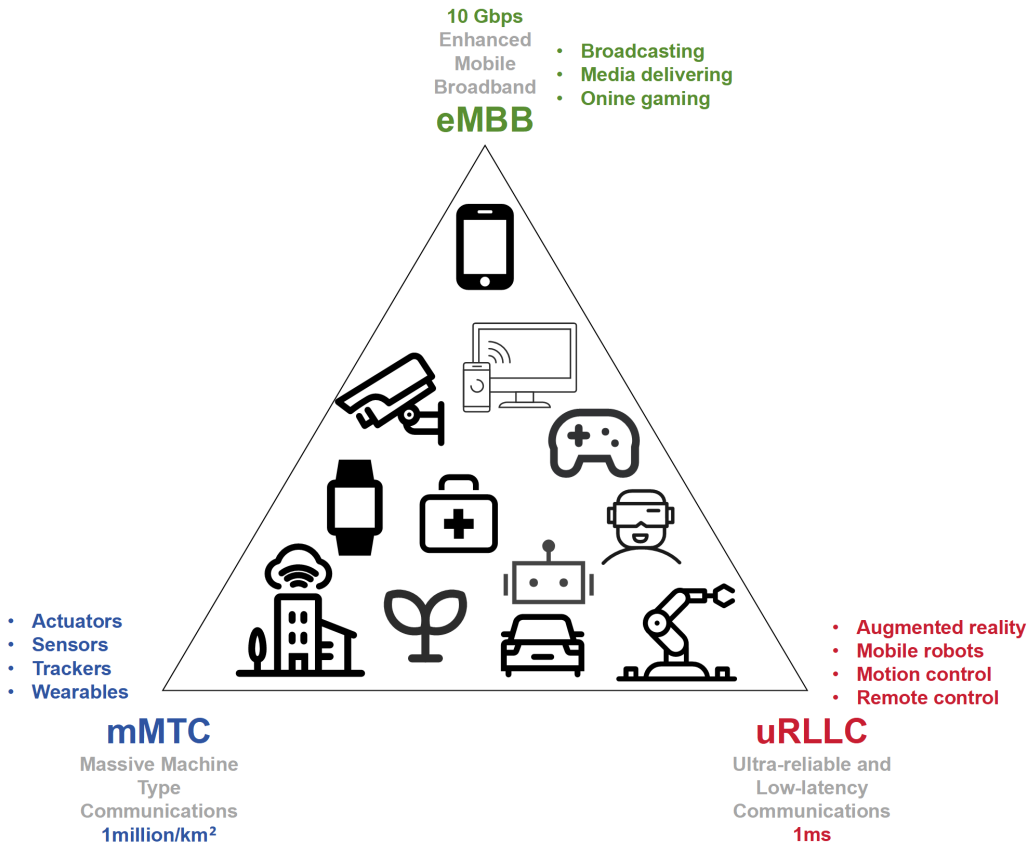


Figure 1.1: 5G use cases [1]

The International Telecommunication Union (ITU), in its 2015 Recommendation ITU-R M.2083, defined 5G's main usage scenarios as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (URLLC).

eMBB addresses the growing demand for a digital lifestyle, focusing on mobile devices and high-bandwidth services such as high-definition video, virtual reality (VR), and augmented reality (AR).

URLLC focuses on meeting the expectations of the digital industry, concentrating on latency-sensitive services such as assisted and automated driving and remote management.

mMTC aims to fulfill the requirements of a more advanced digital society, emphasizing services with high connection density needs, such as smart city and smart agriculture [2].

Figure 1.1 illustrates specific use cases representing the new services offered by network operators in the 5G era, encompassing industrial sensors, wireless watches, and other personal electronics. These specific use cases may exhibit hybrid requirements stemming from one, two, or all three categories.

1.2. Network Capability Exposure

The exposure of 5G network capabilities is crucial to the business model, playing a key role in empowering the network and unlocking significant opportunities within the telecommunications market. Network capability exposure has the potential to open up a sizable and promising market, with the Telecom Application Programming Interface (API) market projected to reach USD 312.83 billion by 2024 and grow to USD 603.66 billion by 2029, reflecting a compound annual growth rate (CAGR) of 14.05% during the

forecast period (2024–2029) [3].

In a 5G network, capability exposure is primarily managed by the Network Exposure Function (NEF), which serves as an API gateway between external applications and internal network functions. Network functions like mobility management, session handling, and user plane functions are exposed through APIs that comply with Third Generation Partnership Project (3GPP) standards. For example, the NEF exposes the Policy Control Function (PCF) and the Access and Mobility Management Function (AMF), enabling external services to interact with user session data and mobility management.

The NEF also facilitates service orchestration, allowing external applications to manage network slices, Quality of Service (QoS), and session continuity. For instance, a third-party service can use API calls to reserve a network slice with specific latency and bandwidth requirements. Security and policy enforcement are integral parts of this exposure process, as the NEF ensures that access control, encryption, and data integrity are maintained. Protocols like Open Authorization (OAuth) [4] and mutual TLS (mTLS) guarantee that only authorized services access network resources.

Network capability exposure allows communication service providers and third parties to access essential network services and data. For example, location-based services can be provided by exposing APIs like the MonitoringEvent API, which retrieves geographic data from connected devices. Applications like emergency response systems benefit from real-time location updates. Similarly, the NEF enables external applications to request specific QoS parameters through the AsSessionWithQoS API. This is particularly useful for services like video conferencing, where low-latency and high-reliability connections are essential. Additionally, the NEF supports network data analytics by exposing functions like the AnalyticsExposure API, giving businesses access to traffic patterns and device behaviors. This helps in optimizing operations, such as predictive maintenance in industrial IoT environments.

In summary, network capability exposure through the NEF is fundamental to realizing the full potential of 5G networks. As the Telecom API market grows, network capability exposure will continue to drive programmability, flexibility, and monetization in 5G and beyond, enabling transformative use cases across industries.

1.3. Thesis Objectives

The objective of this thesis is to describe and analyze network capability exposure in 5G mobile communication networks and propose additional network exposure capabilities to support AR-enhanced communications and drone collision avoidance. The following activities (ACT) and research questions (RQ) are defined to achieve this objective:

- **ACT1** Analyze the architecture of the 5G network, both core and radio, along with the IMS network architecture. These architectures will support addressing the research questions listed below.
- **ACT2** Examine the network capability exposure framework in the 5G network. This includes studying relevant standards from 3GPP, Open Radio Access Network (O-RAN), and the Global System for Mobile Communications Association (GSMA). This framework will form the foundation for the research questions.
- **RQ1** Identify, at a high level, any missing capabilities in the current network capability exposure framework. The results of this research may lead to recommendations for the industry to incorporate these missing capabilities into the network capability exposure standards.
- **RQ2** *Use Case 1.* Investigate the practical use of network capability exposure in an AR-enhanced communication setting. This use case is selected because mobile network operators (MNOs)

are looking to add value to voice calling services, making them more attractive through enriched communication.

- **RQ3** *Use Case 2.* Explore the use of network capability exposure in the context of connected drones. This use case is chosen because the deployment of drones, or Unmanned Aerial Vehicles (UAVs), is becoming increasingly widespread. In particular, reliable communication is critical for UAVs flying "Beyond Visual Line of Sight" (BVLOS).

These ACTs and RQs will form a means for the analysing of network capability exposure and for defining recommendations how this technology may be further enhanced.

1.4. Thesis Structure

The structure of this thesis is as follows: Chapter 2 investigates related papers on current network capability exposure approaches. Chapter 3 provides a detailed description of 5G architectures: 5G core network, next-generation radio access network, and IMS network. In Chapter 4, key concepts of network capability exposure within different standardization organizations are explained. Chapter 5 describes an AR/VR enhanced call use case where new network capability exposure is proposed to control user plane processing. Chapter 6 introduces network capability exposure in a drone connectivity use case and investigates how new network capability exposure can help with collision avoidance. In Chapter 7, the conclusions derived from the findings of this thesis are presented, alongside a discussion on future research directions and potential areas for further exploration.

2

Literature Review

This chapter explores network capability exposure by first examining the key functions that enable it: the SCEF and the NEF. Figure 2.1 illustrates the architecture of SCEF and NEF. The discussion then transitions to a review of relevant research on these two network functions, with the aim of understanding the mechanisms of network exposure and evaluating its impact on emerging technologies and applications.

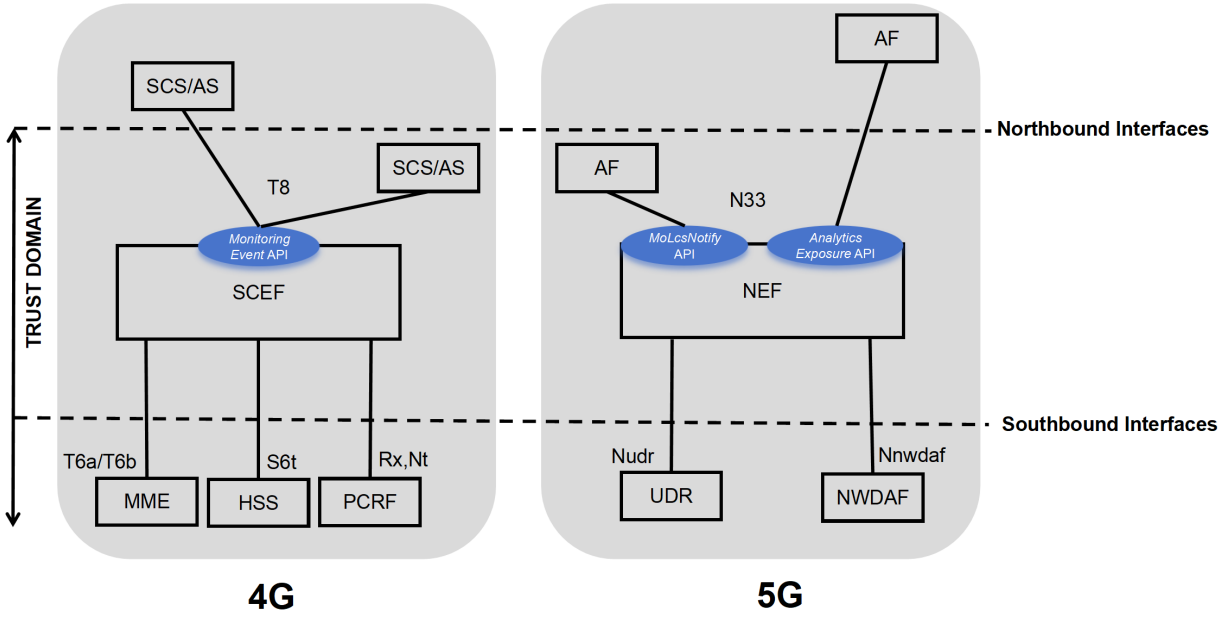


Figure 2.1: Architecture of SCEF and NEF

2.1. Service Capability Exposure Function

In the LTE era, 3GPP introduced the Service Capability Exposure Function (SCEF) in 3GPP TS 23.682 [5] as a dedicated network function designed to securely expose the services and capabilities of 3GPP network interfaces. This exposure is provided through a set of application programming interfaces (APIs) to external systems or applications, specifically the Service Capability Server (SCS) and Application Server (AS), collectively referred to as SCS/AS.

The SCS/AS represents external entities that interact with the core network to enable services such

as IoT device management, data collection, and service control. The SCS is typically responsible for service-specific tasks, such as ensuring appropriate access to network capabilities, while the AS focuses on application-level functionalities tailored to user needs. Together, they allow third-party platforms to securely communicate with the network without direct access to its internal structure.

Using APIs, the SCEF abstracts the internal 3GPP network topology, ensuring that SCS/AS entities access only the services they are authorized to use. This design maintains secure isolation between the SCS/AS and sensitive network components. While the SCEF operates within the network operator's trusted domain, the SCS/AS may exist either within or outside of this domain. For example, an SCS/AS might be a network operator-managed device platform or a third-party service platform with a business relationship with the operator [6].

Starsinic et al. [7] presented an overview of the SCEF API exposing function. This paper then proceeded to provide insights into the services and capabilities exposed by the Northbound APIs. This included illustrative examples across six aspects: APIs for monitoring events and status, APIs for service configuration, APIs for SCS/AS and network coordination, APIs for control plane data exchange, and policy and charging control APIs. These examples effectively demonstrate how these APIs can be employed by a services platform.

Samdanis et al. [8] argued that the SCEF's ability to expose 3GPP service capabilities to third parties transforms operators from simple communication service providers into business enablers. The SCEF plays a crucial role in facilitating operations such as authentication, authorization, and secure access for third parties to the 3GPP network, while ensuring that the infrastructure providers maintain control over the exposed services.

Furthermore, SCEF enables:

- Charging based on offered service and quality provision.
- Quality of service (QoS) provision and service level agreement (SLA) monitoring, allowing third parties to dynamically request and set service priorities.
- Provision of user context information, including real-time user location, user connection properties, average data rate, etc., and network status changes to third parties.
- Admission control for predictable communication patterns, considering factors such as time windows, traffic volume, and pre-scheduled communication timing.

These operations support the allocation of network resources with customized capabilities, taking into account the developer's or third party's business requirements, SLA policy, and service adaptation. This effectively provides the opportunity for network programmability, allowing third parties to efficiently use the retrieved service capability information and optimally exploit the available network resources.

2.2. Network Exposure Function

In the 5G era, 3GPP defined the Network Exposure Function (NEF) as a dedicated Network Function (NF) within the 5G Service-Based Architecture (SBA), with the following functionalities outlined in 3GPP TS 29.522 [9]:

- The NEF shall securely expose network capabilities and events provided by 3GPP NFs to Application Function (AF). The AF is an entity that interacts with 5G core NFs to enable specific services or applications, such as QoS management or policy control. It interfaces with the NEF to access exposed

network capabilities and events, allowing third-party applications to utilize network resources while ensuring security and compliance with operator policies.

- The NEF shall provide a mechanism for the AF to securely transmit relevant information, such as service or policy requests, to the 3GPP network, and may also authenticate, authorize, and regulate the AF's request rate to prevent resource overload, ensuring optimal network performance.
- The NEF shall be able to translate the information received from the AF into the format required by internal 3GPP NFs, and vice versa.
- The NEF shall support to expose information (collected from other 3GPP NFs) to one or more AFs.
- The NEF may support a Packet Flow Description (PFD) function, enabling AFs to provision PFDs, which define the characteristics of specific packet flows such as application identifier, IP protocol, and ports. The NEF can store and retrieve these PFDs in the Unified Data Repository (UDR) for centralized management. Additionally, the NEF provisions the PFDs to the Session Management Function (SMF), which uses them to manage and enforce the appropriate traffic handling and QoS policies for specific application flows within the 5G network.

Gramaglia et al. [10] address the challenge of integrating network operations and service provisioning in the 5G era, where traditional mobile network designs optimized for specific domains fall short of supporting closed-loop automation across all network actors. The authors first highlight scenarios where the 5G core network (5GC) control plane (CP) NFs or network entities (NEs) may need to communicate with other network domains (e.g., AFs provided by a 3rd party, applications, Vehicle to Everything (V2X) servers, or a Multi access Edge Computing (MEC) platform). To facilitate secure cross-domain communication, the importance of service communication restrictions and information translation is emphasized.

Service communication restrictions ensure that only authorized entities can access or exchange information between different domains, thereby preserving data confidentiality and minimizing security risks. By controlling access to network services, these restrictions prevent unauthorized entities from exploiting or exposing sensitive network functions or data. In cross-domain interactions, this is essential as various systems, such as third-party AFs, MEC platforms, or V2X systems, may operate with different security standards.

Information translation is necessary because different domains often use varying identifiers, parameters, and protocols. The NEF handles the conversion of information (such as network slice identifiers, user equipment parameters, and location data) between internal 5G functions and external domains. This ensures seamless interoperability and maintains security by safeguarding sensitive information during the translation process.

3GPP specifies that the NEF can securely expose NF capabilities and events, for example, to third-party AFs and MEC systems. As outlined in 3GPP TS 23.288 [11], one example of this is the collection of data from an external AF for use in network data analytics. Additionally, NEF is responsible for securely providing information from external applications to the 3GPP network, including tasks like data collection from external AFs for network data analytics and User Equipment (UE) parameter/service parameter provision. Beyond security measures, NEF handles the translation of internal-external information in different domains, covering aspects like slice identifier, individual/group identifier, address, and location information.

Then, Gramaglia et al. [10] propose a novel network-wide capability exposure framework aimed at facilitating closed-loop automation by exposing capabilities from various domains, such as network

functions, orchestration, management, and service providers. Closed-loop automation refers to the process where network operations are continuously monitored and automatically adjusted based on real-time data, without requiring manual intervention. This capability is crucial for ensuring optimal performance across the network, especially in 5G, where dynamic use cases like IoT and low-latency services require real-time responsiveness.

The architecture, leveraging registration, discovery, and exposure functions, extends standardization efforts and introduces essential procedures for network capability exposure. The exposure function operates within different domains, such as the network function domain, the orchestration domain, and the management domain, each responsible for specific tasks, such as resource coordination, network operation, and service provision. These domains are responsible for either directly handling intra-domain requests or proxying exposure functions for inter-domain communication between different domains, such as between the core network and service providers.

The framework's feasibility is demonstrated in a real-world testbed, incorporating Artificial Intelligence algorithms for closed-loop management, particularly in the radio access network. The article discusses the application of the proposed framework to three innovative use cases: Edge applications for service provider integration, an enriched interface for service providers, and practical insights into tangible benefits, providing a comprehensive solution to the challenges of network integration in the 5G landscape.

The potential for future investigations resides in the application of the proposed framework to additional use cases. This exploration is contingent on the availability and support of relevant technologies and facilities.

Lin et al. [12] describe the application of 5GC network capabilities. Initially, the paper analyzes various scenarios, use cases, and their corresponding 5GC network capability requirements. Subsequently, it introduces the 5GC network exposure architecture and functions. Most importantly, the paper proposes an enhanced traffic influence capability exposure method.

Capability exposure plays a critical role in this proposed solution, as it enables external applications or other network entities to access specific network functionalities. The solution centers around using edge User Plane Function (UPF) and edge data networks instead of the original core UPF and data network to reduce latency. NEF plays a vital role by exposing key traffic control capabilities, providing independent re-selection capability for the data routing path based on the location of the involved UE. This exposure allows the network to dynamically adjust data routing in real-time, optimizing the path according to the user's position. This aligns with traffic routing configuration under policy charging capability, where the NEF enables the network to make smarter routing decisions autonomously by sharing relevant network data such as UE location and traffic conditions.

In the 5GC lab experiment, the results demonstrate the significant impact of the smart traffic influence capability exposure method. Without it, the average TCore (transfer delay) is approximately 48s, and RCore (transfer rate) is about 1Mbps over seven measurements. With the smart traffic influence capability exposure method, the average TEdge is reduced to 27s, and REdge increases to 1.5Mbps over the same seven measurements. By enabling real-time, location-based traffic routing, the NEF-driven capability exposure reduces delay and improves performance, highlighting its practical benefits for latency-sensitive applications in 5G networks.

The 5GC lab experimental results indicate an improvement in end-to-end service transfer delay and rate. The proposed traffic influence capability exposure method can significantly benefit telecom operators

in the application and deployment of 5GC network capabilities for VR/AR game interaction, Internet of Vehicles, digital venues, and other scenarios that require a high-quality, low-latency experience.

Yu [13] explores network capability exposure for mobile web frameworks, addressing challenges faced by existing 3GPP and European Telecommunications Standards Institute (ETSI) approaches. ETSI's suggestion of third-party applications invoking APIs in a dedicated runtime environment using a special Software Development Kit (SDK) is noted, but potential implementation complexities and costs are highlighted, emphasizing the need for seamless integration with 3rd party Internet Service Providers (ISPs).

Network capability exposure in this paper refers to making specific network functionalities—such as QoS control, user location data, and content delivery—available to external applications or service providers (3rd party ISPs). Through the proposed solution, HTTP Header Enrichment is used to expose these capabilities. By manipulating the HTTP headers during an active session, 3rd party ISPs can access and leverage network data to enhance their services. For example, they can improve service delivery, provide personalized ads, or optimize streaming quality by using the network's QoS data or location-based information. This direct access to network capabilities allows ISPs to provide tailored and improved user experiences without needing to rely on complex API integrations.

The proposed solution employs HTTP Header Enrichment for network capability exposure, allowing 3rd party ISPs to leverage network capabilities within HTTP sessions by manipulating the HTTP header. In this context, 3rd party ISPs refer to external service providers or application developers who are not part of the core telecom network but interact with the network to deliver services to end users. These 3rd party ISPs use exposed network capabilities, such as QoS control, user location, or content delivery, to enhance their services. By enabling such access, telecom operators allow these external ISPs to optimize service delivery, leading to opportunities like personalized advertising or improved streaming quality.

Comprehensive use cases for exposure enablement include Targeted Mobile Advertising, QoS Customization for Video Streaming, and edge Content Delivery Network (CDN).

Advantages of the solution lie in its seamless integration with the mobile web framework, requiring minimal modification to 3rd party ISPs' runtime environments. In this context, a runtime environment refers to the software infrastructure in which the ISP's applications operate, including servers, operating systems, programming languages, and tools. This means that ISPs can implement the solution without needing significant changes to their existing systems, making it cost-effective and easier to deploy.

Future efforts should focus on supporting encrypted web services like HTTPS, which are prevalent in today's networks. Currently, capability exposure via HTTP header manipulation cannot fully resolve this issue, as HTTPS uses encrypted connections with Transport Layer Security (TLS). This encryption makes it difficult to manipulate or extract information from HTTP headers, as they are no longer accessible in the same way. As a result, additional methods or alternative approaches must be explored to support capability exposure within encrypted connections while maintaining the integrity and security of the data.

Li et al. [14] focus on the external exposure of 5G Network Capabilities and its application in real-world markets to address business needs. The study involves the research and development of a 5G capability exposure platform. Through active collaboration with third parties, various end-to-end scenerios are explored: Routing Request Targeting an Individual UE, Routing Request Targeting a Certain Area or Within a Certain Period of Time, QoS Policy and Location Monitoring. and conducts end-to-end field trials.

The research, design, development, and trials of 5G capability exposure are rooted in actual commercial requirements, addressing industry customers' diverse needs for multiple services, rules, user policies,

key information acquisition, and service-related information discovery and management. The goal is to streamline service processes and enable quick duplication of services across the entire network. The field trial results confirm that 5G capability exposure aligns with market demands and substantially improves service experiences. However, the technology is acknowledged to be in an early stage and requires continuous refinement to cater to various service scenarios.

This paper anticipates promising prospects for newly added features, such as 5G local area network (LAN) and Network Data Analytics Function (NWDAF) data analysis in 3GPP R16. These features are identified as potential directions for future research and development, emphasizing the continuous improvement of 5G capability exposure to align with evolving service scenarios.

As 5G-Advanced (5G-A) evolves, harmonizing network communication and sensing becomes crucial. This is because 5G-A aims to integrate both communication and sensing functionalities within the same network infrastructure, allowing wireless systems to simultaneously perform tasks like communication, detection, positioning, and imaging. By harmonizing these functions, networks can not only deliver communication services but also collect and process environmental data to enhance applications like autonomous driving, industrial automation, and augmented reality. This dual functionality helps optimize network resources and enables more intelligent, context-aware services. Network capability exposure technology is important for operators, businesses, and third parties to implement harmonized applications. Lin et al. [15] focus on 5G-A harmonized communication and sensing capability exposure.

The paper outlines the capability exposure hierarchical architecture for harmonized communication and sensing. The proposed network architecture and signaling process integrate capability exposure technology. The paper explores application scenarios based on communication sensing. This insight serves as a reference for technological evolution, network deployment, and application discussions.

Emphasizing harmonized communication and sensing as a key 5G-A/6G technology, the paper stresses the need to integrate this technology with network capability exposure. This fusion is seen as the essential approach for realizing harmonized communication and sensing applications. Lin et al. [15] systematically addresses capability exposure hierarchical architecture for communication integration, analyzes the evolutionary trajectory and network processes, and explores application scenarios. It provides valuable insights for future technological evolution, network deployment, and application discussions for harmonized communication and sensing capabilities.

The integration of harmonized communication and sensing technology with network capability exposure is identified as a crucial pathway, especially in the context of the anticipated 5G-A/6G era characterized by overall exposure, intelligent, and virtual network architectures. The paper systematically examines the hierarchical architecture of capability exposure in the realm of communication integration, analyzes the evolutionary trends and basic network processes of harmonized communication and sensing network architecture, and explores diverse capability exposure application scenarios grounded in harmonized communication and sensing. The comprehensive insights presented in this paper contribute to the understanding of the future technological landscape, network deployment strategies, and application considerations for harmonized communication and sensing capability.

Given the current scenario, NEF plays a crucial role in enabling third-party application providers to harness 5G network capabilities. Although 3GPP defines open standardized APIs through NEF, a market gap remains due to the absence of commercial solutions. Addressing this, Fragkos et al. [16] introduce 'NEFSim', an open-source simulator that provides a configurable environment for developers to experiment with NEF's Northbound APIs.

The paper [16] emphasizes the openness and programmability of 5G, detailing NEFSim's architecture and implementation while classifying NEF services according to 3GPP standards. NEFSim resides in a virtualized, container-based environment that emulates NEF's behavior within a 5G core network, allowing developers to test APIs by simulating 5G small cells, gNBs, and UEs. This setup enables realistic testing of scenarios like user mobility, QoS changes, and session management. While NEFSim operates independently from actual 5GC southbound network functions, it mimics their behavior to support accurate testing of capabilities such as monitoring events, location reporting, and policy control.

Future research will focus on implementing additional NEF APIs to support diverse use cases. The paper envisions integration with Open5GS [17], a platform for deploying 5G core networks, to establish communication through southbound APIs, evolving NEFSim into a 3GPP-compliant emulator. As of writing, Open5GS complies with 3GPP Rel. 16 standards but lacks NEF implementation. NEFSim currently emulates NEF by simulating northbound API interactions, enabling developers to test services like monitoring events, session establishment with QoS, and traffic influence in a virtual 5G network environment without the need for actual 5GC deployment.

Santos et al. [18] present a location-aware framework designed to optimize video delivery in high-speed scenarios. This framework utilizes the dynamic capabilities of Mobile Edge Computing (MEC) within the 5G network, introducing efficiency enhancements to address the unique challenges of delivering video content in such environments. By deploying virtualized CDN (vCDN) nodes at edge sites controlled by a centralized unit, the framework efficiently follows users, conserving both network and computational resources.

The proposed framework leverages 5G core network Northbound APIs to obtain user location data and influence user traffic. Technically, this framework influences key aspects of network traffic, such as routing decisions, QoS adjustments, and traffic prioritization. By accessing user location data, the framework can dynamically reroute traffic to optimize network resources, minimize latency, and ensure higher reliability based on the user's proximity to specific network resources, such as edge servers or base stations. Additionally, it allows fine-grained control over traffic flow, enabling applications to enhance user experience by adjusting QoS parameters or prioritizing critical traffic flows. Focusing on the NEF, which provides this information, the authors specifically explore the mechanisms related to the NEF. The NEF produces data representing the location of a train moving along a defined path. To provide this simulated train location to subscribers, the Monitoring Event API is indeed implemented in NEF. MEC hosts are strategically distributed along the user's movement path. The MEC Platform, functioning as an AF, directs traffic to the MEC Dataplane by interacting with the 5G Core's NEF. Additionally, the NEF supplies monitoring data to the Monitoring Manager (also an AF), which distributes it to other modules for the application of various control algorithms. This process helps optimize network performance by efficiently managing user data and network resources. The results demonstrate a 10.9% reduction in core network load compared to traditional methods without NEF-driven optimizations, where the core network handles a higher volume of data traffic without the offloading capabilities provided by MEC. There is also a 3.4% increase in cache hit ratio, meaning more requests are successfully served from the local cache, reducing the need for repeated data retrieval from the core network. Furthermore, a 35% saving in MEC resources is achieved, indicating that the resource management strategies implemented through NEF allow MEC hosts to function more efficiently, using fewer resources to handle the same workloads.

In the realm of future work, there is potential for extending the utilization of the 5G Core network's APIs. One avenue involves identifying low-coverage zones where users' Quality of Experience (QoE) drops below acceptable levels. Leveraging this capability, combined with strategically placing a cache

within customer premises equipment (CPE) on moving vehicles, allows for content placement during transit through these low-coverage areas. The presented system introduces several configurable parameters, providing opportunities for fine-tuning to optimize outcomes. These parameters, including the location prediction algorithm, node instantiation timings, video chunk size, and viewing status report period, can be adjusted for further refinement. As part of future endeavors, the authors aim to conduct comprehensive testing of this system within a commercial MEC-enabled 5G deployment to analyze its behavior in real-world scenarios.

3

Introduction to 5G

As this thesis focuses on the study of network capability exposure in 5G, a preliminary step involves a comprehensive examination of the overall architecture and components within the 5G network.

This chapter offers an overview of 5G architecture, as shown in Figure 3.1. Sh, Cx, Rx and Gm are interfaces between IMS and 5GS. Further insights into specific aspects are offered in the subsequent sections: Section 3.1 examines the core network architecture, while Section 3.2 explores the RAN architecture. Section 3.3 provides insights into the IMS framework and service.

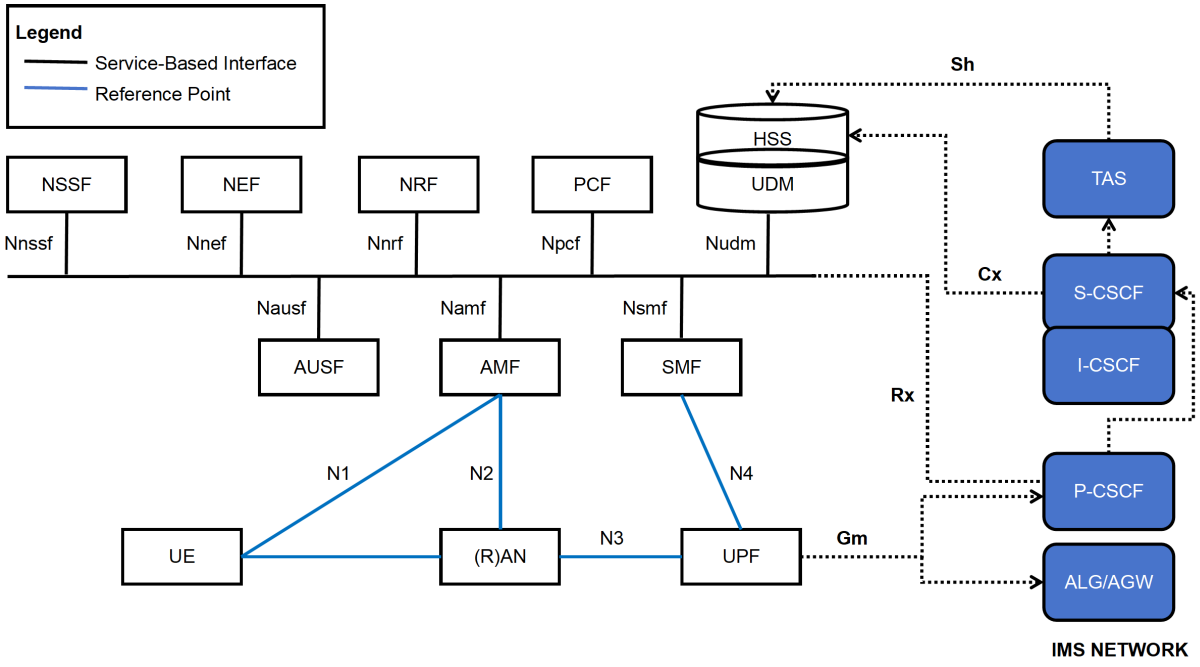


Figure 3.1: 5G Network architecture [19]

3.1. 5G Core Network

The 5G core network architecture, standardized in 3GPP TS 23.501 [20], is designed to meet the increased throughput demand, reduced latency, and enhanced reliability required by diverse applications and services in the 5G ecosystem.

Defined by 3GPP, the new 5G core adopts a cloud-aligned, service-based architecture (SBA), as depicted in Figure 3.2. The upper section of Figure 3.2 represents the 5GC Control Plane (CP) and features a service-based interface provided by individual network functions. In this architecture, a CP network function (e.g., SMF) allows other authorized NFs to access its services. NFs within the 5GC Control Plane should exclusively use service-based interfaces for their interactions [20]. This architecture encompasses all 5G functions and interactions, including authentication, security, session management, and traffic aggregation from end devices. Additionally, the 5G core places significant emphasis on Network Function Virtualization (NFV) as a foundational design concept, integrating virtualized software functions deployable within the network.

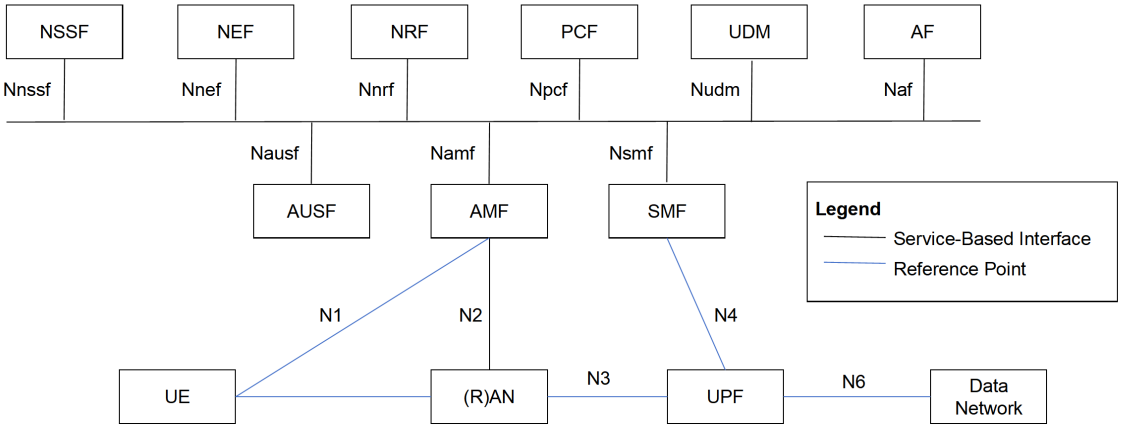


Figure 3.2: Service-based 5G Core Network architecture [19]

The NFs within 5GC and their capabilities are defined as below [20]:

Access and Mobility Management Function (AMF)

- Manages registration, mobility and connection management of UEs in the 5G System.
- Coordinates signaling between UEs and other network functions.
- Provides service operations for handling N2 point-to-point signaling between the RAN and the 5G core network.
- N1 is a Non-Access Stratum (NAS) protocol that manages signaling directly between the UE and the AMF. It is transmitted via the RAN to reach the core network.

Session Management Function (SMF)

- Session Establishment, modify and release, including tunnel maintenance between UPF and UEs.
- UE IP address allocation and management.
- Control part of policy enforcement and QoS.
- Selects and controls the UPF, also configures traffic steering at UPF to route traffic to proper destination.

User Plane Function (UPF)

- Handles the user plane path of Protocol Data Unit (PDU) sessions.
- Enables packet routing, packet inspection, and traffic usage reporting.
- Handles the user plane part of policy rule enforcement, such as gating, redirection, and traffic steering.

Network Exposure Function (NEF)

- Supports the secure exposure of network functions capabilities and events to third parties, for example application functions.
- Manages the external open network data, and all external applications that want to access the internal data of the 5G core.
- Authenticates, authorizes, and regulates the AFs.

Network Repository Function (NRF)

- Supports interconnection between 5G Core NFs by performing managing functions for both monitoring service status and interworking information of 5G Core NFs which changes dynamically.
- Provides the discovery of a set of network function instances with a specific service and the discovery of specific services.

Policy Control Function (PCF)

- Provides a unified policy framework incorporating network slicing, roaming, and mobility management.
- PCF services allow other network functions to create and manage mobility and session policy associations in the PCF to receive policy information.
- To support QoS, the PCF collects packet flow information from the application function for policy control.

Network Slice Selection Function (NSSF)

- Supports the selection of network slices and the selection of an AMF set that can serve the UE.
- Also supports determining the allowed Network Slice Selection Assistance Information (NSSAI) and configured NSSAI, as well as, if needed, the mapping to subscribed Single-NSSAIs (S-NSSAIs).

Application Function (AF)

- Can be an internal AF, which is managed by the network operator, or an external entity that communicates with the 5GC via the NEF.
- Trusted AFs are external entities authenticated and authorized by the network operator, allowing them to interact with NFs via the NEF.
- Untrusted AFs are third-party entities that are not authenticated by the network operator and must access NFs through the NEF to ensure security and policy compliance.
- Internal AFs are inherently trusted and do not fall under the categories of trusted or untrusted, as they are part of the operator's infrastructure.
- Influences routing by steering its traffic toward external edge servers.
- Supports IMS interactions with the 5GC.

Unified Data Management (UDM)

- Supports the registration management of network functions that serve the UE.
- Performs user identification handling, access authorization and subscription data management.
- Provides management of user subscription data and authentication data.
- Acts as a Subscription Locator Function (SLF) by managing the mapping of Subscription Permanent

Identifiers (SUPI) and Generic Public Subscription Identifiers (GPSI) to their corresponding subscription data. This functionality allows for efficient retrieval and management of subscriber information across different network slices and user profiles.

Authentication Server Function (AUSF)

- Provides UE authentication services to consumer AMFs, which are the AMFs that request authentication on behalf of the UE. This functionality supports authentication for both 3GPP and non-3GPP access networks.

3.2. Next Generation Radio Access Network

To accommodate the enhanced features of 5G, including faster data rates, lower latency, and improved reliability, a new radio access technology known as New Radio (NR) has been introduced. This NR technology seamlessly integrates with the 5G core network architecture.

The Next Generation Radio Access Network (NG-RAN), illustrated in Figure 3.3, functions as the link between UE and the 5GC. Key components within NG-RAN include base stations, radio access network nodes, and various network functions [21].

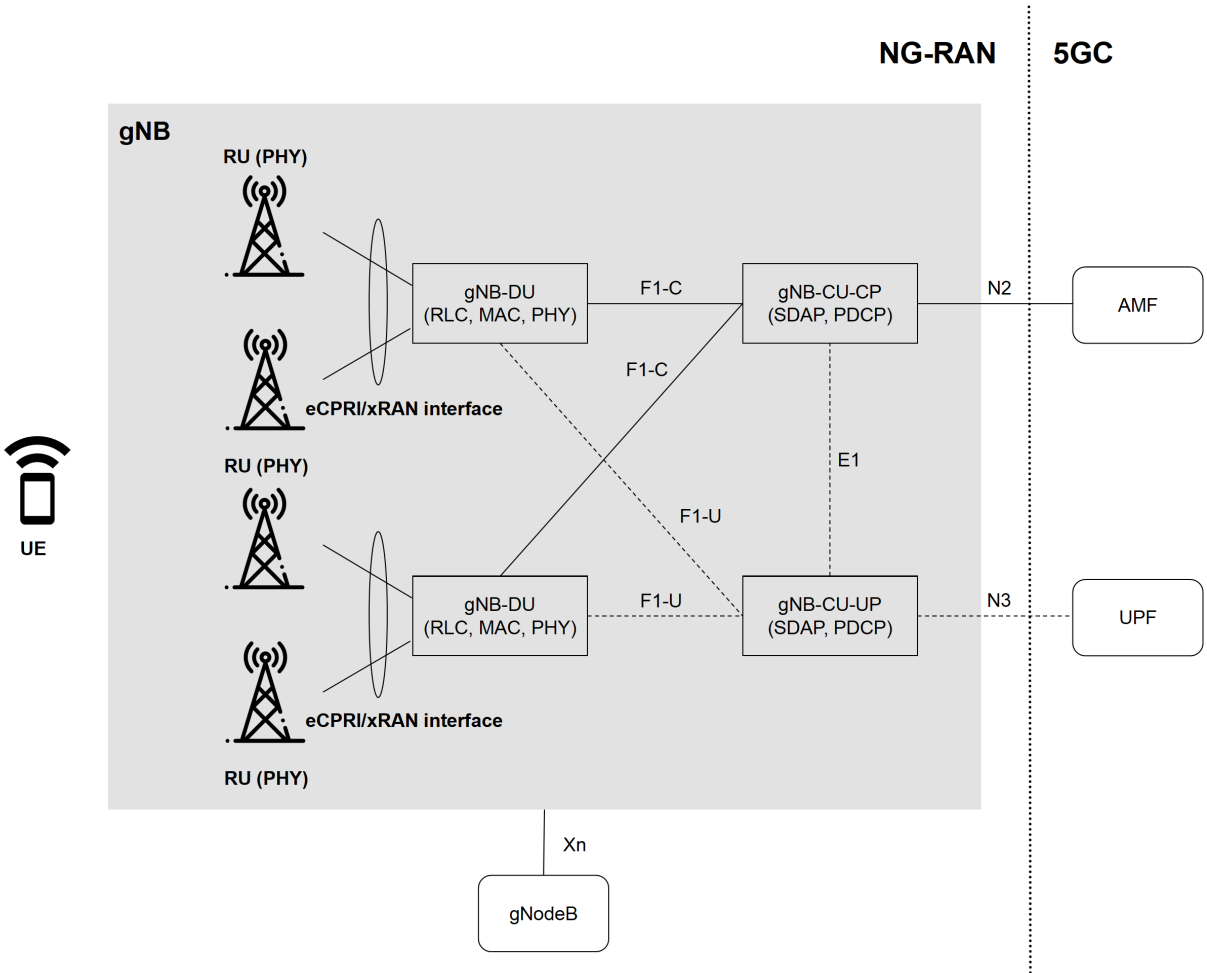


Figure 3.3: NG-RAN Architecture [22] modified

gNodeB The physical radio equipment providing wireless connectivity to UE is referred to as the base

station in cellular networks. In the context of the 5G network, this base station is known as the Next Generation Node B (gNB). The gNB is responsible for both radio transmission and reception, along with the management of radio resources. Connectivity to the 5G core network is established through the NG interface.

The gNB is intentionally designed with a high degree of flexibility and scalability. It accommodates various frequency bands, supports multiple antenna technologies, and adapts to diverse deployment scenarios. This adaptability extends to both centralized and distributed deployment models. In a centralized setup, the gNB is linked to Central Units (CUs) and Distributed Units (DUs), which can also be combined into a single entity.

Radio Access Network Nodes The RAN nodes encompass the CU and DU, which are responsible for radio resource management functions. The CU centrally manages higher-layer radio resource control functions, while the DU handles lower-layer functions closer to the physical layer. Together, the CU and DU ensure efficient and scalable radio resource management, establishing connectivity with user equipment. In a 5G network, these RAN nodes connect to the 5G Core Network via the NG interface. The X2 interface is used between eNodeBs in LTE networks for inter-cell signaling and handover, but in 5G, this function is replaced by the Xn interface, which enables similar communication between gNBs for inter-gNB signaling and handovers.

3.3. IMS Network

IMS is a standardized architecture that facilitates the delivery of a wide range of multimedia services over Internet Protocol (IP) networks, including VoLTE, Video over Long-Term Evolution (ViLTE), messaging, and data services [11]. Defined by 3GPP, IMS is widely implemented in mobile networks. Figure 3.4 illustrates the IMS reference architecture, highlighting interfaces with the CS network and other IP-based multimedia networks.

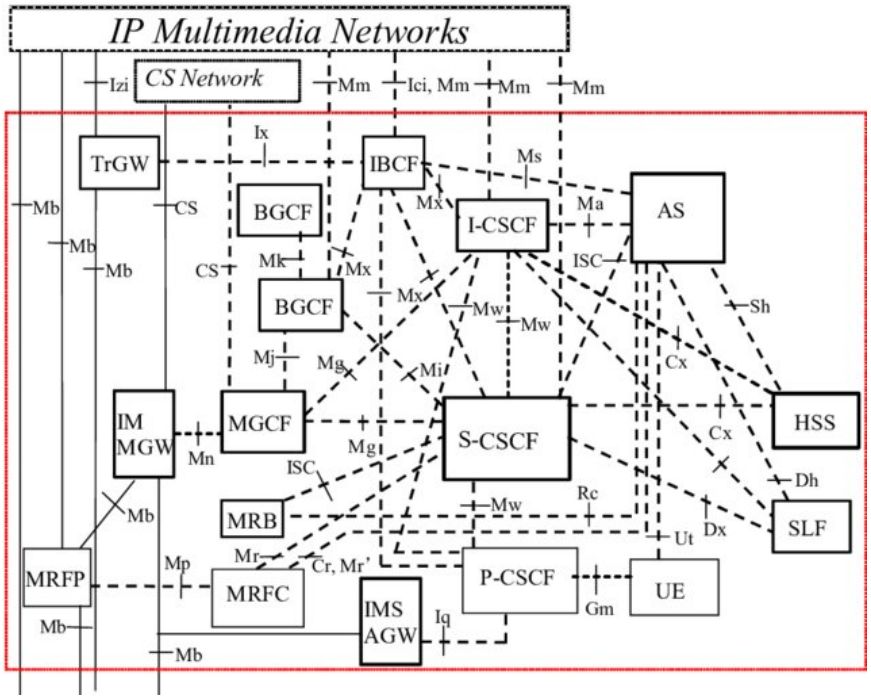


Figure 3.4: Supporting Vo5G/VoNR with enhanced IMS within the Service-Based Architecture [11]

3.3.1. IMS Core Components

The IMS architecture involves several core components crucial for enabling communication. The descriptions of the components and their respective roles are as follows [23]:

Call Session Control Function (CSCF)

The Call Session Control Function (CSCF) is responsible for handling call/session registration, control, and routing within the IMS network. There are three types of CSCFs, each serving specific roles within the IMS framework.

Proxy CSCF (P-CSCF) The Proxy Call Session Control Function (P-CSCF) serves as the initial point of contact for UE when establishing communication sessions. It operates as a proxy and, in specific scenarios, acts as a User Agent. The P-CSCF ensures the integrity of Session Initiation Protocol (SIP) messages throughout the communication process.

Key Functions

- Forwarding SIP register requests from the UE to the appropriate I-CSCF based on the UE's home domain.
- Relaying SIP messages received from the UE to the designated SIP server (e.g., S-CSCF) obtained during the registration process.
- Handling emergency session establishment requests according to specified error handling procedures.
- Generating Call Detail Records for transaction records.
- Establishing and maintaining Security Associations with each UE.
- Performing SIP message compression and decompression.
- Authorizing bearer resources and managing QoS.

Interrogating CSCF (I-CSCF) The I-CSCF routes requests to the appropriate S-CSCF within the network or for roaming users. It queries the HSS to obtain the address of the relevant S-CSCF to process SIP initiation requests. The I-CSCF primarily handles routing during registration to determine the assigned S-CSCF and routes SIP requests from other SIP networks by referencing the HSS.

Key Functions

- Assigning a Serving-CSCF (S-CSCF) to a user during the SIP registration process.
- Routing SIP requests received from other networks to the appropriate S-CSCF.
- Retrieving the S-CSCF address from the HSS.
- Forwarding SIP requests or responses to the determined S-CSCF.
- Generating Call Detail Records to maintain transaction records.

Serving CSCF (S-CSCF) The S-CSCF maintains session state information essential for supporting various services as required by the network operator, including registration data. Different S-CSCFs within an operator's network may have varying functionalities tailored to specific requirements.

Key Functions

- Acting as a registrar, accepting registration requests, and making its information available through the location server (e.g., HSS).
- Controlling sessions for registered endpoints, including rejecting IMS communication to/from barred public user identities.
- Behaving as a Proxy Server, handling requests internally or forwarding them.
- Behaving as a User Agent, capable of terminating and independently generating SIP transactions.
- Providing endpoints with service event-related information, such as notification of tones/announcements and location of additional media resources, along with billing notifications.
- Obtaining the Address of the I-CSCF for the network operator serving the destination user and forwarding SIP requests/responses accordingly.
- Routing SIP requests/responses based on the type of procedure.

Home Subscriber Server (HSS)

The HSS is a core network element in the IMS architecture, playing a pivotal role in facilitating various services and functions.

Key Functions

- Subscriber Profile Management: The HSS stores detailed information about each subscriber, including their IMS public user identity, contact information, allocated services, and preferences. This data is essential for session setup, authorization, and routing decisions.
- Authentication and Authorization: The HSS authenticates users when they attempt to access IMS services, verifying their credentials (such as username and password) before granting access. It also provides information about the services and capabilities available to each subscriber.
- S-CSCF Assignment: The HSS tracks the assigned S-CSCF for each subscriber, which is crucial for directing session initiation requests and routing calls and messages within the network.
- Subscriber Location: The HSS maintains information about subscribers' locations within the network, which is vital for ensuring that calls and messages reach the appropriate serving network elements.
- Security Functions: The HSS contributes to the security architecture of IMS by storing security-related keys and information necessary for secure communication between the subscriber's device and the IMS network.
- Policy and Charging Control: The HSS may also be involved in policy control and charging decisions, providing information about the subscriber's subscription and policy rules that dictate service delivery and charging.

3.3.2. Protocols Used in IMS

In the realm of communication services, IMS serves as the carrier's service platform, distinguished by its access-agnostic nature, allowing support for various access types. It collaborates with access networks to deliver carrier-grade quality calls and ensure quality of service (QoS). This characteristic positions IMS as an optimal platform for telecommunication services, as it theoretically requires only one service platform to provide communication services across the diverse access networks operated by the carrier.

The primary protocols within IMS are the Session Initiation Protocol (SIP), the Session Description Protocol (SDP), and the RTP (Real-time Transport Protocol). SIP facilitates the communication of call

and session-related signaling between network nodes and clients, while SDP is employed to describe the pertinent media aspects of the call or session, such as voice media details (e.g., codec specifications). While these protocols are also utilized by non-carrier Voice over IP (VoIP) services, IMS stands out due to its close interaction with access networks. This is in contrast to typical VoIP services, which often view the underlying transport as a mere data pipe lacking guaranteed quality.

The subsequent section offers a detailed introduction to and the functions of these two protocols [24].

Session Initiation Protocol (SIP)

SIP serves as a signaling protocol crucial for initiating, maintaining, modifying, and terminating real-time sessions involving video, voice, messaging, and other communication applications and services over the internet. Positioned at the application layer of the Open Systems Interconnection (OSI) model, SIP is widely employed for Voice over Internet Protocol (VoIP) and various multimedia communication applications.

Within its functionality, SIP is utilized to initiate communication sessions among devices or applications. Whether a user is making a call, sending a message, or establishing a multimedia session, SIP takes on the responsibility of setting up the connection. It manages the signaling for call control functions, covering operations such as call setup, call hold, call transfer, and call release. The protocol precisely defines how these operations should be communicated between endpoints.

Furthermore, SIP facilitates the invocation of various services within IMS, including call forwarding, call waiting, and other supplementary services. It is also tasked with modifying existing sessions, such as adding participants to a call, and terminating sessions when the user concludes the communication.

It is crucial to note that while SIP excels at signaling, it doesn't handle the actual media transfer (audio, video, etc.). To achieve media transmission, SIP collaborates with protocols like RTP.

Signaling for Establishing, Modifying, and Terminating Sessions

The SIP signaling session establishment process for both mobile-originated and mobile-terminated calls is illustrated in Figure 3.5, with some SIP messages also traversing intermediate nodes not depicted in the figure.

Session Establishment [25]:

1. **SIP INVITE:** The VoNR Calling (A) Party User initiates a Voice Call by sending a SIP INVITE request. This SIP Invite includes the SDP offer with IMS media capabilities, specifying details such as the required codec and bandwidth.

   100 Trying: Upon receiving SIP INVITE messages, each node acknowledges the previous node by sending a '100 Trying' response. The I-CSCF locates the S-CSCF address by contacting the UDM network function.

2. **SIP 183 Progress:** The terminating (B) party responds with an SDP answer in a SIP 183 Progress message. This SDP answer specifies the supported codec and indicates that preconditions are desired but not yet met on the terminating side. During this process, dedicated bearers are established on both the A and B party sides in 4G networks, facilitated by the PCRF connecting the P-CSCF and the 4G network.

3. **PRACK:** The Originator (A) Party sends a Provisional Response Acknowledgment, a provisional acknowledgment used to acknowledge SIP provisional responses like 180 Ringing, 183 Session Progress, etc.

   200 OK for PRACK: The Called (B) Party responds to the PRACK with a 200 OK.

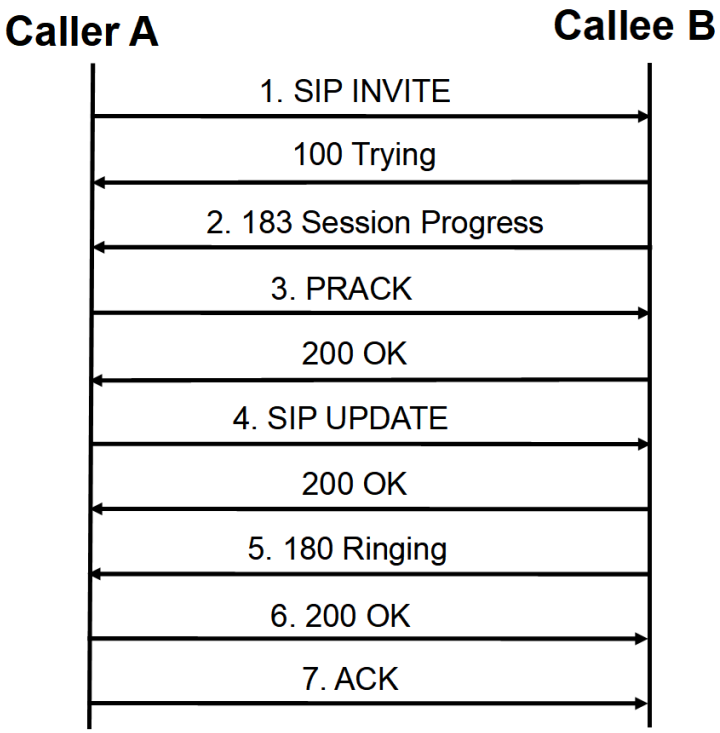


Figure 3.5: SIP signaling for session establishment [25] modified

4. **SIP Update:** The Calling (A) Party reserves internal resources to reflect the SDP answer, confirming resource reservation by sending a SIP UPDATE message with a new SDP Offer. The offer contains the selected codec and information indicating that the local preconditions have been met at the originating (A) Party side, and the media stream is now set to active.

   200 OK for Update: The 200 OK for the SIP UPDATE response with the SDP answer contains the agreed voice codec and confirmation that the preconditions are met at the terminating (B) Party side, and the media stream is active.

5. **SIP 180 Ringing:** The Called (B) Party initiates ringing and replies with a SIP 180 Ringing response.
6. **200 OK for INVITE:** The Called (B) Party answers the call, responding with a 200 OK to the Calling (A) Party.
7. **ACK:** The last ACK confirms that the call has been established. The voice traffic flows over the dedicated bearer from A Party IMS to B Party IMS, then to B Party and finally to the Called Party User via the dedicated LTE bearer of B Party.

Session Modification:

If a user wants to modify ongoing session parameters, such as changing the media codec, the SIP signaling is used to send an UPDATE request. This request is processed by the SIP servers and communicated to the affected parties.

Session Termination:

When a user decides to end a session, their device sends a SIP BYE message. This message is processed by the SIP servers, and appropriate responses are sent to indicate the termination of the session, typically including a SIP 200 OK response from the recipient party acknowledging the termination.

Additionally, any involved SIP proxies or servers may also send their own 200 OK responses to confirm the receipt of the BYE message. SIP is a text-based protocol, and it uses URLs similar to those used on the web to identify users and services. This makes it versatile and easy to integrate into existing internet infrastructure.

SDP (Session Description Protocol)

SDP is a protocol used in multimedia communications to negotiate and describe sessions between participants. It provides a concise and human-readable way to convey information about the characteristics of a multimedia session.

Session negotiation involves information about the type of media (e.g., audio, video, application data) that will be exchanged during the session, codecs and formats used for media encoding and decoding, the transport protocol and ports for sending and receiving media packets, information about the IP addresses and ports where media should be sent, and details about timing aspects like session start and end times, as well as synchronization of different media streams.

SDP provides a structured text-based format for describing session parameters, such as media types, codec information, and transport details. This session description is typically conveyed within SIP messages.

SDP is crucial for establishing media streams, specifying how media should be encoded, transported, and synchronized between participants to ensure that audio, video, and other data are transmitted and received correctly. It also promotes interoperability, allowing devices and applications from different vendors to communicate effectively in diverse multimedia environments.

Furthermore, SDP can be updated dynamically during a session. For example, if network conditions change, SDP can be used to renegotiate parameters like codecs or bitrates to ensure the best possible quality.

3.4. Voice Call in 5G Networks

This section explores the key interactions and protocols that enable voice call functionality within 5G networks. Specifically, we examine how the PCF interacts with the P-CSCF and Access Session Border Gateway (A-SBG), as well as the role of WebRTC in facilitating communication between the UE and the Multimedia Telephony Application Server (MTAS). These interactions provide the foundational understanding for voice service implementation in a 5G environment, which will be discussed in greater detail in the following chapters.

3.4.1. PCF interaction with P-CSCF and A-SBG

In the VoNR call flow, the PCF interacts with the A-SBG to enforce policy rules, manage QoS, and apply charging policies related to the voice session. Below shows how these interactions occur:

1. PCF and P-CSCF Interaction:

The P-CSCF is responsible for managing the signaling related to SIP (Session Initiation Protocol) and acts as the initial point of contact for the UE in the IMS core network during the VoNR call setup.

During the call establishment, after receiving a request for service (e.g., voice or video call) from the UE, the P-CSCF communicates with the PCF to obtain PCC rules.

PCF Functions in Interaction with P-CSCF:

- Npcf_PolicyAuthorization_Update: The P-CSCF sends this request to the PCF to trigger policy decisions, such as service prioritization, QoS enforcement, traffic shaping, and policing.
- The PCF decides how the session should be managed based on the network's policies and the subscriber's profile, such as allowed QoS levels, charging rules, and any applicable service restrictions.
- The PCF sends the policy decision back to the P-CSCF, allowing it to enforce the correct QoS and apply the appropriate service policies for the voice call.

2. PCF and A-SBG Interaction:

The A-SBG sits at the edge of the network and handles the control and media plane traffic between the access and core networks. It is responsible for managing IP multimedia sessions and providing security features like NAT traversal, encryption, and session management.

The PCF may interact with the A-SBG indirectly through the P-CSCF to enforce policies for managing media sessions and ensuring secure media transport.

PCF Functions in Interaction with A-SBG:

- QoS Enforcement: The PCF provides the necessary policies for ensuring that the QoS rules applied at the A-SBG are aligned with the desired QoS for the VoNR session. For instance, the PCF can enforce QoS levels that ensure low latency and high reliability for the voice session.
- Session Authorization: The PCF decides whether the session initiated by the UE is allowed based on the subscriber's profile and network conditions. This authorization helps the A-SBG determine whether to allow the session to proceed or apply any restrictions.
- Charging Policies: The PCF communicates charging information to the A-SBG via the P-CSCF to apply the correct charging policies to the voice session.

3.4.2. WebRTC between UE and MTAS

WebRTC enables peer-to-peer (P2P) communication by allowing browsers and mobile applications to communicate directly without requiring plugins. It uses SIP signaling for call setup and SRTP (Secure Real-time Transport Protocol) for media encryption.

The UE, such as a mobile phone or a tablet, participates in a WebRTC-based session for voice or video communication. The UE interacts with the MTAS via the 5G network and IMS core using WebRTC for real-time media transmission and SIP signaling for session management.

The key components on the UE side for WebRTC communication include:

- WebRTC APIs: These enable the UE to capture media (e.g., audio and video streams) and handle peer connections.
- SIP Client: Manages session setup, modification, and teardown using SIP messages.
- STUN/TURN/ICE Framework: Manages firewall/NAT traversal, helping the UE discover the best media path to the MTAS.

The MTAS is an IMS-based telephony server responsible for multimedia services like voice, video, and messaging in 4G/5G networks. It manages SIP signaling for call control, session handling, and QoS enforcement. In a WebRTC scenario, the MTAS acts as the bridge between the UE and the traditional IMS-based telephony services. Figure 3.6 shows the position of MTAS in the network.

The general WebRTC call flow between the UE and MTAS is as follows:

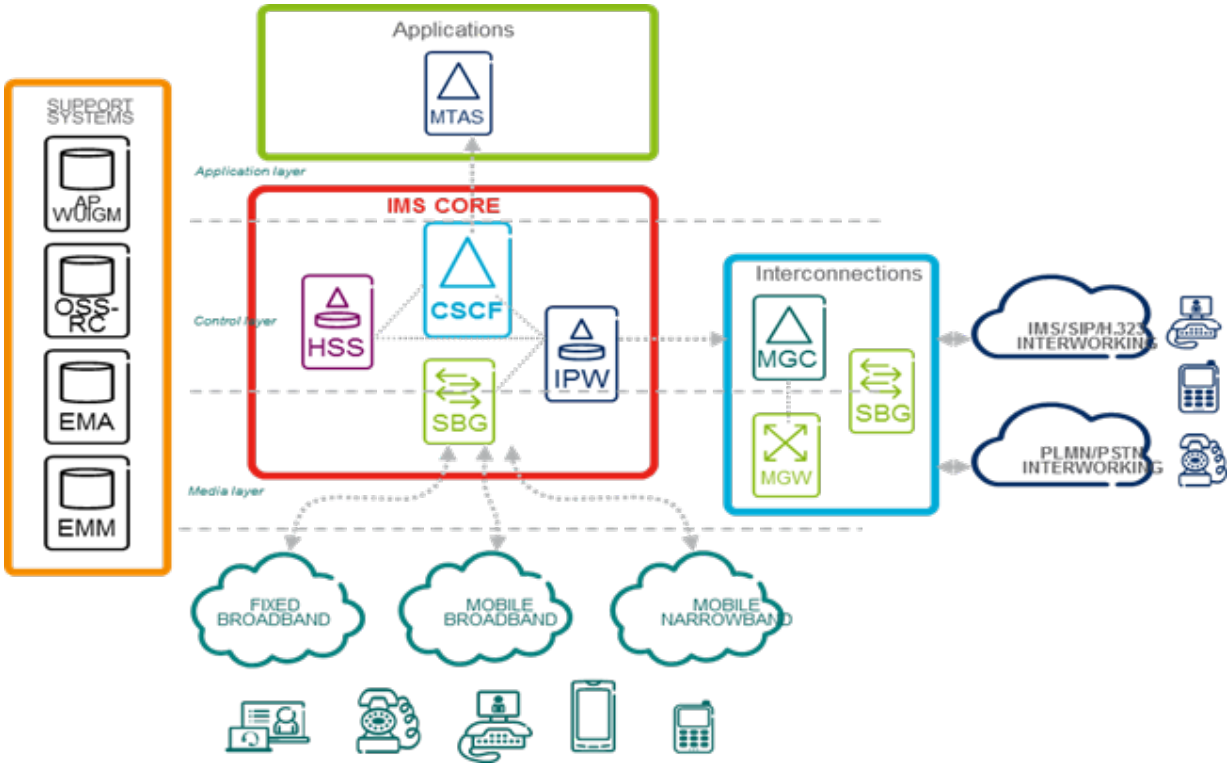


Figure 3.6: Position of the MTAS in the Network Architecture [26]

Step 1: WebRTC Call Initialization (SIP signaling) The UE initiates a WebRTC session by sending a SIP INVITE request to the MTAS through the 5G network and the IMS core. This request is sent over the signaling channel.

The MTAS receives the INVITE, processes the request, and sends back a SIP 200 OK to the UE, acknowledging the session initiation.

Step 2: SDP Exchange for Media Negotiation In the SIP INVITE, the UE includes an SDP (Session Description Protocol) offer, which contains information about the media formats, codecs (e.g., Opus for audio, VP8/VP9 for video), and transport protocols it supports.

The MTAS responds with an SDP answer in the 200 OK message, confirming the media parameters and setting up the session. At this point, both the UE and the MTAS have agreed on the codecs, ports, and protocols for media transmission.

Step 3: ICE/STUN/TURN for NAT Traversal

WebRTC relies on the Interactive Connectivity Establishment (ICE) framework to establish the best possible path for media flow between the UE and MTAS.

The UE sends STUN or TURN requests to discover its public IP and port if it's behind a NAT (Network Address Translation) device. This helps to find a direct or relayed media path. The MTAS also participates in ICE, ensuring it can connect with the UE either directly or through a relay server (TURN) if necessary.

Step 4: Media Exchange (SRTP)

After the media path is established via ICE, the media streams (audio and/or video) are exchanged between the UE and MTAS using SRTP for encryption and DTLS (Datagram Transport Layer Security) for key negotiation.

The UE captures the audio/video streams via WebRTC's APIs and sends them to the MTAS, which routes the media to its destination (e.g., another user, PSTN, or IMS user).

The MTAS processes the media, potentially applying transcoding if the codecs used by the UE and the other party differ.

Step 5: Call Control and QoS Management

During the session, the MTAS ensures QoS (Quality of Service) by interacting with the PCF (Policy Control Function) and P-CSCF to apply network policies that prioritize voice and video traffic, ensuring low latency and high reliability. Call control (hold, mute, end call) is managed via SIP signaling between the UE and MTAS.

Step 6: Session Termination

When the user ends the call, the UE sends a BYE request to the MTAS via SIP.

The MTAS acknowledges the termination by sending a 200 OK response.

The media path is torn down, and the session is closed.

In general, WebRTC enables real-time media exchange between the UE and the MTAS by leveraging SIP for session management and SRTP for secure media transmission. The UE communicates with the MTAS via the IMS core network, utilizing SIP signaling for call setup and control. The MTAS ensures QoS and policy enforcement by interacting with the network's PCF and other IMS components. This allows real-time communication with high security and low latency over a 5G network.

4

Network Capability Exposure

This chapter examines the processes involved in network capability exposure within the core network architecture, O-RAN architecture, operator platform and CAMARA Project. It explores how communication service providers and other parties can access network capabilities, including data and services. By implementing security and data integrity policies, network data and resources can be made available to various ecosystems, fostering innovation in enterprise applications. This chapter explores the fundamental elements and functionalities associated with network capability exposure, focusing on the core network, radio access network, operator platform, and the CAMARA Project. From comprehending the 5G system architecture to unraveling the structures governing exposure capabilities, the chapter explores the layers, functions, and applications that characterize network exposure in contemporary telecommunication systems.

4.1. Network Capability Exposure in Core Network

3GPP defined network capability exposure in 3GPP TS 23.502 [27] to be:

- **Exposure of network events externally as well as internally towards core network NFs:** Network events such as user activity (e.g., mobility events, session establishment, or termination) or network status changes (e.g., load status, fault occurrences) are made accessible via standardized APIs to both external entities (like third-party applications) and internal NFs (e.g., AMF, SMF). This enhances coordination and responsiveness by enabling real-time access and reaction to network conditions.
- **Exposure of provisioning capability towards external functions:** Provisioning capabilities, such as user account setup, device configuration, or service activation and deactivation, are exposed to external functions via APIs. This allows third-party applications to interact with the network for resource allocation and service management, facilitating dynamic and automated service provisioning.
- **Exposure of policy and charging capabilities towards external functions:** Policy control mechanisms, which include QoS enforcement, access control, and usage monitoring, as well as charging functions like billing records generation and real-time charging, are made available to external entities. This exposure enables external applications to influence network behavior and monetization strategies, allowing for customized and differentiated service offerings.
- **Exposure of core network internal capabilities for analytics:** Core network capabilities, such as traffic management, performance monitoring, and user experience metrics, are shared internally within the network functions, mainly between NWDAF and NEF. This exposure supports comprehensive

analytics processes, enabling network optimization, anomaly detection, and predictive maintenance through deep insights into network operations.

- **Exposure of analytics to external party:** Analytical insights derived from internal network data, including traffic patterns, usage statistics, and performance metrics, are made accessible to external parties. This data helps external entities, such as service providers and application developers, to understand network behavior, optimize their services, and enhance user experience by leveraging network analytics.
- **Retrieval of data from external party by NWDAF:** The Network Data Analytics Function (NWDAF) retrieves data from external sources, such as third-party service platforms, IoT devices, or user equipment, to augment its analytics capabilities. This integration of external data allows for a more holistic analysis, combining internal network metrics with external inputs to improve decision-making and network management.

Detailed explanation regarding external network capability exposure is in Section 4.1.1.

Within the 5G core network, the main NF responsible for network capability exposure is the Network Exposure Function (NEF). 3GPP TS 23.501 [20] defines it to support the following functionality:

- **External Exposure of capabilities and events:** NF capabilities and network events (such as service availability or status changes) can be securely exposed to third-party entities (e.g., AFs, Edge Computing platforms) via the NEF. This allows these external entities to access and utilize network features and information, enhancing service integration.
- **Secure provision of information from external application to 3GPP network:** External Application Functions can securely provide information to the 3GPP network through the NEF. This includes data such as Expected UE Behavior, 5G Virtual Network (VN) group information, time synchronization service details, and other service-specific information. For example, a drone management application might send Expected UE Behavior data to the 3GPP network, detailing the typical movement patterns of drones to optimize network resource allocation and connectivity. The NEF ensures secure communication by authenticating, authorizing, and potentially throttling the data flow from the Application Functions to maintain network security and performance.
- **Translation of internal-external information:** The NEF translates between information exchanged with external AFs and the internal network functions. For example, it maps an AF-Service-Identifier to internal 5G Core identifiers such as Data Network Name (DNN) and Single Network Slice Selection Assistance Information (S-NSSAI). Additionally, the NEF masks sensitive network and user information according to network policies before exposing it to external AFs, ensuring data privacy and security.
- **Redirecting the AF to a more suitable NEF/Local-NEF:** The NEF can redirect an AF to a more appropriate NEF or Local-NEF instance when it detects that another instance is better suited to handle the AF's request for local information exposure. This ensures that AF requests are efficiently and optimally processed.
- **Internal Exposure:** The NEF collects information from other network functions based on their exposed capabilities and stores this data as structured information in a Unified Data Repository (UDR) using standardized interfaces. This centralized storage allows for efficient data management and retrieval.
- **Exposure of analytics:** Analytics generated by the NWDAF can be securely exposed to external

parties through the NEF. This enables third-party entities to leverage network analytics for enhanced decision-making, service optimization, and performance analysis.

- **Retrieval of data from external party by NWDAF:** The NWDAF can collect data from external parties via the NEF for the purpose of generating analytics. The NEF handles and forwards requests and notifications between the NWDAF and AFs, facilitating data exchange and integration for analytics.

In the broader context of network capability exposure, this process occurs in two primary directions: first, the exposure of network data such as location information and other capabilities to external AFs, and second, the provisioning of information from AFs to internal NFs for further orchestration and service configuration.

This bidirectional flow is facilitated by the Service-Based Architecture (SBA) of 5G, where NFs communicate through Service-Based Interfaces (SBIs) via APIs, enabling a modular and dynamic network. Within this framework, the NRF plays a crucial role by allowing NFs to register themselves and discover services provided by other NFs. Each NF, such as the NEF, registers with the NRF, providing information about its type and available services. When an NF needs to interact with another, it queries the NRF, which responds with details of available NF instances, enabling direct communication.

Moreover, NEF acts as a bridge between southbound interfaces, which connect internal network functions, and northbound APIs, which expose these services to external applications. SBA allows NEF to seamlessly facilitate this exposure by dynamically discovering services and updating NFs as needed, enabling real-time adaptability and programmability. This is particularly important for supporting use cases like network slicing and edge computing, where NEF needs to expose network capabilities to third-party AFs in a secure and controlled manner.

By leveraging SBA's modular framework and service discovery capabilities, NEF can efficiently manage the exposure of network resources while ensuring secure and scalable interactions with both internal and external functions. This adaptability is critical to supporting the dynamic nature of 5G services, as network capability exposure is fundamental to both internal and external operations.

Network capability exposure in a 5G network can be categorized into external and internal exposure, each serving different purposes but unified by the SBA's flexible architecture.

In the 5G core network, internal capability exposure refers to the ability of NFs to communicate and exchange functionalities and parameters with other NFs or external entities via standardized interfaces and protocols. This capability allows NFs to programmatically expose their functionalities, enabling dynamic interaction and orchestration. The NEF plays a pivotal role in this process by communicating with other NFs through various southbound interfaces, as defined in 3GPP TS 29.522 [9]. These interfaces enable the NEF to exchange information with core network functions such as the AMF, SMF, PCF, and UDR, ensuring seamless and secure interaction within the network's core.

This internal exposure of capabilities brings the following advantages to the 5G network:

- Dynamic Interaction: NFs, such as those responsible for subscriber authentication, can seamlessly share their capabilities through APIs like RESTful APIs, HTTP/2, or gRPC. This enables functions like policy control or traffic steering to request authentication services as needed.
- Support for Network Slicing: Internally exposed capabilities are crucial for features like network slicing, where NFs must reveal slice-specific parameters such as QoS requisites, network function instantiation policies, or security configurations. This allows the orchestrator to dynamically instantiate and manage slices according to service demands.

- Efficient Resource Management: Internal exposure streamlines resource management by allowing NFs to provide real-time information about their resource utilization and requirements. This enables the orchestrator to optimize resource allocation, ensuring the efficient utilization of network resources.

Figure 4.1 illustrates the following key points:

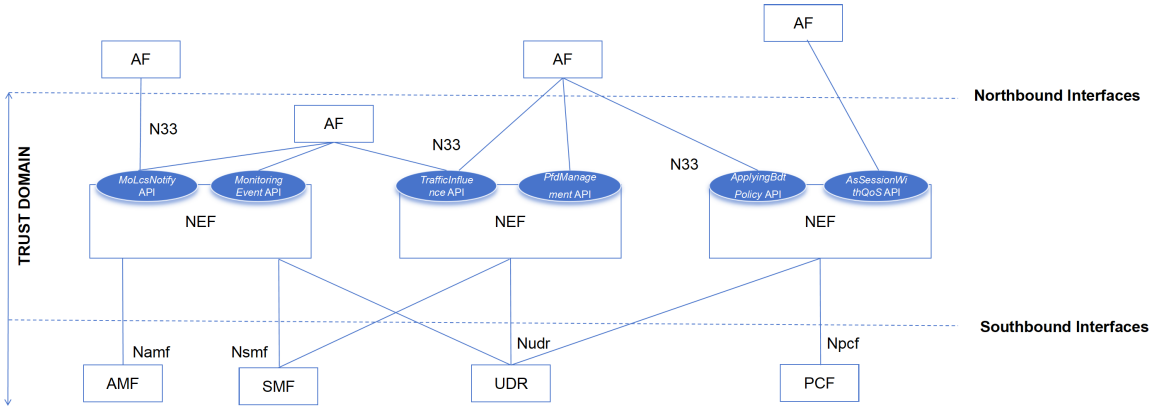


Figure 4.1: NEF Architecture in a Service-Oriented Environment

- Trust Domain: The NEF operates within a trust domain, ensuring secure interactions with external AFs.
- Northbound Interface (N33): The N33 interface is used by the NEF to expose various network capabilities as APIs to AFs. This allows AFs to interact securely with specific services and data within the 5G core network, enabling third-party applications to access network functions like session management, quality of service control, and location information.
- Southbound Interfaces: The NEF communicates with internal NFs like AMF, SMF, PCF, and UDR to retrieve and manage network data.

As shown in Figure 4.1, the NEF acts as the anchor in the capability exposure process, exposing APIs through the N33 northbound interface to internal and external AFs, which represent third-party companies or affiliations for telecom operators [12]. The NEF ensures that external applications can interact securely and efficiently with the 5G core network, providing capabilities such as monitoring events, managing traffic influence, applying policies, and handling session QoS.

The presence of multiple NEFs in Figure 4.1 reflects the distributed nature of the 5G architecture, where multiple NEF instances can be deployed to handle different aspects of network exposure, such as monitoring, policy application, and traffic management. This distribution allows for load balancing, redundancy, and specialized handling of different types of requests, enhancing the overall efficiency and robustness of the network.

The specific process of 5G capability exposure, primarily managed by NEF, is as follows: NEF receives information from other NFs based on their capabilities. Subsequently, if storing is required, NEF stores the received information as structured data using a standardized interface (Nudr) to a UDR. This stored information can then be accessed and 're-exposed' by NEF to other NFs and to AFs.

An example of internal exposure would be the *Nupf_EventExposure* service, which enables the UPF to share UPF-related details with service consumers such as the NEF. Figure 4.2 illustrates this process. The procedure of *Nupf_EventExposure* for reporting QoS Monitoring can be outlined as follows [28] [29]:

1. An AF initiates the setup of an AF session along with the required QoS procedure. In this request, the AF may subscribe to direct notification of QoS monitoring for the service data flow to the PCF, through the NEF. If such a subscription is made, the AF must include the corresponding QoS monitoring parameters.
2. Upon subscription to direct notification of QoS monitoring, the PCF incorporates the indication of direct event notification (including the target NEF address) for the service data flow within the Policy Control and Charging (PCC) rule. The PCF then sends the indication of direct event notification to the SMF.
3. Upon receiving the indication from the PCF, and determining that the UPF supports such reporting, the SMF instructs the UPF to report QoS Monitoring events to a NEF via the N4 interface.
4. Upon the occurrence of the specified event, the UPF notifies the SMF via the N4 interface. The SMF then forwards this information to the NEF using the Nsmf interface. This sequence allows the NEF to receive event notifications from the UPF indirectly, enabling it to expose relevant data to external applications if needed.
5. Upon success, the NEF responds with ”204 No Content”.

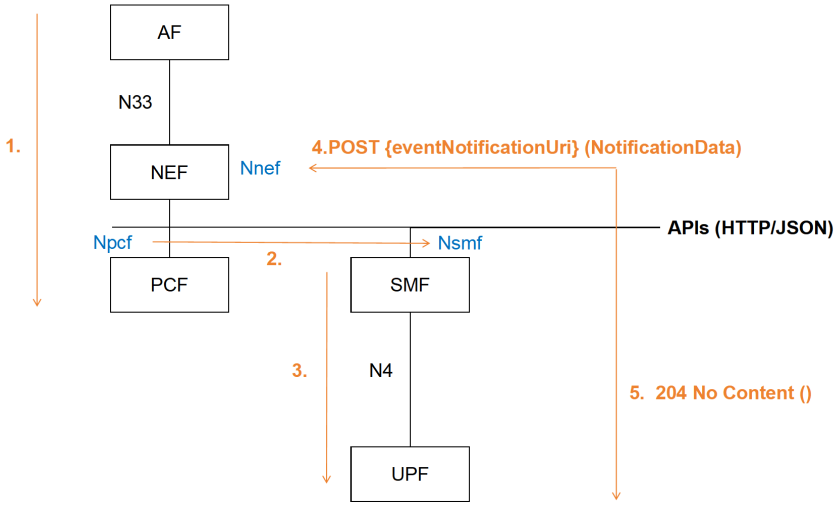


Figure 4.2: Process of *Nupf_EventExposure* Service

These event notifications may encompass details such as a QoS Monitoring report, furnishing information such as end-to-end delay for a specific QoS flow within a PDU session.

4.1.1. External Exposure of Network Capabilities

According to 3GPP TS 23.501 [20], the NEF supports the external exposure of network function capabilities, which can be classified into four types: monitoring, provisioning, policy/charging, and analytics reporting. These categories encompass a range of functionalities that contribute to the flexibility, adaptability, and optimized operation of the 5G network, aligning with the diverse requirements of emerging applications and services. Table 4.1 provides an overview of the service classification, the associated relationships with SCEF and NEF, as well as references to relevant documents.

Monitoring Capability

In 3GPP TS 29.122 [6], several monitoring events are introduced to enhance network functionality and enable efficient application interactions:

Northbound API	Type of Capability	Related 3GPP Document
MonitoringEvent	Monitoring	TS 29.122
CpProvisioning	Provisioning	TS 29.122
NpConfiguration		
ECRControl		
RacsParameterProvisioning		
ServiceParameter		TS 29.522
5GLANParameters		
LpiParameterProvision		
TrafficInfluence		
AnalyticsExposure	Analytics	TS 29.522
AsSessionWithQoS	Policy/Charging	TS 29.122
ChargeableParty		
ResouremanagementOfBdt		
PfdManagement		
ApplyingBdtPolicy		TS 29.522

Table 4.1: Available Northbound APIs for External Exposure [30]

Location Reporting: This event provides the location of a UE, encompassing either the Current Location or the Last Known Location.

UE Reachability: Used when applications need to discern when a UE becomes reachable after an extended power-saving sleep cycle. The NEF informs the application of the UE's availability, enabling the application to transmit data intended for downlink transmission.

Loss of Connectivity: Notifies an application when the UE experiences a loss of connectivity, such as detaching from the 3GPP Network or not communicating with the network after a predefined time.

Change of IMSI-IMEI Association: Notifies an application when the International Mobile Subscriber Identity (IMSI) of the UE is suddenly associated with a different device.

Number of UEs in a Geographic Area: Indicates the count of UEs in a specific geographical area, allowing applications to monitor the number of devices within a particular location.

Packet Data Network (PDN) Connectivity Status: Detected when a PDU session is established or released.

Downlink Data Delivery Status: Indicates the status of downlink data delivery in the core network, reporting events at the initial occurrence of packet buffering, transmission, or discarding.

Availability after Downlink Data Notification Failure: Triggered when the UE becomes reachable again after a downlink data delivery failure, informing the application about the device's availability.

Provisioning Capability

Besides facilitating the external exposure of 5G network capabilities, the NEF also enables applications to transmit application-related information to the network. For instance, in V2X communications, vehicles use the NEF as the entry point to share destination and route details with NFs such as the PCF, SMF, and UPF. This data exchange is crucial for optimizing processes like handovers between base stations, ensuring seamless connectivity as vehicles move.

By receiving this information, the NEF interacts with the BSF (Bootstrapping Server Function) to retrieve the associated PCF information by invoking the *Nbsf_Management_Discovery* service, as defined in 3GPP TS 29.521 [31]. Once the PCF is identified, the NEF interacts with it using the *Npcf_PolicyAuthorization* service (3GPP TS 29.514 [32]) to manage the traffic policies. These policies are applied based on the vehicle's route and destination, enabling the network to optimize handovers and manage resources effectively.

The PCF then plays a critical role in adjusting traffic policies dynamically as the vehicle moves through the network. It predicts the vehicle's trajectory, determines how traffic should be routed, and ensures the appropriate QoS is applied to the data flows. The SMF, in coordination with the UPF, handles session management and user plane resource allocation based on the policies provided by the PCF. These resources are pre-allocated at upcoming base stations before the vehicle reaches them.

When the handover is imminent, the network triggers a pre-configured handover (PCH), where the target base station is prepared in advance. The resources are allocated, and the connection context is transferred ahead of time, ensuring that the handover is smooth and fast, minimizing service disruption. This proactive approach enhances both network reliability and the user experience by reducing latency during handovers.

Additionally, the NEF, along with the N33 interface, provides APIs for applications to update parameters configured in the 5GC, such as QoS and traffic routing. For example, applications can use the *CpProvision* API to modify QoS settings, allowing them to prioritize traffic types like video streaming or online gaming. This capability is particularly beneficial for MEC, where traffic can be offloaded to local edge servers, significantly reducing latency and improving performance for applications like AR or real-time video conferencing.

The NEF's ability to influence network behavior through APIs like *Npcf_PolicyAuthorization* allows applications to provide the network with information about expected communication patterns. This helps the network optimize its operations, reducing unnecessary UE state transitions and improving overall efficiency, as previously illustrated.

In conclusion, the NEF's provisioning capability empowers external applications to interact with 5G NFs securely and effectively. This process involves retrieving necessary policy and traffic information from the PCF through the BSF, as well as receiving, storing, and distributing critical external information across relevant NFs such as the PCF, SMF, and UPF. By enabling this flow of information, the 5G network can dynamically adjust to changing conditions and optimize service delivery.

Policy/Charging Capability

The NEF extends its services to include policy and charging capabilities within the 5G Core, managed by the PCF. The PCF enables mobile operators to enforce enhanced QoS and charging control, which is crucial in wireless networks where efficient utilization of radio resources is essential. Given the diverse QoS requirements of various services and the dynamic nature of network conditions, the *AsSessionWithQoS* API plays a vital role. This API allows applications to establish sessions with specific QoS parameters, such as latency and priority, for a particular IP traffic flow.

The QoS monitoring process is an example of this capability, wherein an AF interacts with the NEF to request and monitor QoS parameters. The sequence is illustrated in Figure 4.3 and unfolds as follows:

Request Initiation: The AF sends a Nnef_AFSessionWithQoS_Update Request to the NEF, specifying the desired QoS parameters for the session.

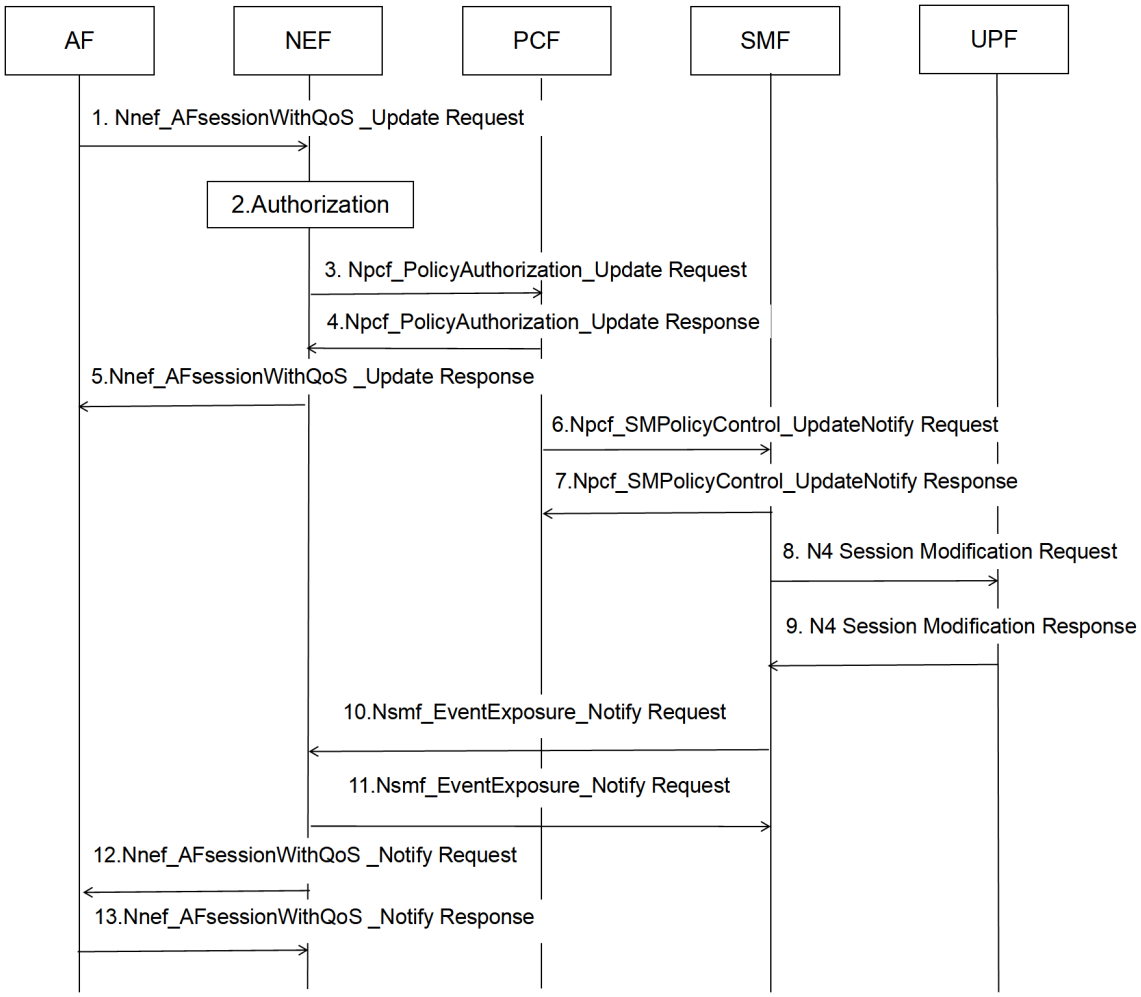


Figure 4.3: QoS Monitoring Process [33] modified

Authorization: The NEF checks the request and forwards it through the Npcf_PolicyAuthorization_Update Request to the PCF. The PCF evaluates the request based on the existing policy rules and responds with the Npcf_PolicyAuthorization_Update Response.

Session Establishment: After authorization, the NEF sends a response back to the AF, confirming the QoS parameters through the Nnef_AFSessionWithQoS_Update Response.

Policy Enforcement: The PCF communicates the updated policy control to the SMF via the Npcf_SMPolicyControl_UpdateNotify Request. The SMF then sends an N4 Session Modification Request to the UPF to enforce the QoS settings. The UPF acknowledges this with an N4 Session Modification Response.

QoS Monitoring: The SMF then initiates QoS monitoring by requesting both the UPF and NG-RAN to continuously monitor the QoS for the session. If the QoS cannot be guaranteed or if there are significant changes in the network conditions, the SMF notifies the PCF and AF through the Nsmf_EventExposure_Notify Request.

Moreover, the AsSessionWithQoS API can provide real-time insights to applications regarding the likelihood of guaranteeing the desired QoS in the near future, considering network fluctuations.

Furthermore, the ChargeableParty API offers applications the capability to communicate their intention to initiate or terminate sponsorship for a specific traffic flow. Another important API is the PfdManagement API, which allows applications to furnish Packet Flow Descriptors (PFDs) to the Packet Flow Description

Function (PFDF) in the 3GPP network. These descriptors aid in detecting specific traffic types and applying relevant PCC rules to the identified flows.

In certain scenarios, applications have foreknowledge about the data volume they need to exchange with multiple UEs within a specific geographical area. For instance, an IoT agriculture application may need to send weather forecast information to one hundred sensors during the night. In such cases, the *ApplyingBdtPolicy* and *ResourceManagementOfBdt* APIs, exposed by the NEF, prove valuable. These APIs allow the application to communicate essential data transfer requirements—such as the number of UEs, data per UE, and time constraints—to the 5G network. The PCF then uses this information to formulate policies that determine the optimal timing and conditions for data transmission, ensuring efficient network resource usage.

Analytics Reporting Capability

The Analytics Exposure capability of the 5G Core network incorporates various elements designed to facilitate the discovery and utilization of analytics information by external entities, primarily through the NWDAF [20]. Serving as a network-aware entity interfacing with multiple 5GC network functions, the NWDAF collects relevant data and events for analysis. While the NWDAF handles the necessary functionalities for these tasks, the Analytics Exposure API, integrated with the NEF, is responsible for exposing the gathered data to applications.

This capability is particularly crucial in vertical use cases within industrial automation. 3GPP has precisely defined communication standards tailored for future factories, delineating application areas and mapping applications such as motion control, massive wireless sensor networks, augmented reality, process automation, connectivity for the factory floor, and inbound logistics for manufacturing.

In such scenarios, data analytics ensures network availability and provides features like predictive maintenance. The analytics framework encompasses operations like efficient QoS management, traffic steering, and mobility management. Notably, in Release 16, 3GPP introduced the concept of network prediction, allowing the 5G system to notify a V2X application about potential downgrades in UE communication QoS due to factors such as predicted poor network conditions or radio congestion. Analytics significantly contributes to harmonizing the 5GC network, enabling applications to achieve higher service availability and quality.

Applications can leverage the following types of data provided through the Analytics Exposure feature:

- Network performance analytics: Data to perform network optimization.
- UE communication analytics: Data to predict communication patterns, aiding in optimizing operations such as traffic routing and QoS improvements.
- UE mobility analytics: Data for location predictions and anomaly detection, helping to alert applications to unexpected changes in behavior.
- QoS sustainability analytics: Information regarding QoS change statistics, helping to manage and sustain quality of service.

4.2. Network Exposure in Radio Access Network

5G applications, such as Cloud VR, are characterized by high bandwidth consumption and sensitivity to latency. Nevertheless, existing semi-static QoS frameworks may prove inadequate in meeting the diverse QoE requirements of these traffic-intensive and highly interactive applications, particularly when considering

potential fluctuations in radio transmission capability. The utilization of QoE estimation/prediction is anticipated to address such uncertainties, enhance radio resource efficiency, and ultimately elevate the user experience. The exposure of RAN analytics information (RAI) as a RAN service to a specific UE or to groups of UEs (e.g., per slice, per cell, per PLMN) is envisioned to be instrumental in enhancing the user experience for these applications [34]. The detailed procedure of O-RAN Analytics Exposure will be introduced in Section 4.2.4.

4.2.1. 3GPP RAN Standards and O-RAN Standards

Before introducing the architecture and specific elements that compose the O-RAN architecture, this subsection begins by comparing the differences between 3GPP-standardized RAN and O-RAN.

O-RAN is built upon the foundation set forth by 3GPP to clearly define requirements for an open, virtualized, and interoperable RAN. The O-RAN Alliance's architecture extends the 3GPP's RAN architecture to further disaggregate the radio access network and open it up to multiple vendors. As such, the O-RAN architecture not only disaggregates many components found in the 3GPP architecture but also introduces new functions such as the RAN Intelligent Controller (RIC) and Service Management and Orchestration (SMO), frontHaul split, and new interfaces, as illustrated in Figure 4.4.

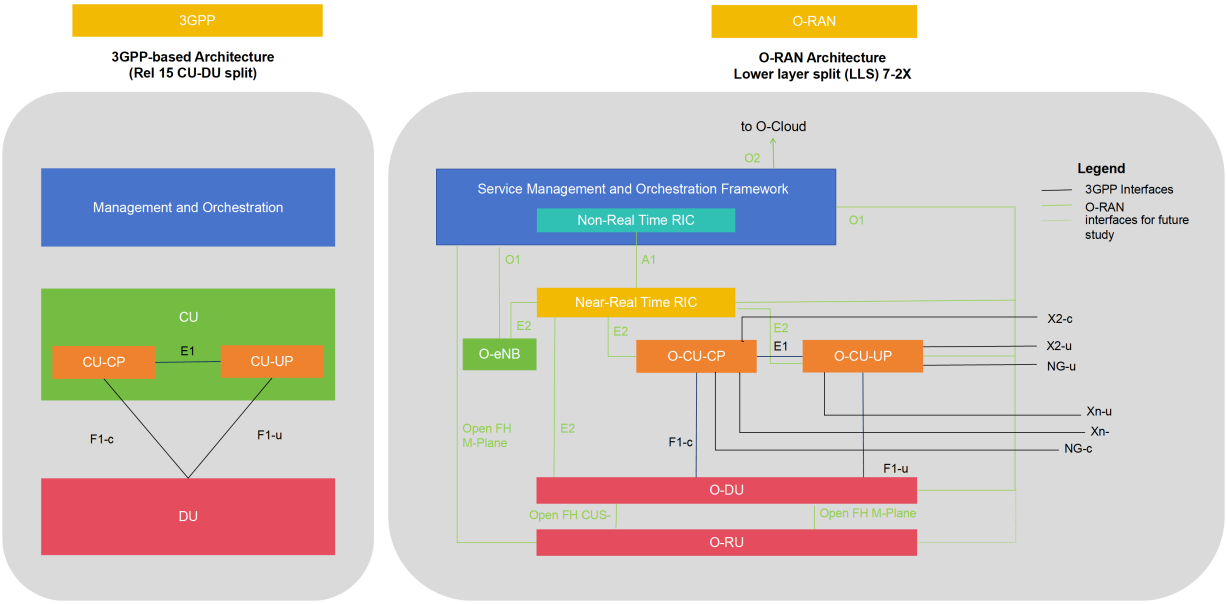


Figure 4.4: 3GPP RAN vs O-RAN architecture [35] [34] modified

In 4G and 5G architectures, the RAN consists of base stations connected to the core network through a backhaul network. Base stations manage RF processing and modulation, while the core network handles control and signaling functions.

In common, both RAN architectures are divided into two main components: the Central Unit (CU) and Distributed Unit (DU). In O-RAN architecture, the CU handles high-level functions such as control and signaling, while the DU manages low-level tasks like radio frequency processing. An open interface connects the CU and DU, allowing different vendors' equipment and software to be used in the RAN. This modular approach enhances flexibility and interoperability in the RAN.

Figure 4.4 illustrates the architecture of 3GPP RAN and O-RAN. The differences between 3GPP RAN and O-RAN in terms of functions and interfaces are shown in Table 4.2. Overall, O-RAN complements

3GPP RAN by offering more modularity and flexibility, allowing the use of equipment and software from diverse vendors in the RAN.

3GPP RAN	O-RAN
Functions: 1. Management and Orchestration 2. Central Unit - Control Plane/Central Unit - User Plane 3. DU	**Additinal Functions:** 1. Service Management and Orchestration layer 2. Non-Real-Time RIC 3. Near-Real-Time RIC
Interfaces: 1. E1 2. F1-C/F1-U	**Additional Interfaces:** 1. A1 2. E2 3. O1 4. O2 5. Open Fronthaul

Table 4.2: Difference in Functions and Interfaces Between 3GPP RAN and O-RAN [36]

4.2.2. O-RAN Components

Figure 4.5 illustrates the components of O-RAN, and their respective functions are explained below [37] [38] [39].

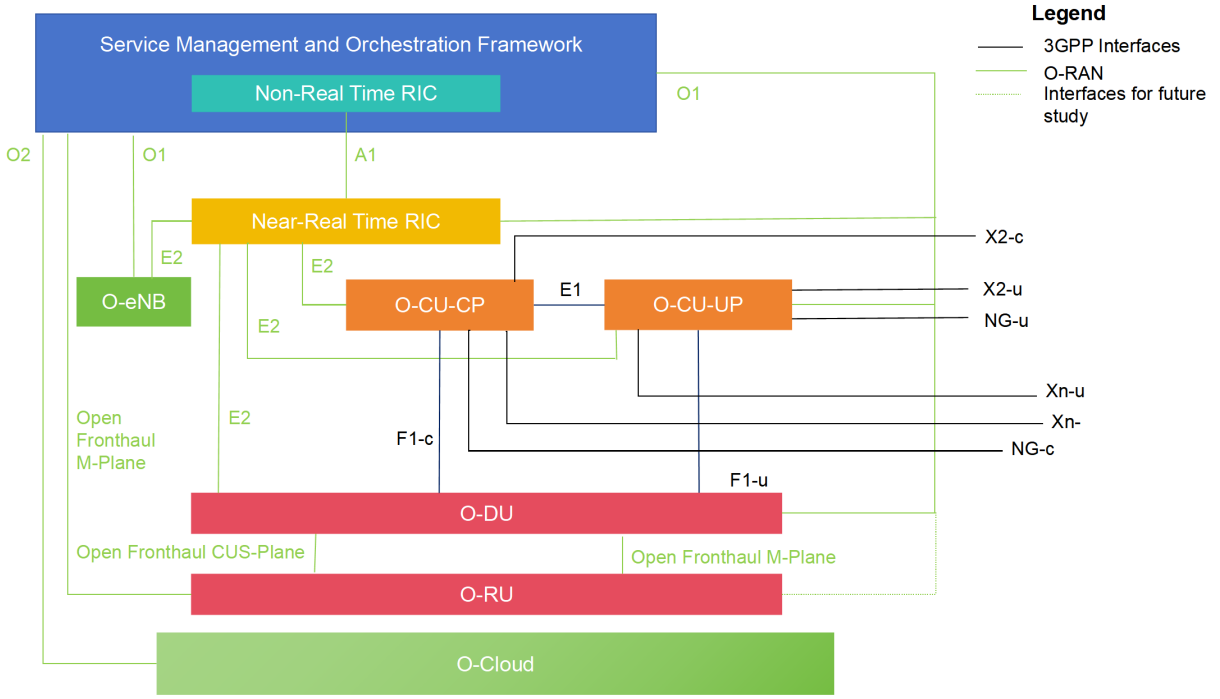


Figure 4.5: O-RAN architecture overview [40]

Near-Real-Time RAN Intelligent Controller (Near-RT RIC): This logical function facilitates near-real-time control and optimization of O-RAN elements and resources through fine-grained data collection and actions over the E2 interface. Additionally, the near-RT RIC hosts cloud-native microservice-based applications, known as xApps.

Non-Real-Time RAN Intelligent Controller (Non-RT RIC): This logical function facilitates non-real-time

control and optimization of RAN elements and resources, as well as managing AI/ML workflows, including model training and updates. Additionally, it provides policy-based guidance for applications/features in near-RT RIC. Operating from within the RIC's SMO platform, the Non-RT RIC oversees lifecycle management for network elements, configuration management, and other crucial network functions. It optimizes RAN functions by delivering policy-based guidance, managing models, and providing enrichment information to the near-RT RIC function.

SMO: A layer that manages the components and network functions of an open RAN. This layer includes the non-RT RIC.

O-Cloud: A cloud platform that hosts the relevant O-RAN functions (i.e., Near-RT RIC, O-CU-CP, O-CU-UP, and O-DU) as all these functions can be virtualized and deployed using software on this hardware, which is O-Cloud.

xAPP: An independent software plug-in for the Near-RT RIC platform that enhances and extends the functionality of the RAN. xApps offer specific capabilities such as real-time data analytics, dynamic resource management, mobility optimization, and interference mitigation. Developed by third parties, xApps operate within the Communication Service Provider's (CSP) environment, providing customized control and optimization of network operations. They enable near-real-time decision-making, which enhances overall network performance and efficiency.

vAPP: A specialized application designed to operate within the O-RAN framework, typically running on the Near-Real-Time RIC (RAN Intelligent Controller). Leveraging advanced features such as network slicing, dynamic resource allocation, and low-latency communication, vApps make use of O-RAN's open interfaces and APIs to interact with network functions, enabling functionalities tailored to specific industry needs. Through real-time data processing and analytics, vApps facilitate optimized network operations for sectors like healthcare, transportation, and smart cities, ensuring efficient resource utilization and improved service delivery. The modular design of O-RAN supports seamless vApp integration, enhancing the adaptability and programmability of the RAN environment.

4.2.3. Entities/Resources Involved in the Exposure Functionality

In the O-RAN architecture for analytics exposure, illustrated in Figure 4.6, the radio side consists of the Near-RT RIC, O-CU-CP, O-CU-UP, O-DU, and O-RU. On the management side, the Service Management and Orchestration (SMO) Framework includes the Non-RT RIC function. The Y1 service interface enables Y1 consumers to subscribe to or request RAN analytics data provided by the Near-RT RIC. Y1 consumers are typically entities within the trusted domain of a Public Land Mobile Network (PLMN), such as applications within the SMO or other authorized analytics tools. These consumers leverage the data to optimize network performance, monitor RAN conditions, or make informed decisions about resource allocation and configuration adjustments.

The following is a list of specific entities involved in network capability exposure within O-RAN [42]:

1. **Non-RT RIC**
    - QoE Metrics Retrieval and AI/ML Model Construction: The Non-RT RIC retrieves QoE-related measurement metrics from network-level reports and the SMO, which can include application data. These metrics are used to construct and train AI/ML models that are later deployed in the Near-RT RIC for assisting in QoE optimization tasks, such as application classification, QoE prediction, and available bandwidth prediction.

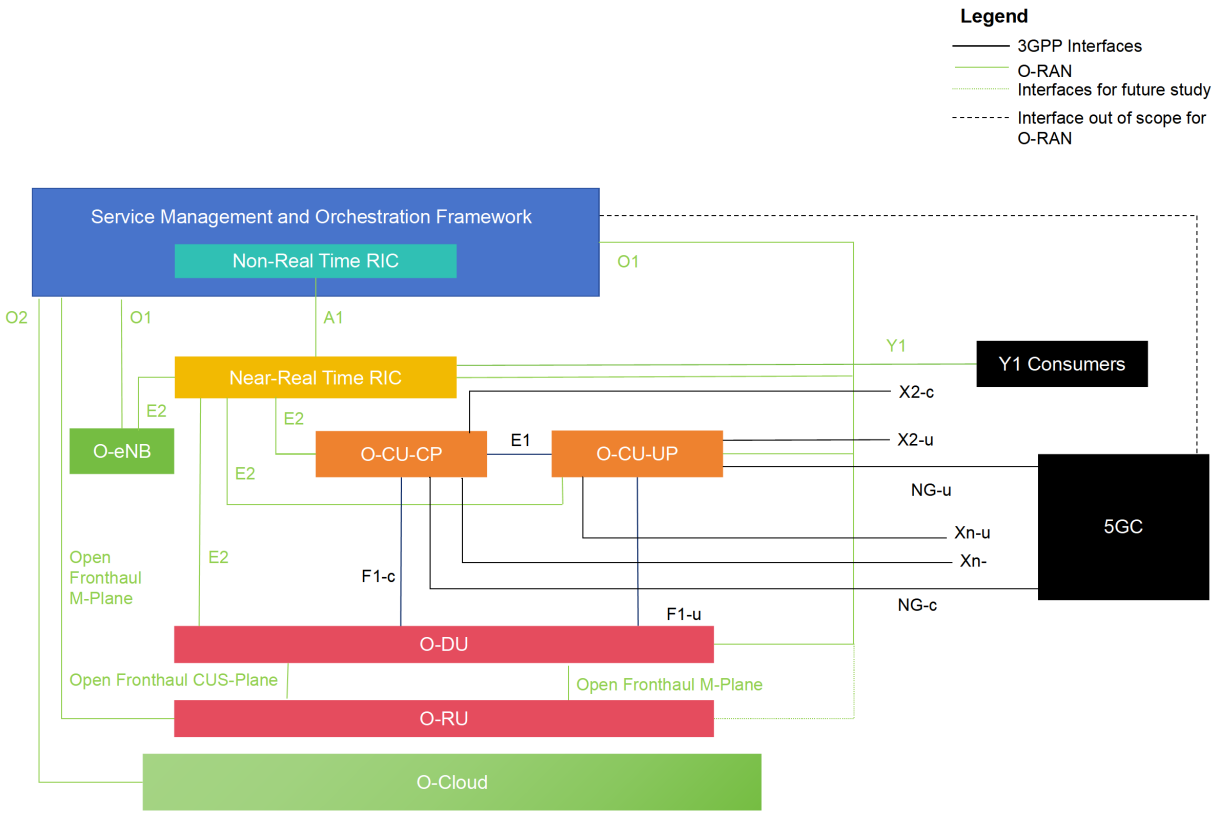


Figure 4.6: O-RAN Architecture for Analytics Exposure [41]

- ML Model Training for Predictive QoE Optimization: It trains ML models aimed at predictive QoE optimization. These models can autonomously recognize traffic types, predict the quality of experience, or estimate available radio bandwidth, enabling more effective resource management in the RAN.
- Policy and Intent Communication: The Non-RT RIC sends policies or intents to the Near-RT RIC, providing guidance for QoE optimization at the RAN level. These directives help shape expected behavior in line with QoE objectives.

2. **Near-RT RIC**

- Subscription and Request Handling: The Near-RT RIC supports the reception of request or subscription messages from RAN analytics information service consumers. This allows external applications and network functions to subscribe to or request specific RAN analytics data, enabling tailored and dynamic management of network resources.
- Network State and UE Performance Monitoring: It collects and processes real-time reports on network state and UE performance directly from the RAN. This data forms the basis for understanding current network conditions and the behavior of connected devices.
- RAN Analytics Information Exposure: It supports the exposure of inferred RAN analytics information to RAN Analytics Information (RAI) service consumers. This ensures that relevant stakeholders, such as network management systems or external applications, can access the insights generated by the Near-RT RIC to make informed decisions regarding network operation and service delivery.

- Data Analysis and AI/ML Execution: The Near-RT RIC performs advanced data analysis and executes AI/ML models [43] to infer critical RAN analytics information. This includes predictive analytics such as Quality of Experience (QoE) prediction and available bandwidth prediction, which are vital for optimizing network performance and user experience.

3. **Y1 Interface:** An interface through which the Near-RT RIC exposes RAN analytics services to Y1 consumers [42].

4.2.4. Exposure Procedure of O-RAN Analytics Information

O-RAN provides an example in [43] that describes the specific process of how analytics exposure is conducted. This use case centers on exposing RAN analytics to external applications or MEC platforms to optimize QoE for 5G applications like Cloud VR. It addresses the limitations of the current semi-static QoS framework by using multi-dimensional data (e.g., user traffic, QoE, and network reports) processed via ML algorithms to enable real-time QoE prediction and QoS enforcement. The objective is to support various QoE use cases within the O-RAN architecture through analytics exposure and an analytics-driven approach.

In this scenario, the purpose is to expose RAN analytics information to RAI service consumers for QoE optimization. RAN Performance analytics may be requested by the RAI service consumer using either a Request/Response solution or a Subscription-Based solution.

RAN Performance Analytics Process

This process outlines how RAN analytics information is exposed to external applications or MEC platforms to optimize QoE in 5G networks. The process is illustarted in Figure 4.7. The key actors involved are the Non-RT RIC, Near-RT RIC, SMO, and the application server or MEC. The process assumes that all relevant components are instantiated, and connectivity between the Non-RT RIC and other interfaces is established.

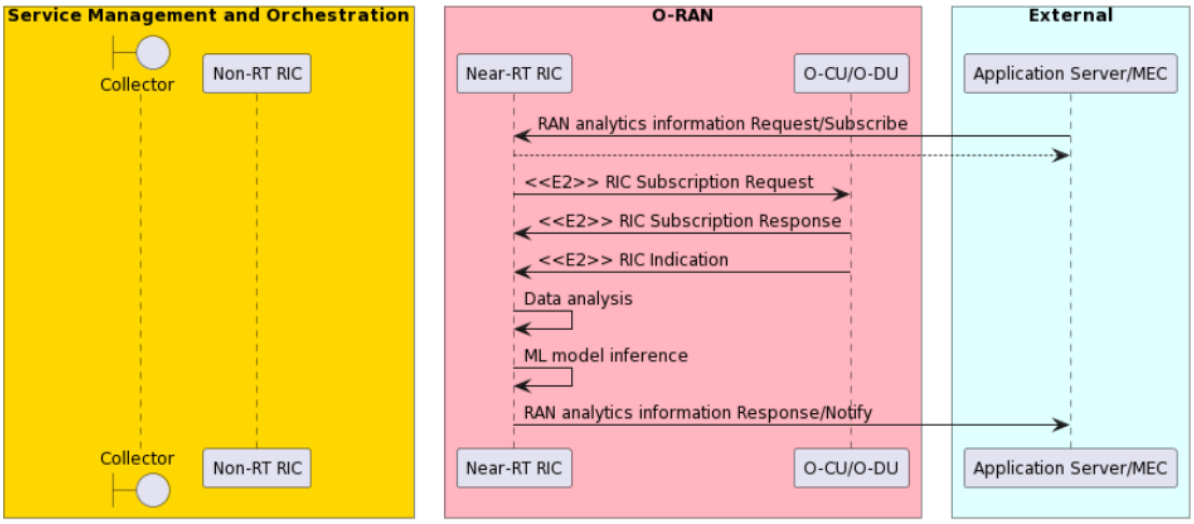


Figure 4.7: O-RAN Analytics Exposure upon Request [41]

The process begins when an application server or MEC requests or subscribes to RAN analytics information from the Near-RT RIC. Upon receiving this request, the Near-RT RIC subscribes to measurement data from O-CU/O-DU and utilizes QoE-related AI/ML models to infer RAN analytics. This information is

then exposed to the application server or MEC, either as a response or through event-triggered notifications. The process concludes when the application server receives the response or cancels the subscription. Post-process, the application server may execute logic controls, such as adjusting TCP transmission windows or video coding rates, to enhance QoE [43].

4.3. Network Exposure in Operator Platform

Operators in the 5G landscape aim to monetize their network capabilities by leveraging their existing relationships, local presence, and experience in delivering services. The challenge is in effectively standardizing and exposing these network capabilities across multiple operators. The Operator Platform (OP) addresses this need by providing a unified framework for network service exposure and management.

Given the diverse use cases operators must address, from healthcare to industrial IoT, a generic platform is essential. It enables operators to package both existing assets (e.g., voice, messaging, IP data services, billing, security, identity management) and new 5G capabilities (e.g., Edge Cloud, network slicing) to meet the evolving needs of enterprise customers.

To stay competitive against global players, operators aim to offer their assets consistently across networks and national boundaries. Thus, the GSMA envisions operators collaborating to provide a unified ”operator platform.” In Phase 1 of the Operator Platform [44], this platform will federate multiple operators’ edge computing infrastructure, granting application providers access to a global edge cloud through common APIs for running innovative, distributed, and low-latency services.

4.3.1. Introduction to the Operator Platform Concept

The Operator Platform concept, initially introduced by GSMA in [44], consists of several key components:

Application Provider: An entity that develops and deploys applications or services (e.g., cloud-based services, VR applications). Application providers rely on network resources and services to ensure the performance and delivery of their applications to end-users.

Aggregator: An intermediary that integrates multiple operators under a unified framework. The aggregator manages the coordination of network resources and capabilities across different operators within a specific region or across regions. For instance, Aggregator A might operate in Europe, coordinating with multiple operators in that region to provide consistent services to application providers.

Operator: The entity that owns and operates the physical and virtual network infrastructure, including 5G capabilities like edge computing, IP communication, and network slicing. Operators provide the network services and resources that application providers access through the Operator Platform.

Operator Platform (OP): A framework enabling operators to expose their network services and capabilities (such as edge cloud services and network slicing) to external application providers and other operators. Each operator maintains an independent instance of the OP, which includes interfaces for communication with other OPs within a federation (via the East-Westbound Interface), connection to its own network infrastructure (via the Southbound Interface), and interaction with application providers (via the Northbound Interface).

Application Providers utilize the Northbound APIs of the Operator Platform to request network services and capabilities, facilitating the deployment and management of their applications across multiple operator networks.

Aggregators oversee multiple operators within a region or across regions, facilitating the integration of network resources to ensure seamless deployment of applications across different operator networks.

Operators maintain individual OP instances, which encompass the required network services. These instances communicate through the East-Westbound Interface, enabling a federated network service environment.

Operator Platforms standardize the exposure of network capabilities across different operators, simplifying the process for application providers to leverage these capabilities without managing the complexities of interfacing with each operator individually.

Figure 4.8 illustrates this concept: Aggregator A coordinates services across Operator B and Operator C. Application Provider X, which operates in multiple regions, uses the Operator Platform managed by Aggregator A to deploy its services. Aggregator A, in turn, coordinates with the platforms of Operators B and C to ensure seamless delivery of Application Provider X's services to end-users across different geographic areas.

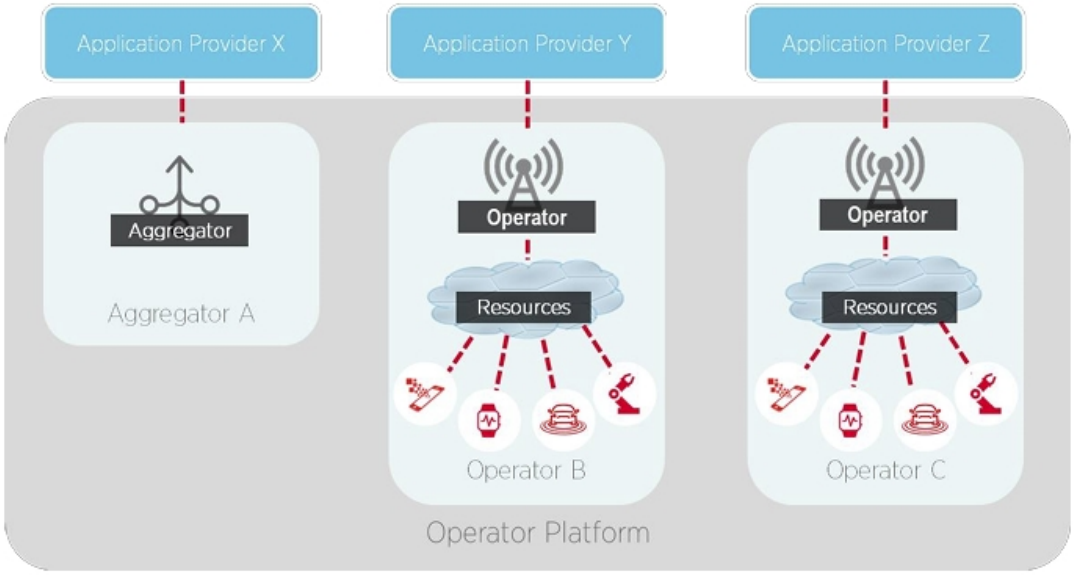


Figure 4.8: Operator Platform Example for Inter-Cloud Federation [44]

This architecture facilitates efficient exposure of network capabilities to third-party application providers while allowing operators to maintain control over their networks and promote collaboration among multiple operators within a federated framework.

Further whitepapers delve into specific topics such as edge services, associated commercial principles [45], technical requirements, and a provisional architecture [46].

4.3.2. Network Capability Exposure in the Operator Platform

The functional mapping between the Operator Platform and NEF in the 5G Core Network is illustrated in Figure 4.9. The interface connecting the Operator Platform and NEF is known as the Southbound Interface – Network Resources (SBI-NR) according to the Operator Platform's definition and is defined as N33 by 3GPP.

A crucial role within the Operator Platform is the Capabilities Exposure Role (CER). The CER enables

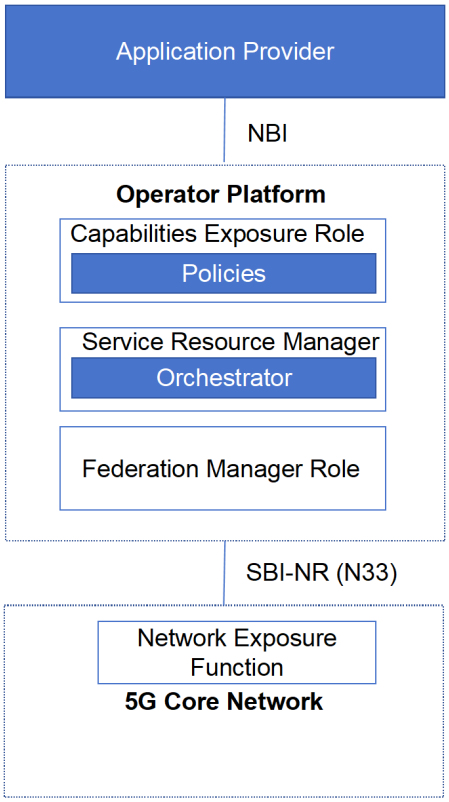


Figure 4.9: Functional mapping between OP and NEF [47]

Application Providers to effectively operate their applications by facilitating the discovery of Operator Platform capabilities. This includes functional aspects such as onboarding and instantiation procedures, as well as details on deployment locations and QoS attributes. The CER is responsible for exposing these capabilities to Application Providers via the NBI.

4.4. Network Exposure in CAMARA Project

CAMARA [48] is an open-source project under the Linux Foundation, focused on developing service APIs by integrating network APIs across telecom operator domains. CAMARA is specifically designed to create standardized, developer-friendly APIs that abstract the complexities of telecom networks, making them more accessible to developers, enterprises, and service providers [49]. Figure 4.10 shows an Open Gateway Network as a Service (NaaS) system architecture, highlighting the interplay between different API standards. It illustrates how CAMARA APIs influence both the third-party domain, where APIs are made available for external use, and the CSP domain, where these APIs are implemented based on the internal capabilities of network operators.

While NEF focuses on the technical aspects of exposing these network services, CAMARA goes a step further by abstracting these low-level network APIs into more simplified, service-oriented APIs. This abstraction is critical for making the APIs easier to consume for developers who may not have specialized telecom expertise. CAMARA also ensures that these APIs are compliant with data privacy and regulatory requirements, offering a higher level of accessibility and usability than NEF alone [48].

The GSMA Operator Platform is an initiative aimed at standardizing how operators expose their network capabilities. It provides the framework and guidelines for API exposure but does not create the APIs. CAMARA, on the other hand, collaborates with the GSMA Operator Platform Group (OPG) [51] to ensure

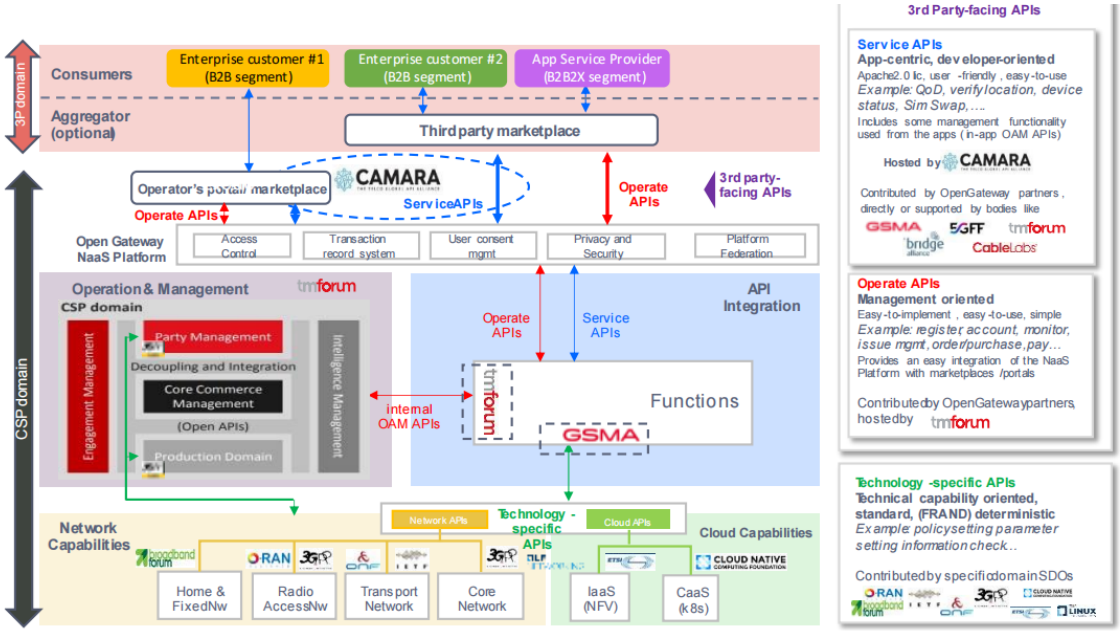


Figure 4.10: Different entities participating in the NaaS service standardization [50]

that the APIs it develops are aligned with industry standards.

Specifically, CAMARA collects API requirements from the GSMA OPG, which defines the reference architecture and requirements for the platform operators use to expose their capabilities to customers via APIs . This includes gathering a list of API families seen as useful for customers, functional descriptions of the APIs (attributes, functions, results), and non-functional requirements (such as response time, scalability, and performance) [52].

CAMARA's role is to operationalize these standards by developing and releasing the APIs that streamline telecom network complexity and ensure interoperability across various networks and countries.

4.4.1. CAMARA Architecture

Figure 4.11 illustrates the reference architectural framework of CAMARA. It comprises the following components [53]:

Network APIs: These APIs are integrated into telecommunications assets, encompassing network resources (core, access, and transport functions), cloud resources (virtualized and cloud-native workload hosting infrastructures), and IT resources (OSS and orchestration tools). Typically defined by standard bodies or industry forums, these APIs are closely tied to the underlying technology. Examples include those defined by 3GPP, ETSI, and TMForum, among others.

Service APIs: These NaaS APIs are intended for consumption by third-party applications, also known as external applications. They differ significantly from traditional Network APIs, which often require a deep understanding of complex telecom-specific parameters, such as those defined by 3GPP standards.

For example, using a Network API might require the application to specify a variety of low-level configuration parameters, such as specific 3GPP network settings. This level of detail can be challenging for developers who are not experts in telecom networks. CAMARA addresses this challenge by offering Service APIs that abstract away these complexities, providing a more user-friendly interface.

A practical example of this abstraction is the QoD (Quality of Delivery) API proposed by CAMARA.

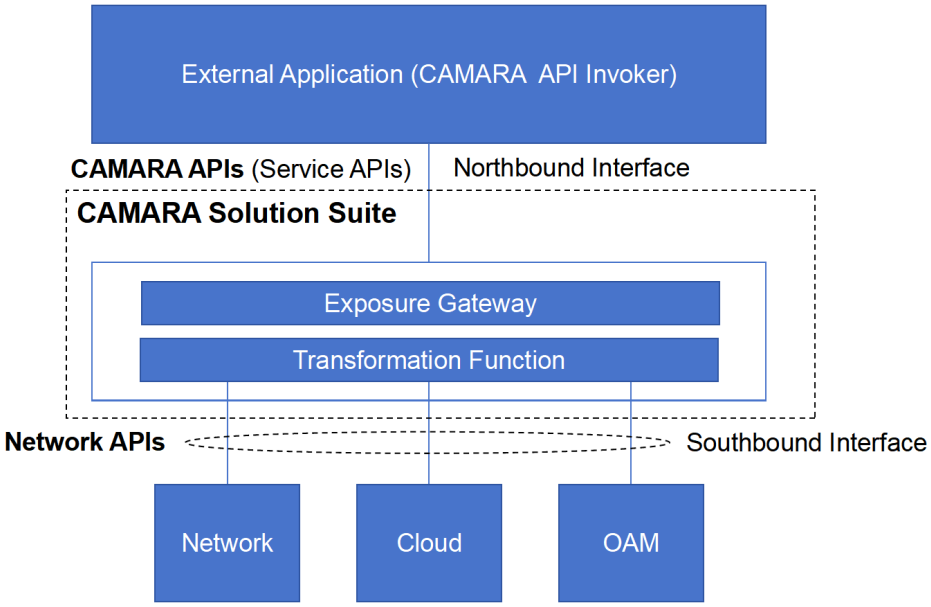


Figure 4.11: CAMARA Architectural Framework [53]

Instead of requiring the application to pass in detailed network parameters, the QoD API simplifies the process by allowing the application to specify only high-level, understandable inputs—such as endpoint sockets, transport protocol (TCP or UDP), and a session performance tag (e.g., ”best-effort,” ”throughput_L,” ”stable-latency”).

When the application invokes the QoD API, CAMARA’s solution suite automatically transforms this high-level API call into the corresponding low-level network API call, such as the *AsSessionWithQoS* API. This transformation involves translating the simple performance tags into the appropriate 3GPP network parameter values, which are then used to configure the network according to the desired quality of service.

Transformation Function: This component maintains the mappings between service APIs and network APIs and executes workflows to enforce these mappings. It can be deployed as a microservice equipped with a workflow engine.

API Gateway: This gateway provides essential capabilities for managing the interaction between the operator and external applications concerning service API invocation. These capabilities include service API publication and discovery, access control (authentication and authorization of applications), auditing, accounting, and logging.

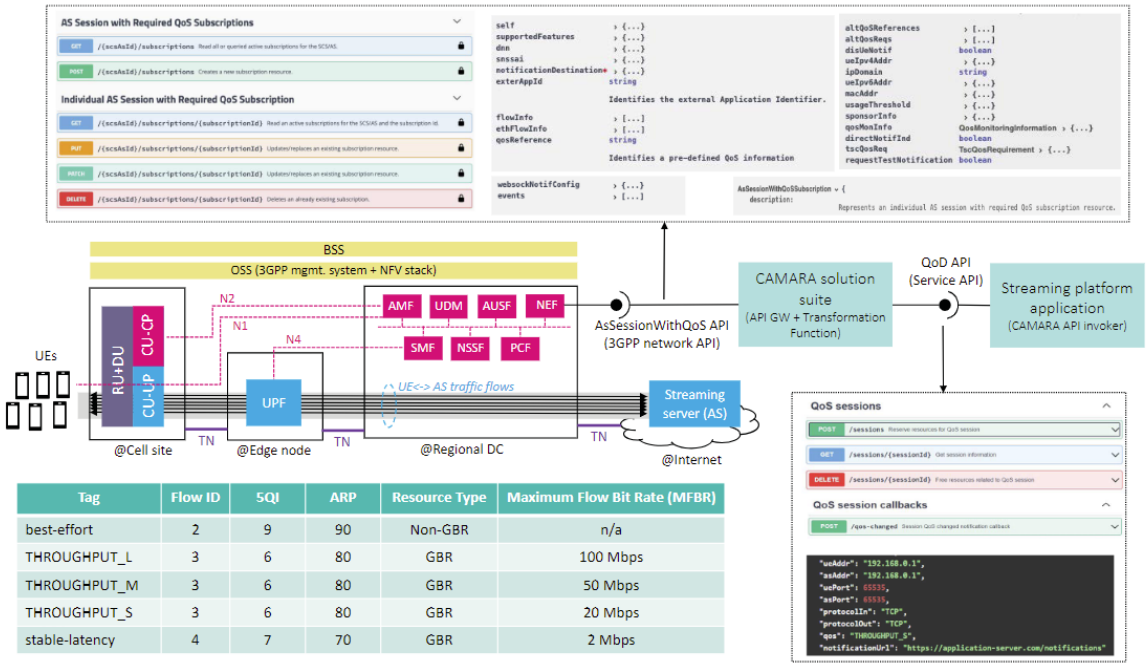
As depicted, CAMARA focuses on service APIs and their ability to build upon existing network APIs with the assistance of the transformation function and API Gateway.

4.4.2. Quality on Demand API

This subsection introduces the Quality on Demand (QoD) API as a practical example of how CAMARA APIs function. The QoD API is designed to allow application developers to manage specific QoS parameters for sessions between a UE and an AS. It also provides feedback from the network regarding the feasibility of granting the requested QoS parameters. The API enables the configuration and modification of key QoS attributes for mobile connections, which include:

Bandwidth: The user can select from different levels of throughput, namely large (”throughput_L”), medium (”throughput_M”), and small (”throughput_S”).

Latency: The user can enable stable latency, which aims to reduce jitter and provide near-deterministic data flow for the session.



Tag	Flow ID	5QI	ARP	Resource Type	Maximum Flow Bit Rate (MFBR)
best-effort	2	9	90	Non-GBR	n/a
THROUGHPUT_L	3	6	80	GBR	100 Mbps
THROUGHPUT_M	3	6	80	GBR	50 Mbps
THROUGHPUT_S	3	6	80	GBR	20 Mbps
stable-latency	4	7	70	GBR	2 Mbps

Figure 4.12: Application scenario for QoD API usage [53]

In the application scenario depicted in Figure 4.12, a third-party video streaming service utilizes the QoD API to enhance the QoS for its subscribers. The 5G standalone network facilitates connections between UEs and the application server, with the 5G core network and gNB providing the necessary coverage and control functions.

Traffic flows between UEs and the AS are tagged based on their specific performance requirements—such as ”best-effort,” ”throughput_L,” ”throughput_M,” ”throughput_S,” or ”stable-latency.” Each of these tags corresponds to specific 3GPP network parameters, which include:

- **5G Quality Indicator (5QI):** Determines the priority and packet delay budget for the traffic.
- **Allocation and Retention Priority (ARP):** Defines the priority for resource allocation and retention in congested conditions.
- **Guaranteed Bit Rate (GBR) or Non-Guaranteed Bit Rate (Non-GBR):** Specifies whether a minimum bit rate is guaranteed for the session.
- **Maximum Flow Bit Rate (MFBR):** Sets the upper limit for the bit rate of a flow.

These parameters are configured in the network through the NEF using the *AsSessionWithQoS* API.

To maintain an optimal user experience, especially during periods of network congestion, the streaming service’s traffic management application dynamically adjusts the QoS settings. It does so by modifying the tags assigned to UE↔AS sessions to prevent performance degradation.

Instead of directly using the complex *AsSessionWithQoS* API—which requires a deep understanding of 3GPP network parameters—the application opts for the more intuitive QoD API. The QoD API simplifies the interaction by only requiring the following inputs:

- **Endpoint Sockets:** Identifying the UE and AS involved in the session.

- **Transport Protocol:** Specifying whether the session uses TCP or UDP.
- **Session Performance Tag:** Indicating the desired QoS level.
- **Callback URL:** For receiving network feedback and notifications.

When the QoD API is invoked, CAMARA's solution suite translates this high-level request into a corresponding *AsSessionWithQoS* API call. This translation involves mapping the session performance tag to the appropriate 3GPP network parameters, thereby ensuring that the 5G network is configured to meet the requested QoS specifications.

4.5. Conclusion

In this chapter, network capability exposure within four key standards is explored: 3GPP core network, O-RAN architecture, GSMA's Operator Platform, and the CAMARA Project.

Within the 5G core network, the NEF is the main element in enabling network capability exposure. It facilitates the exchange of information between external AFs and internal NFs, tailored to the specific exposure requirements.

In the O-RAN architecture, network capability exposure primarily involves the analytics information exposed by the Near-RT RIC. This enhances the user experience for UEs by leveraging analytics-driven insights.

The GSMA's Operator Platform offers a uniform platform for packaging existing and new assets and capabilities. This approach aims to provide flexibility for modern enterprise customers by offering a wide range of services, including Edge Cloud and network slicing.

The CAMARA Project introduces a suite of service APIs designed to simplify telco complexity and make APIs accessible to customers without telco expertise. These user-friendly APIs aim to simplify the consumption of telco services.

These standards enable seamless access to network resources and services, facilitating efficient communication among various network elements. Additionally, robust security measures are crucial for maintaining the integrity of network data and resources across these standards.

As stakeholders continue to leverage and refine these capabilities, we anticipate further innovation and evolution in network architectures and services.

5

Controlling User Plane Processing through Network Capability Exposure

This chapter introduces a new capability exposure for call control in a remote expert calling use case. The proposed capability exposure is designed as a generic solution for managing call control, specifically by routing media through a media server. It discusses potential enhancements for the remote expert calling scenario and other applications involving user plane processing, with a focus on the required features for effective capability exposure.

5.1. Use Case Description

To improve understanding of how network exposure can enhance user plane processing, consider the following illustrative use case. Figure 5.1 depicts the scenario.

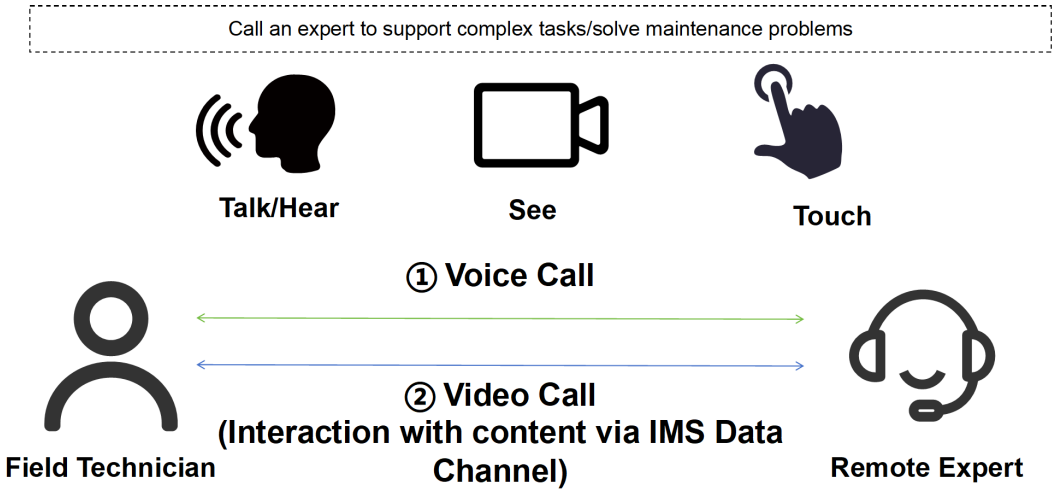


Figure 5.1: Illustration depicting the use case scenario [54] modified

A voice call is established between a 5G mobile subscriber and a service desk agent. The service desk agent does not need to be a 5G subscriber; the agent can use a wireline VoIP extension of an enterprise network. The mobile subscriber, for example, is a field engineer or maintenance technician. During the call, the two parties decide to enrich the voice call with interactive video content, supported by an IMS

data channel. Video is transferred between the two parties, e.g., footage from the field engineer's mobile device camera to the service desk agent. The agent can edit the real-time video, for instance, by drawing a line or arrow to point to a specific point in the picture as an instruction to the engineer. Alternatively, the agent can capture a frame from the video, edit the frame, and then send the frame as a picture back to the engineer through the data channel. The video editing commands need to be superimposed on the video footage generated by the field technician. This can be done in the following locations:

- In the communication device of the agent.
- In the communication device (mobile phone) of the field technician.
- In the network, for example, within a designated media server such as the UPF, or a media server positioned in the user plane.

This may require network capability exposure, as discussed in the following sections of this chapter.

5.2. Implementation of Video Editing Commands

Figure 5.2 illustrates the functional network entities involved in establishing an IMS call session. In this scenario, UE-A is the originating user equipment (field technician) initiating the call, while UE-B is the terminating user equipment (remote expert) receiving the call. Although UE-B does not strictly need to be registered as an IMS subscriber, for the purpose of demonstrating the call flow in this section, we assume that both UE-A and UE-B are IMS-registered devices.

UE-A is attached to the IMS network through A-SBG (A) and is registered in P-CSCF (A) and S-CSCF (A). UE-A is also registered in MMTel-AS (A) for VoLTE / Voice over 5G System (Vo5GS).

UE-B is attached to the IMS network through A-SBG (B) and is registered in P-CSCF (B) and S-CSCF (B). UE-B is also registered in MMTel-AS (B) for VoLTE / Vo5GS.

In this example architecture diagram, for a voice call from UE-A to UE-B, the user plane is established directly between the A-BGF that is selected by A-SBG (A) for this call, and the A-BGF that is selected by A-SBG (B) for this call.

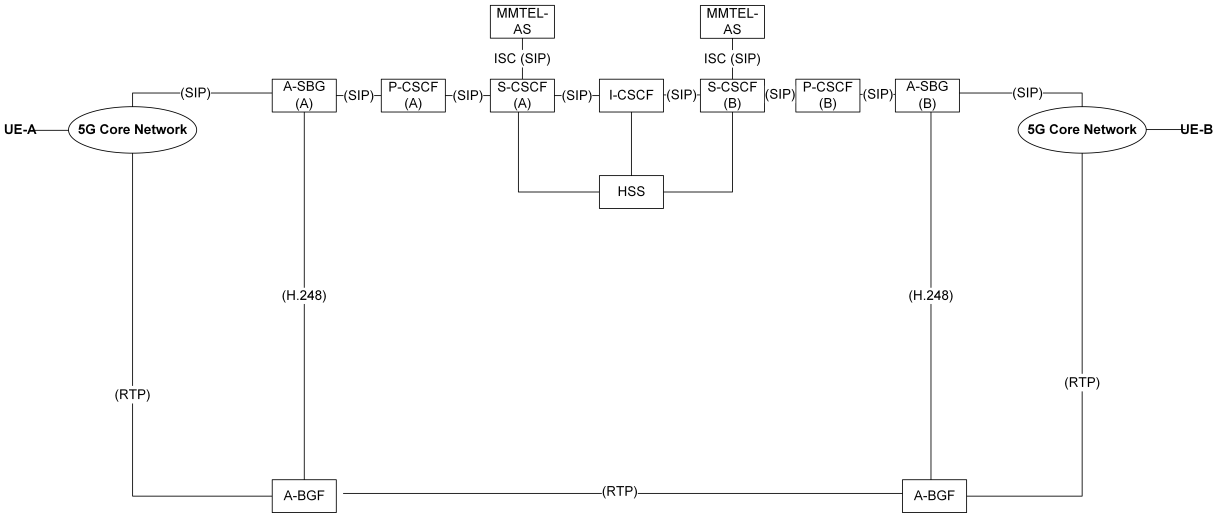


Figure 5.2: Architecture of the use case

Video editing commands in this use case refer to actions performed by the agent to modify the real-time video. For example, the agent can draw a line or arrow to highlight a specific point in the video as an

instruction for the engineer. Alternatively, the agent can capture a frame from the video, edit it, and send the edited frame as an image back to the engineer through the data channel.

In this video call scenario, one potential method for implementing video editing commands is as follows: The agent, referred to as Party B, actively monitors the received video footage and applies real-time edits. These edited commands are then transmitted to the technician, referred to as Party A. The technician's mobile device replicates its video footage and overlays the received editing commands onto the replica.

The next issue to address in this calling process is determining where specifically the video editing commands should be implemented. In this approach, four variants are identified:

1. Technician's UE: The technician's terminal receives the video editing commands. It also creates a replica of the video and integrates the commands into this replica within the technician's device.
2. UPF within the 5G Core Network: The UPF integrates the editing commands received from the expert into the video and sends the synchronized video to the technician. An applet within the UPF handles the task of embedding the editing commands into the video replica.
3. Specialist's Device: The video editing commands are implemented into the video within the specialist's terminal. Subsequently, the video, already embedded with editing commands, is transmitted to the field technician.
4. Media Server (within the IMS Network): A media server within the IMS network acts as an intermediary, facilitating call enrichment between the two parties by implementing video editing commands. The editing commands are sent from the expert to the media server.

The media server is chosen to implement the video editing commands because this thesis focuses on network capability exposure. Implementing these commands within the user terminals (as in the first and third options) does not involve network capability exposure.

Additionally, using the media server leverages the existing IMS infrastructure, which is already optimized for multimedia services like video calls. The media server is specialized in processing and managing media streams, making it more suitable for tasks like video editing compared to the UPF, which primarily handles data forwarding and QoS enforcement in the user plane.

Utilizing the media server also aligns with the principle of decoupling application-layer processing from core network functions. This separation allows for more flexible and scalable deployment of services, as the media server can be independently scaled or upgraded without impacting the core network's UPF. Moreover, this architecture enhances security and reliability, as the media server can be isolated and protected more effectively than a UPF, which handles a broader range of critical network functions.

Implementing the video editing commands within the media server directly aligns with the goals of this thesis by showcasing how network capability exposure can enhance multimedia services within the IMS framework, providing a clear use case for network operators looking to enrich their service offerings.

In summary, leveraging the media server offers integration benefits with the IMS network and existing SIP application server capabilities, though it may require enhancements to current 3GPP standards and may present challenges in managing dependencies with external application servers. Nevertheless, it provides an approach for enriching voice calls with video editing functionalities and demonstrates potential for broader usage across various voice call enrichment scenarios.

5.3. IMS Data Channel Solution

IMS data channel is part of a possible solution solution of transmitting video editing commands in this use case.

5.3.1. Introduction of IMS Data Channel

In 5G, 3GPP has expanded the capabilities of IMS by introducing new features such as a data channel application repository and a data channel server in 3GPP TS 26.114 [55]. In this context, UE-A and UE-B represent the two end terminals engaged in this communication process, as shown in Figure 5.3.

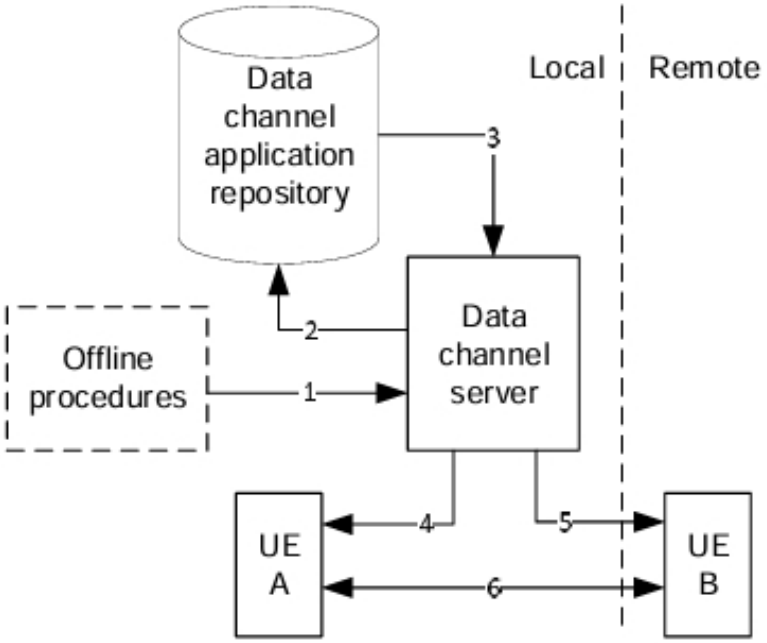


Figure 5.3: IMS Data Channel Workflow [55]

IMS data channel enables a wide array of information-sharing scenarios between UEs and the network without necessitating additional standardization or implementation efforts within the UE or IMS network. Notably, 3GPP TS 26.114 [55] specifies a mechanism, known as the bootstrap channel, which enables the distribution of an application chosen by one UE to both UEs participating in a call [54].

The data channel application involves the following steps, as depicted by the numbered arrows in Figure 5.3 [55] [54]:

1. Applications are uploaded to the network by the UE user or another authorized party.
2. Uploaded applications are stored in a data channel application repository.
3. To use a data channel during a voice call, it must first be established between the two parties involved in the call. This process is facilitated by the Data Channel Server, which manages the bootstrap data channel media. An application is then selected from the repository to be utilized within the voice call.
4. A bootstrap data channel is utilized to provide access to an application menu for both parties involved in the voice call. Once an application is selected, procedures ensure that both parties receive the same application. The selected application is sent through a bootstrap data channel to the local UE-A.

5. The selected application is sent through a bootstrap data channel to the remote UE-B. This transmission may occur in parallel with, and is rather independent of, the previous step.
6. Additional data channels, required by the data channel application itself, are established (logically) between UE-A and UE-B. Data transmission on these channels does not commence until confirmation is received that both UEs have instantiated the data channel.

5.3.2. Exposure within IMS Data Channel

The Data Channel (DC) Control API is a programmable interface provided by the MMTel-AS, enabling third-party control over multimedia telephony services. The architecture of the DC Control API is illustrated in Figure 5.4. Authorized third-party service consumers access this API through an interface defined by the network operator. Additionally, intermediary platforms, such as network capability exposure platforms (e.g., NEF), may be deployed between the MMTel-AS and these consumers to manage interactions [54]. As shown in Figure 5.4, NEF is not directly connected to the MMTel-AS.

In this use case, DC media may encompass various types of data, including raw video data, video editing command data, or pre-edited video data. The specific content transmitted via the DC media depends on the selections made in Section 5.2 regarding video editing preferences and configurations.

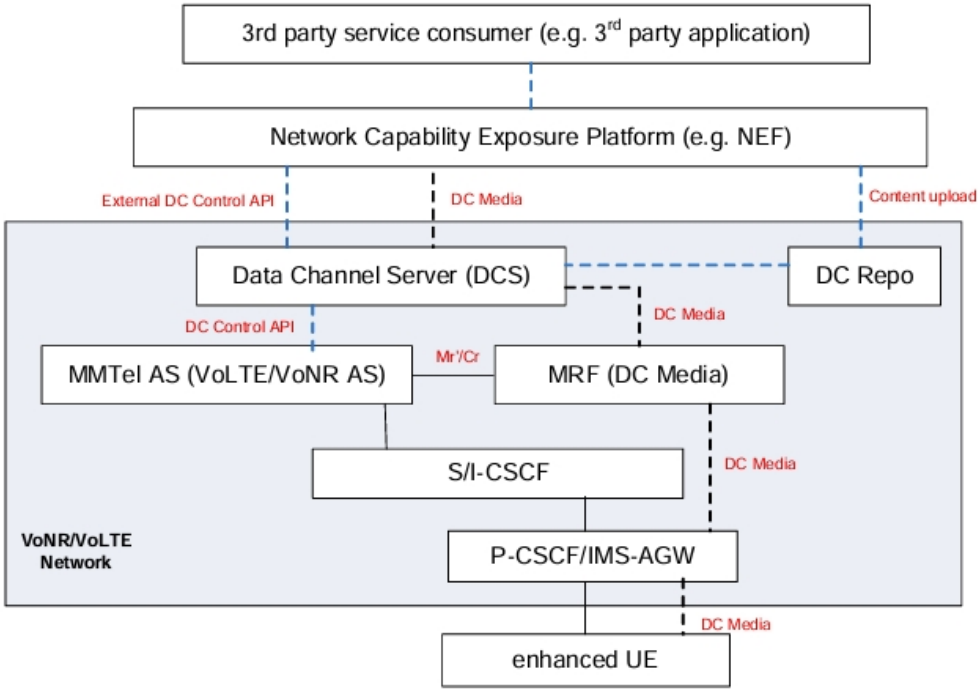


Figure 5.4: The DC control API and its relationship to other network entities [54]

Regarding the functionality of the DC Control API, it is designed to manage the life cycle of data channel services. This includes support for bootstrapping and managing application data channels, encompassing setup, takedown, and accounting processes. Specific functionalities of the DC Control API include [54]:

- **IMS Call Control:** This feature enables the management of call event subscription and notification, as well as call control mechanisms.
- **Bootstrap Data Channel Management:** The API supports the setup and release processes of bootstrap data channels.

- **Application Data Channel Management:** It oversees the setup, release, and related processes of application data channels. This includes handling communication between two UEs, or anchoring in Media Resource Function (MRF) for DC media breakout to web server logic in DCS or provided by a third party.

5.3.3. Procedure for Establishing IMS Data Channel Between VoNR End-User and Remote Party in IP Network

The processes outlined below describes the interactions between the Data Channel Server (DCS), MMTel-AS, and enterprise systems, illustrating the establishment and management of data channels for call events and applications within the overall end-to-end architecture shown in Figure 5.5.

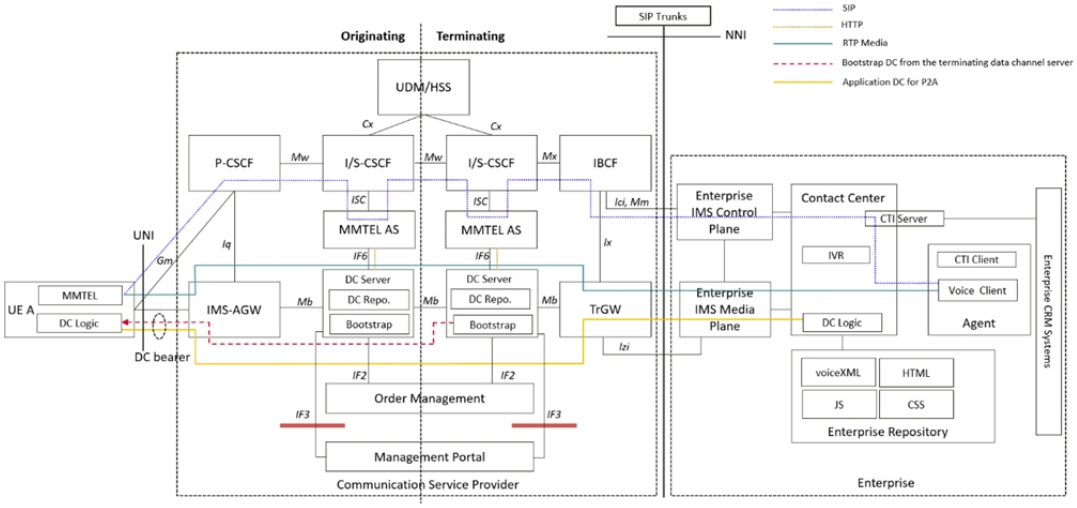


Figure 5.5: Call Control Exposure's End-to-End Architecture [54]

The IF6 interface [54] facilitates communication between the MMTel-AS and the DCS, particularly in reporting call events. When a 5G subscriber initiates a call with data channel capability, as specified in section 6.2.10 of 3GPP TS 26.114 [55], the MMTel-AS sends a notification of the call events to the DCS, which then uses the DCS subscription information to determine whether it should be notified of the specific call. The IF6 interface is important in this context because it highlights the process by which call events are communicated and managed between core network elements, providing a foundation for understanding how video editing commands and other call enrichments can be implemented and managed within the broader framework of network capability exposure.

The integrated process of establishing a voice call and enabling data channel functionality between two users involves the following steps [54]:

1. Voice Call Establishment and Data Channel Subscription:

A voice call is initially established between two subscribers. The originating subscriber's Data Channel Service (DCS) subscribes to Multimedia Telephony (MMTEL) call event notifications using the DataChannelCallEventSubscribe operation, which the DCS uses to subscribe to MMTEL AS call notification events.

2. Call Event Processing and Bootstrap Channel Establishment:

Upon receiving a call event notification via the DataChannelCallEventNotify operation, which MMTEL-AS

uses to notify the DCS about related call events, the DCS determines the permissibility of the peer-to-application (P2A) scenario based on its policy.

If allowed, the terminating DCS triggers the DataChannelCallControl operation, instructing MMTEL-AS to establish a bootstrap channel between the calling party and the DCS. The bootstrap data channel is established by UE A sending an SDP offer to UE B, and UE B sending an SDP answer back. UE A creates the initial offer, indicating it allows data channel applications from all sources specified by 3GPP TS 26.114 [55]. UE B determines which sources will be used. These notifications serve as input for DCS policy analysis. The IP addresses of DCS-A and DCS-B are determined by the network as part of the initial offer and are transparent to the UE [56].

3. Bootstrap and Application Data Channel Initialization:

The bootstrap data channel is initiated between the calling party's UE and the DCS once the SIP negotiation of the bootstrap data channel capability succeeds in the P2A scenario.

The DCS negotiates the establishment of the bootstrap data channel toward the calling party.

4. Traffic Routing and Anchor Points:

P2A is anchored to the IMS-Access Gateway (IMS-AGW) using the Gm and Mb interfaces. From there, the RTP payload is transmitted from the IMS-AGW on the UE-A side to the Transition Gateway (TrGW) on the UE-B side via the IMS bootstrap data channels. Specifically, in the AR/VR calling use case, the RTP payload consists of live video from UE-A to UE-B and the media stream containing video editing commands from UE-B to UE-A, both of which are transmitted through the IMS bootstrap data channels.

This demonstrates how the IMS data channel can be involved in transmitting traffic flow for enhanced voice calls, such as AR/VR calling.

5.4. End-to-End Solution: Call Control Capability Exposure

This use case presents a communication service enhancement for Vo5GS. As highlighted in section 5.2, conducting video editing within the IMS network instead of the 5G core network (where the UPF resides) demonstrates the need for a new capability exposure mechanism.

IMS data channels provide multimedia communication capabilities and standardized data transmission within the IMS framework, but they have limitations. These include integration complexity with the existing IMS architecture, constraints from 3GPP standards, and limited adaptability for specific use cases, such as AR video calls. Additionally, IMS data channels can lead to higher latency and reduced efficiency due to the involvement of multiple network elements, such as the Interconnection Border Control Function (IBCF), TrGW, and IMS-AGW. These elements coordinate signaling and media flows across interfaces like Gm and Mb, and SIP-based paths, resulting in complex setup procedures, increased processing time, and delays from protocol handling, session management, and policy enforcement. This layered architecture ensures standardization but makes IMS data channels less flexible and efficient for modern communication scenarios.

IBCF is responsible for managing the signaling traffic between different networks, such as between the IMS network and external networks. It handles protocols like SIP, which are used to initiate, modify, and terminate communication sessions. This processing adds a delay to the signaling process (control plane latency), as the signaling messages must pass through the IBCF, increasing the time required to establish or modify sessions.

TrGW serves as a bridge between different network domains and facilitates the transport of media (user plane) and signaling (control plane). It plays a critical role in media transmission but can introduce latency, especially if media packets need to be routed through multiple network elements or if protocol conversions are required. This results in added delays in the user plane, affecting real-time media communication, such as video calls or AR/VR applications.

IMS-AGW is the entry point for IMS-based communications. It interfaces between the IMS network and the access networks, handling signaling and media traffic. It processes both control plane (signaling) and user plane (media) traffic, which can contribute to latency. During call setup, signaling messages are processed by the IMS-AGW, and delays can occur due to session management, protocol handling, and the need for policy enforcement. In terms of media, the IMS-AGW can also cause delays due to routing, codec negotiation, and potential transcoding of media streams.

This layered architecture ensures standardization but makes IMS data channels less flexible and efficient for modern communication scenarios.

Control plane latency primarily arises from signaling and session management processes, which involve the IBCF and IMS-AGW. These elements are responsible for establishing and managing communication sessions, which involve multiple message exchanges and protocol handling, resulting in a delay before the actual media transmission can begin.

User plane latency is associated with the transmission of media (e.g., voice, video) between devices. This latency can be exacerbated by the involvement of multiple network elements like TrGW and IMS-AGW, which handle the routing of media streams and potentially perform transformations (e.g., transcoding, protocol conversion). These steps introduce delays, especially in real-time applications like AR/VR or video calls, where low latency is crucial.

To address these limitations, we propose a mechanism called 'call control capability exposure', which facilitates direct interaction between call parties while reducing the complexities associated with full IMS integration and 3GPP standardization. Although IMS components like the P-CSCF, S-CSCF, and MMTel-AS are still involved, this approach leverages the media server's capabilities exposed to the NEF, creating a more flexible architecture. Here, the MMTel-AS communicates with the NEF, which notifies an external application function, reducing reliance on extensive SIP signaling and coordination across multiple IMS elements. This design suggests reductions in latency and setup complexity, providing greater flexibility compared to fully IMS-based approaches. However, practical validation is needed to confirm these benefits.

5.4.1. Call Control Capability Exposure

Call Control Capability Exposure is proposed in this thesis to enhance the user plane capabilities in telecommunications networks. This concept enables third-party applications to interact with network functions and manage various aspects of call control, such as routing data through specific media servers, allocating resources on those servers, and executing tasks like call recording. By exposing these capabilities, service providers can offer more sophisticated and customizable communication services.

These interactions are securely and efficiently facilitated through standardized network functions like the NEF, ensuring smooth communication between applications and network elements.

In the communication process, the third-party application function acts as the central controller, meaning it orchestrates and manages the entire communication workflow. As the central controller, this application is responsible for coordinating the interaction between the network's resources and the communication

tasks, such as video editing. It interfaces with the network through the NEF, which serves as a gateway for securely accessing and utilizing network services.

To perform its role effectively, the application needs to access specific user plane control functions within the network. These functions are responsible for managing data flows and integrating video editing commands directly into the video footage being transmitted. The ability of the application to access and control these functions is made possible by a mechanism called 'call control capability exposure.' This capability exposure allows the central controller to interact with the network, ensuring that the necessary video editing commands are properly executed within the communication process.

The exposed capabilities of the MMTel-AS include the ability to route the user plane through a designated media server responsible for video editing. This capability enables the MMTel-AS to interact with external applications via the NEF and functions as an application server that supports call enrichment for the parties involved in the examined use case.

Moreover, this capability facilitates the establishment and termination of a control channel between the designated media server and the external application server. This capability exposure also comprises the capability for the third party application to indicate to the media server what actions it shall take on the media. The architecture of this network capability exposure is shown in Figure 5.6.

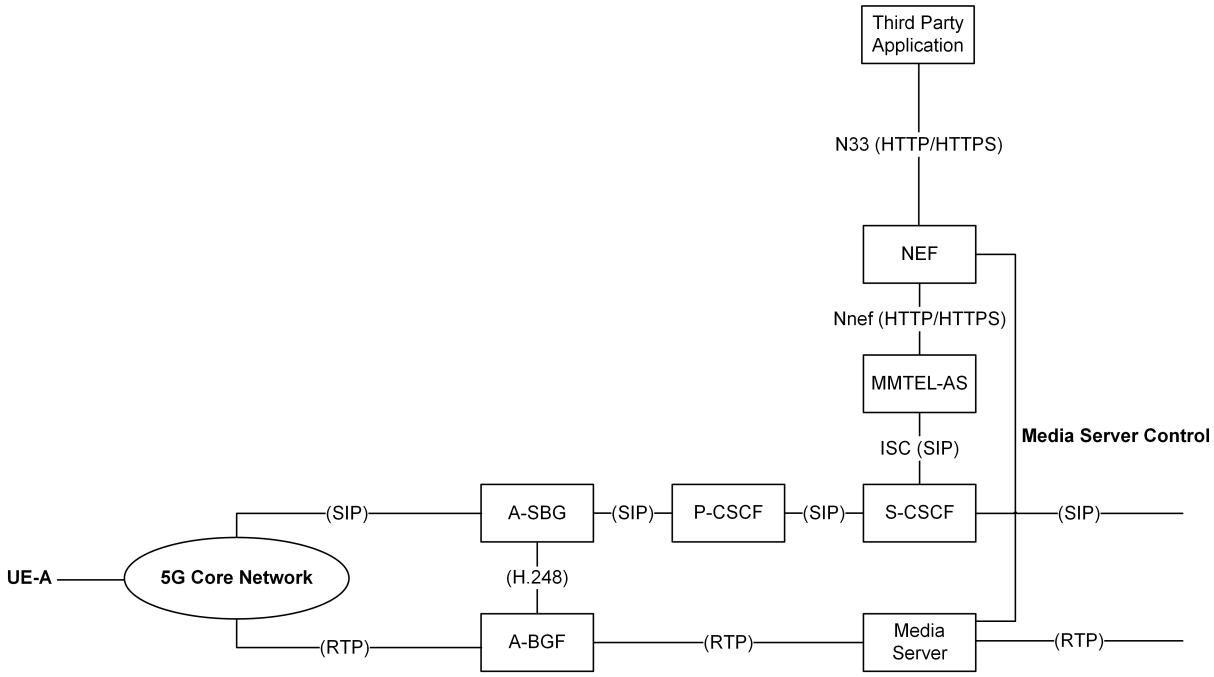


Figure 5.6: Architecture of Call Control Capability Exposure

In the example network architecture shown in Figure 5.6, a media server is deployed in the IMS network. When a call is established by UE-A, UE-A and UE-B may decide to upgrade the call to an AR call. They initiate this upgrade through WebRTC with a third-party application. The external application then performs the following actions:

- Allocates a media server processing instance via the NEF. The allocated media server instance informs the external application of the user plane (RTP) termination addresses it has reserved for media connections with both the calling party (UE-A) and the called party (UE-B);

- Instructs the MMTel-AS to route the media through the selected media server. This instruction includes the user plane termination addresses for UE-A and UE-B at the media server;

Within Figure 5.6, several protocols are shown, which are used to realize this capability exposure.

SIP is used between control plane network functions in the IMS network—such as A-SBG, P-CSCF, S-CSCF, MMTel-AS, and N-SBG—for signaling and controlling multimedia communication sessions, including the initiation and termination of calls. SIP messages are used to set up the initial call between the parties and to send re-INVITE messages for establishing additional data channels, if needed. The re-INVITE is used to negotiate updates in the SDP for media configuration. Following this, the A-SBG should instruct the 5GC to create or modify the required bearers.

The Real-Time Transport Protocol (RTP) is used for delivering audio and video over IP networks in the user plane, which consists of A-BGF, the media server, and TrGw.

H.248, also known as Megaco, is a protocol used for media gateway control. It facilitates the communication between Media Gateway Controllers (MGCs) and Media Gateways (MGs) in order to control multimedia streams across communication networks. In this architecture, H.248 is used to manage the control and signaling between various gateway components within the IMS architecture.

It is essential to explore the current extent of capability exposure (northbound APIs) defined in 3GPP TS 29.522 [9] regarding call control. Presently, it appears that this exposure does not encompass the feature to establish a control channel with a designated user plane function, necessitating its incorporation into the standard.

Protocol used in Call Control Capability Exposure

In the context of call control capability exposure, a protocol facilitates communication between the involved components, primarily the NEF, the third-party application, the MMTel-AS, and the media server. This protocol includes several key operations, each with specific endpoints and methods, to manage subscriptions, resource allocation, routing, termination, and release. Figure 5.7 illustrates the signaling sequence for this capability exposure. The purpose of this sequence is to demonstrate the operations between two different components and does not represent the actual sequence flow.

Below are the detailed and tailored descriptions of the subscription and notification events:

Step 1 Subscription

Endpoint: /ee-subscriptions

Method: POST

Summary: Subscribe to Call Control Notifications

Details: This operation allows third-party applications to create a subscription to NEF to receive notifications about specific call control events, such as call start, call end, or resource allocation status. The client must provide details such as the type of events they are interested in (e.g., `CALL_START`, `CALL_END`, `RESOURCE_ALLOCATION`), and the callback URL where they wish to receive these notifications. This ensures that the client can be informed in real-time about changes or updates in the call control process.

Request (JSON):

```
POST /ee-subscriptions
{
  "eventTypes": ["CALL_START", "CALL_END", "RESOURCE_ALLOCATION"],
```

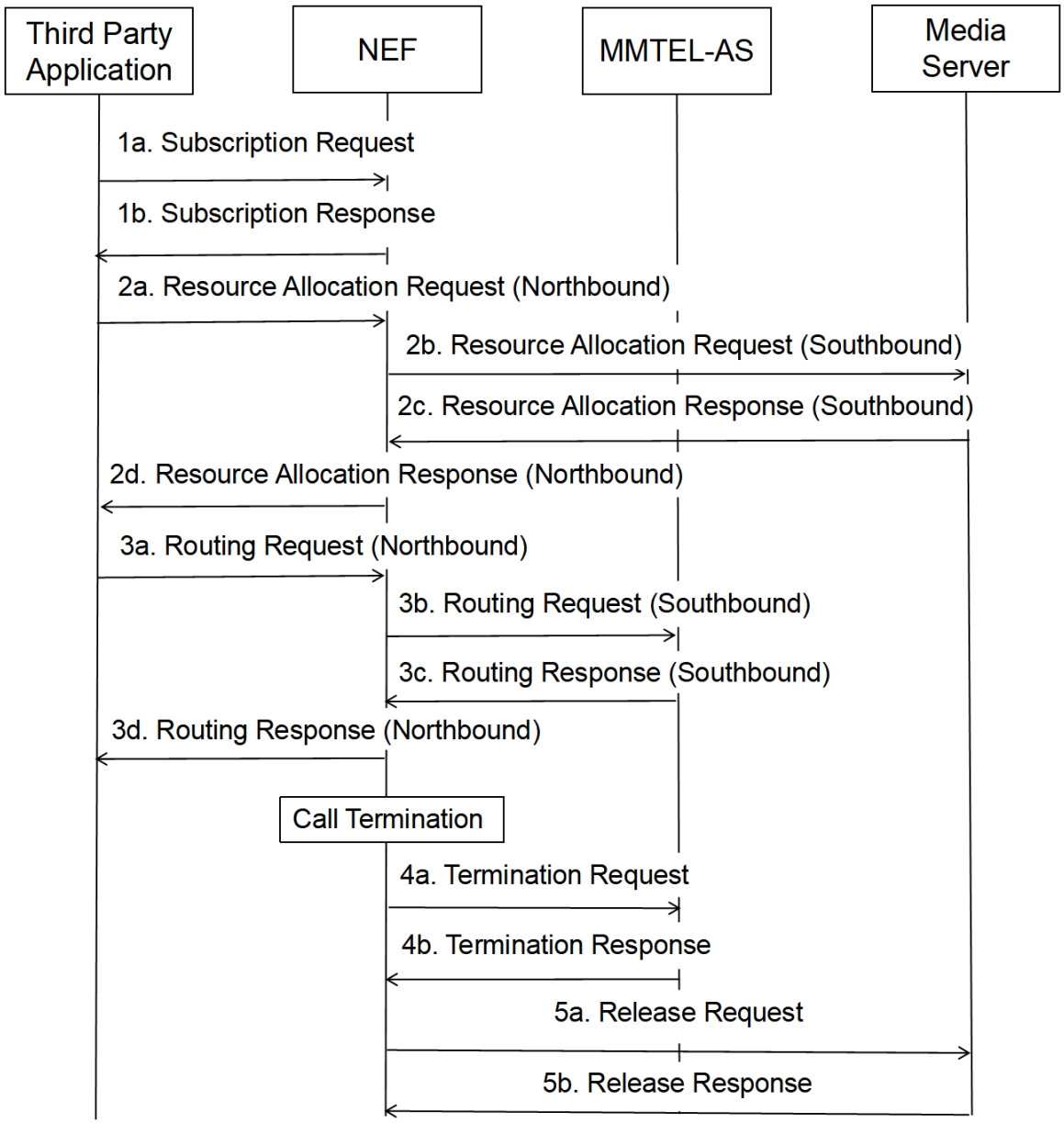


Figure 5.7: Signal Sequence Diagram of Call Control Capability Exposure

```
  "callbackUri": "https://client.example.com/notifications"
}
```

Response (JSON):

```
201 Created
{
  "subscriptionId": "12345",
  "status": "active",
  "eventTypes": ["CALL_START", "CALL_END", "RESOURCE_ALLOCATION"],
  "callbackUri": "https://client.example.com/notifications"
}
```

Attributes:

- `eventTypes`: An array of event types the client wants to subscribe to.
- `callbackUri`: The Uniform Resource Identifier (URI) where notifications will be sent. In this context, URI is used instead of URL because a URI can represent a broader range of identifiers, including URLs. While a URL specifies the exact location of a resource on the web, a URI provides a more general way to identify a resource, making it more suitable for diverse types of resources and identifiers used in telecommunications. This ensures clarity and alignment with the broader standards used in 3GPP and telecommunications.
- `subscriptionId`: A unique identifier for the subscription.
- `status`: The status of the subscription (e.g., active).

Step 2 Allocation

Allocation - Northbound

Endpoint: /resource-allocations

Method: POST

Details: This operation allows third-party applications to request resource allocation within a media server via the NEF. The third-party application specifies the type and amount of resources needed, and the NEF interprets the request and forwards it to the media server for processing.

Northbound Request (JSON) from Third-party Application to NEF:

```
POST /resource-allocations
{
  "resourceType": "media",
  "amount": 1,
  "details": {
    "mediaType": "video",
    "resolution": "HD"
  }
}
```

Northbound Response (JSON) from NEF to Third-party Application:

```
200 OK
{
  "allocationRequestId": "req-67890",
  "status": "accepted"
}
```

Attributes:

- `resourceType`: The type of resource being requested (e.g., media).
- `amount`: The quantity of resources to allocate.
- `details`: Specific details about the resources.
- `allocationRequestId`: A unique identifier for the resource allocation request.
- `status`: The status of the request (e.g., accepted).

Allocation - Southbound

Endpoint: /resource-allocations

Method: POST

Summary: Allocate resources in the media server

Details: This operation is sent from the NEF to a media server to allocate resources. The NEF interprets the northbound request from the third-party application and forwards the necessary details to the media server, including the type and amount of resources needed. The media server responds with the allocation status and resource details. The allocationRequestId from the northbound request is mapped to a specific allocationId in the southbound request.

Southbound Request (JSON) from NEF to Media Server:

```
POST /resource-allocations
{
  "allocationRequestId": "req-67890",
  "resourceType": "media",
  "amount": 1,
  "details": {
    "mediaType": "video",
    "resolution": "HD"
  }
}
```

Southbound Response (JSON) from Media Server to NEF:

```
200 OK
{
  "allocationId": "alloc-67890",
  "status": "allocated",
  "resources": {
    "ipAddress": "192.168.1.100",
    "port": 5000
  }
}
```

Attributes:

- `allocationId`: A unique identifier for the resource allocation.
- `resources`: Information about the allocated resources, including IP address and port.
- `callId`: A unique identifier for the call to be routed.
- `mediaServer`: Information about the media server, including IP address and port.
- `instructions`: Additional instructions for routing.
- `status`: The status of the operation (e.g., success).
- `message`: A message indicating the result of the operation.

Step 3 Route

Northbound Endpoint (NEF): /nef/user-plane-route

Southbound Endpoint (MMTel-AS): /mmtel-as/user-plane-route

Method: POST

Summary: Route user plane through the media server

Details: This operation is used by the third-party application to instruct the NEF to direct the MMTel-AS to route the user plane through a designated media server. The southbound request includes necessary details such as the call ID and media server information (e.g., IP address, port number). There are separate endpoints for northbound (NEF) and southbound (MMTel-AS) routing to manage different directions of traffic efficiently.

Northbound Request (JSON) from Third-party Application to NEF:

```
POST /nef/user-plane-route
{
"callId": "12345abcde",
"mediaServerDetails": {
"capacity": "High"
},
"instructions": "Route user plane through a high capacity media server"
}
```

Southbound Request (JSON) from NEF to MMTel-AS:

```
POST /mmtel-as/user-plane-route
{
"callId": "12345abcde",
"mediaServer": {
"ipAddress": "192.168.1.100",
"port": 5060
},
"resourceAllocation": {
"bandwidth": "10Mbps",
"codec": "G.711"
},
"instructions": "Route user plane through the designated media server"
}
```

Southbound Response (JSON) from MMTel-AS to NEF:

```
200 OK
{
"status": "success",
"callId": "12345abcde",
"message": "User plane routing instruction successfully processed by MMTel-AS."
}
```

Northbound Response (JSON) from NEF to Third-party Application:

```
200 OK
{
"status": "accepted",
"callId": "12345abcde",
"message": "Routing instruction received by NEF."
}
```

Step 4 Termination

Endpoint: /terminate-session

Method: POST

Summary: Terminate an ongoing session

Details: This operation is used to terminate an ongoing session. The NEF sends this request to the MMTEL-AS, providing the session ID and any other relevant information required to properly terminate the session.

Request (JSON):

```
POST /terminate-session
{
"sessionId": "abc123",
"reason": "user request"
}
```

Response (JSON):

```
200 OK
{
"status": "terminated"
}
```

Attributes:

- `sessionId`: A unique identifier for the session to be terminated.
- `reason`: The reason for termination.
- `status`: The status of the termination (e.g., terminated).

Step 5 Release

Endpoint: /resource-releases

Method: POST

Summary: Release allocated resources

Details: This operation is sent from the NEF to the media server to release the resources that were previously allocated. The request includes information about the resources to be released, and the media server confirms the release and updates the resource status.

Request (JSON):

```
POST /resource-releases
{
  "allocationId": "67890"
}
```

Response (JSON):

```
200 OK
{
  "status": "released"
}
```

Attributes:

- `allocationId`: A unique identifier for the resource allocation to be released.
- `status`: The status of the release (e.g., released).

Figure 5.8 illustrates the architecture where all these operations are executed.

5.4.2. End-to-end Workflow for the Use Case

The overall architecture for call control capability exposure is depicted in Figure 5.9. This architecture shows the interaction between network elements within the IMS environment, supporting enhanced call use cases with media handling capabilities. The reference points within the IMS architecture, as shown in Figure 5.9, are detailed in [23].

Pre-amble

Two parties have a voice call established, with the field technician connected via VoNR. The field technician uses WebRTC to communicate with a third-party application, while a remote expert, also using WebRTC, interacts with the same application. Both user devices are capable of sending instructions. This interaction relies on HTTP signaling communication between the user devices and the application server, meaning both the technician's and expert's devices act as HTTP clients to interact with the server, which hosts an HTTP server. The remote expert may also send video editing commands to the third-party application via WebRTC, which are then routed to a media server for processing based on instructions from the third-party application. The third-party application interfaces with the media server via the NEF to manage media resources.

Action

Step 1 Initiating the Voice Call:

User-A, a field technician, initiates a voice call with User-B, a remote expert. During the call, they decide to upgrade to a video call, and AR features are added. These AR features enable User-B to make real-time annotations, such as drawing circles, on the video.

Step 2 Upgrade Request and Command Routing:

To initiate the AR video upgrade, either User-A or User-B sends a command through WebRTC to a third-party application, which will handle the following operations:

- Resource Allocation: The third-party application instructs the media server in the IMS network via the

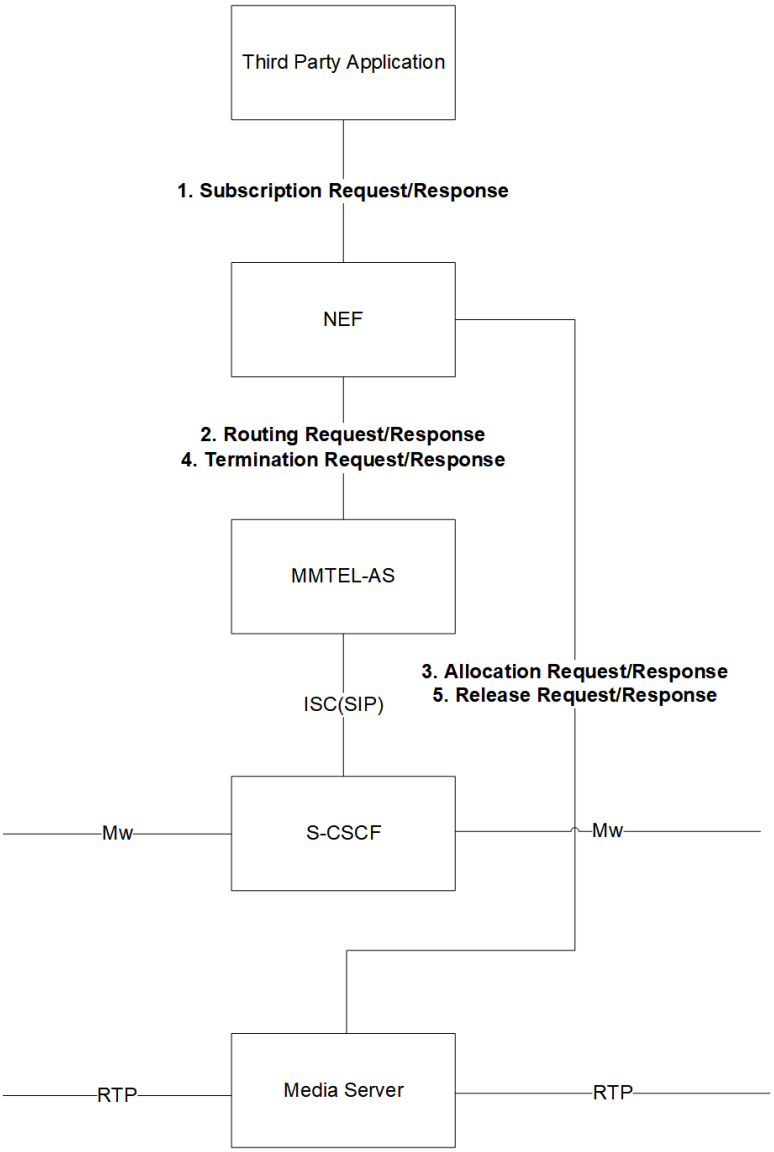


Figure 5.8: Architecture of Call Capability Exposure Operations

NEF to allocate a recording instance for implementing video editing commands. The media server then provides the necessary IP addresses and reports back to the third-party application through the NEF.

- Video Routing Configuration: The third-party application sends a command via the NEF to the MMTel-AS, directing it to route the video replica from User-A through the media server. The MMTel-AS updates the call routing to direct RTP streams through the media server.

Step 3 Command Transfer for AR Annotations:

The video editing commands, such as annotation instructions, are transmitted from User-B to the third-party application using WebRTC. The third-party application then forwards these commands to the media server via the NEF and instructs the media server to implement the AR annotations in real time.

Step 4 Video Processing by the Media Server:

The media server receives the video stream from User-A via the A-BGF and applies the AR annotations,

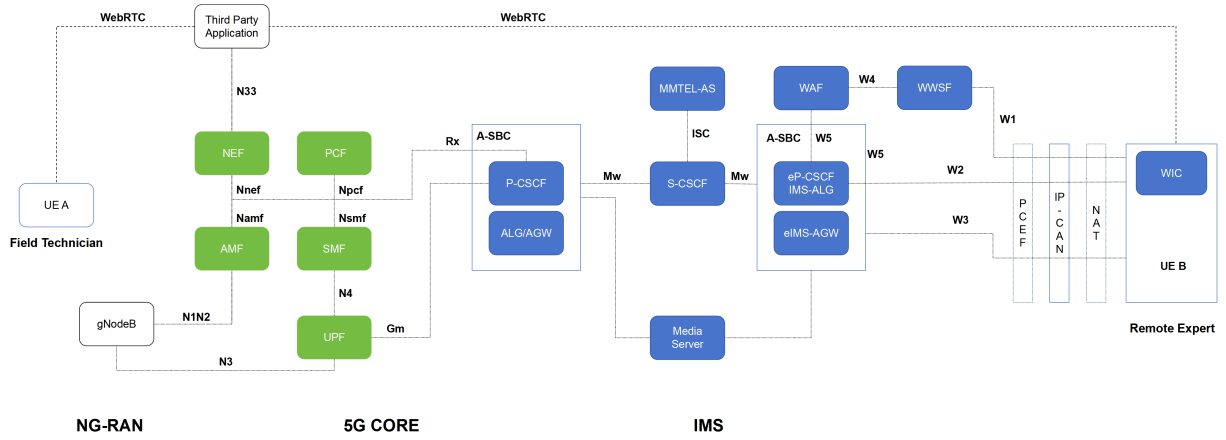


Figure 5.9: Overall Architecture of Enhanced Call Use Case

implementing the commands in real time. Once processed, the media server sends the augmented video stream back to User-A, routing it through the A-BGF and UPF.

Step 5 Call Termination and Resource Release:

At the end of the call, a user sends a termination command to the third-party application via WebRTC or simply sends a SIP BYE. User-A and User-B may also choose to disable the AR features while keeping the call ongoing as either a voice call or a video call. The application then:

- **Stopping AR Processing:** Instructs the media server via the NEF to stop AR processing and release any allocated resources.
- **Media and Command Termination:** Sends a command through the NEF to the MMTel-AS, directing it to stop routing the media and commands and to terminate the session.

5.4.3. Another Example Usage of Call Control Capability Exposure: Call Recording Scenario

One specific use case of call control capability exposure involves call recording, where a user or the network desires to record a voice call. NEF facilitates communication and instructions between an AS, which acts as the core service logic handler, the third-party application responsible for managing the recording, and the media server that performs the actual recording. Here's how the capability exposure can be utilized to achieve this:

Call Setup: When a user (UE-A) initiates a call to another user (UE-B), a SIP INVITE message is sent to the IMS core, first reaching the P-CSCF, which then forwards the message to the S-CSCF.

The S-CSCF acts as the central control point for call routing. It decides whether to forward the call to an AS for additional services, such as call recording, based on preconfigured service triggers. If recording is required, the S-CSCF forwards the call setup to the appropriate AS (not the MMTel-AS in this case).

Routing to the Application Server: Once the S-CSCF receives the call setup request, it forwards the message to the AS responsible for managing the service logic. The AS processes the SIP INVITE and, if call recording is required, triggers the necessary interactions, including notifying the third-party application via the NEF. The NEF relays key call setup metadata (such as session ID, call participants, media type, and call routing information) to the third-party application, enabling it to manage the recording process effectively.

NEF Interaction and Communication: The AS interacts with the NEF to notify the third-party application about the call establishment. The NEF acts as an intermediary and facilitates the exchange of information between the network functions (like the AS) and external applications (like the third-party recording application). NEF securely relays the call setup notification to the third-party application, providing it with the session ID needed for managing recording and retrieval operations.

End-to-End Call Recording Process:

- User initiates recording: Either UE-A or UE-B requests to record the call, which triggers an instruction sent to the third-party application via the NEF.
- Allocating resources for call recording: Upon receiving the notification, the third-party application allocates a recording instance on the media server. It also requests the IP addresses and port numbers for media termination (where the RTP media will be sent). The third-party application then sends an instruction (again via NEF) to the AS, directing it to route the user plane traffic (the RTP media stream) through the media server using the provided IP addresses and port numbers.
- Routing media via media server: The AS updates the call setup process to route the RTP streams between UE-A and UE-B through the media server, which is now responsible for capturing and recording the media streams.
- Recording process: When the call is answered, the AS notifies the third-party application via the NEF. The application then instructs the media server to start recording the RTP streams (audio and video, if applicable). At this point, both UE-A and UE-B receive a notification from the third-party application that the call is being recorded. The media server processes the streams, stores the recordings, and releases resources when the call ends.

Call Termination and Retrieval: At the end of the call, the AS notifies the third-party application via the NEF to stop recording and release the allocated resources. The third-party application then retrieves the recorded media by sending a request to the media server through the NEF, which securely manages the data transfer. After the call ends, the media server stores the recorded media (e.g., audio, video, or both). The media server delivers the media to the third-party application via HTTP/HTTPS. The request and response may include metadata (e.g., session ID, timestamp) to identify the specific recording. The recorded media is typically provided in a standard file format such as MP3, WAV, or MP4, depending on the media type.

The architecture of the process described above can be found in Figure 5.10. MMTEL-AS is not shown in the figure. A text description of how the components shown in the figure are involved in this use case is as follows:

- UE-A sends a SIP INVITE to the P-CSCF, which forwards it to the S-CSCF.
- S-CSCF routes the SIP INVITE to the AS (Application Server).
- AS processes the SIP INVITE and checks if call recording is required. If so, it sends a notification to the NEF.
- NEF relays the notification to the third-party application.
- Third-party application allocates a recording instance on the media server via the NEF.
- Media server provides IP addresses and port numbers for media termination to the third-party application through the NEF.
- Third-party application instructs the AS via the NEF to route media (RTP streams) through the media server.

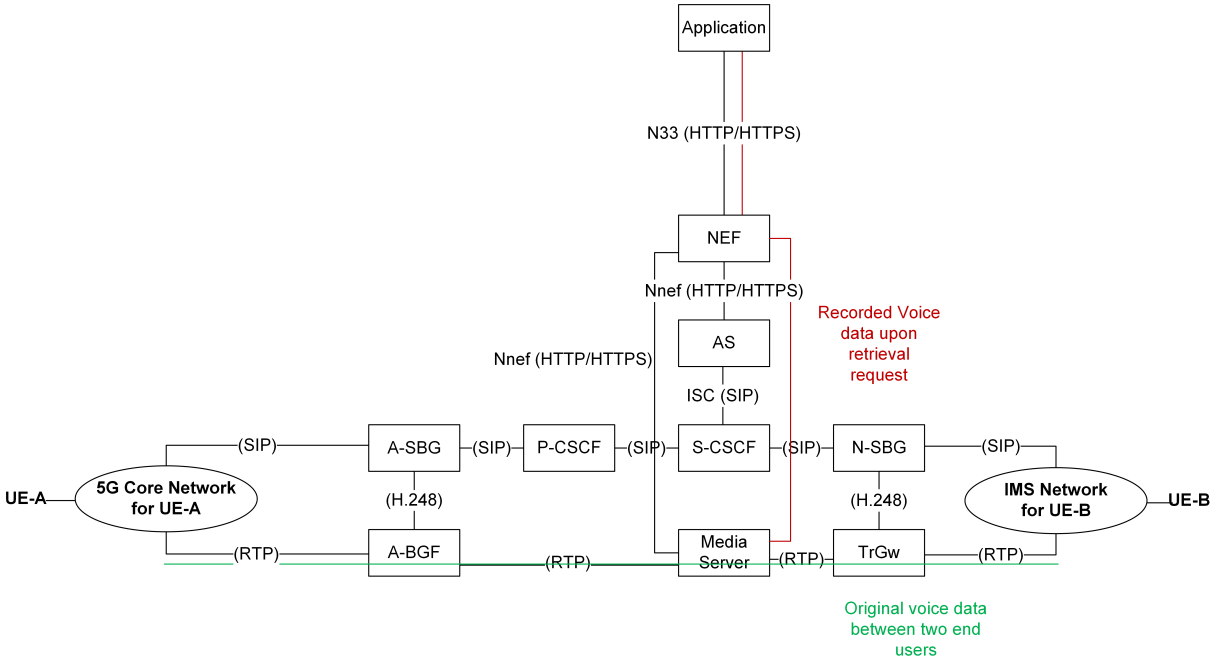


Figure 5.10: Architecture of Call Control Capability Exposure in Call Recording Scenario

- AS updates the call routing to direct RTP streams between UE-A and UE-B through the media server.
- AS notifies the third-party application when the call is answered via the NEF.
- Third-party application instructs the media server to start recording the RTP streams via the NEF.
- Media server records the RTP streams.
- UE-B ends the call.
- AS notifies the third-party application of the call termination via the NEF.
- Third-party application instructs the media server to stop recording via the NEF.
- Media server releases resources and stores the recorded media.
- Third-party application requests the recorded media from the media server via the NEF.
- Media server provides the recorded media to the third-party application via the NEF.
- Recorded media is processed or stored as needed by the third-party application.

The following protocols are used in the process of call control capability exposure for call recording:

SIP is responsible for managing signaling and call control across various network functions. This includes handling the initiation, modification (through re-INVITE messages), and termination of communication sessions.

RTP is used in the user plane to deliver audio and video streams over IP networks, ensuring real-time transmission of multimedia content between endpoints.

H.248 is the protocol used between the AS and the Media Server. It controls multimedia streams and manages resources within the Media Server, facilitating the coordination of media stream routing, session control, and resource allocation.

5.5. Conclusion

In this chapter, we proposed a new capability exposure mechanism: call control capability exposure, demonstrated through specific use cases that enable third-party applications to control user plane processing. This capability, which is currently not standardized within 3GPP specifications, allows applications to influence media routing and apply enhancements—such as voice-to-text conversion and live translation—through a general media server equipped with a voice detection system. This system detects spoken language, converts it into text, translates it into another language, and then sends the enhanced media back to the third-party application via the NEF.

This new capability is implemented within the NEF and affects control plane elements (AMF, PCF, SMF), user plane components (UPF), and IMS components such as the MMTel-AS and the media server.

We explored two use cases to demonstrate the capability's utility: in the first, the MMTel-AS within the IMS network was instructed to route commands to a designated media server to integrate AR annotations into a live video stream; in the second, call control capability exposure facilitated a call recording scenario by directing the AS to route voice data to a media server, illustrating the management of user plane resources for specialized tasks.

In summary, this chapter has outlined the theoretical network architecture, protocol interactions, and end-to-end processes for enabling call control capability exposure through the NEF. This mechanism provides third-party applications with granular control over user plane processing and media flow, demonstrating its potential for precise, dynamic interaction with network resources in future 5G and beyond use cases.

6

Network Exposure for Connected Drones

This chapter introduces a use case focused on network capabilities for connected drones. It begins by explaining what drone connectivity is, the types of available drone connectivity, and why it is needed. Following the description of the use case, it analyzes the network APIs and service APIs involved. Additionally, enhancements to some CAMARA APIs are proposed to expand the current use case for connected drones, while other capabilities beyond standard exposure are detailed in the final section.

6.1. Drone Connectivity

There are different types of connectivity, including satellite, and cellular networks like 4G/LTE and 5G, each with its own advantages and limitations. Robust connectivity is essential for tasks such as navigation, remote control, and real-time data transmission, ensuring that drones operate safely and effectively.

Drone communications over unlicensed spectrum are limited in range and are restricted to visual line of sight (VLOS). Among the various connectivity methods for drones, radio frequency is considered the least secure due to its susceptibility to signal hijacking.

Satellite connectivity is mostly used in large military UAVs, which cover extensive distances and altitudes. While satellites offer excellent coverage, they come with significant latency and high costs. Additionally, satellite communication equipment is often too bulky and heavy for drones. Although the growth of drone usage and regulatory developments is expected to increase demand for satellite connectivity by the end of the decade, there are currently limited commercial solutions using satellite communication for drones.

Cellular connectivity for drones is available via 4G/LTE and 5G networks. 4G enables faster and higher-volume data transfers compared to earlier network generations and provides a good range over long distances. However, drive tests have shown that while the existing 4G network can support command and control, as well as payload communications for beyond-visual-line-of-sight (BVLOS) operations, advanced use cases may require improvements in network latency and uplink data rates [57]. This is where 5G excels, offering significantly higher data speeds and reduced latency, which is essential for transmitting real-time high-definition video during autonomous or semi-autonomous operations.

Regardless of the type of data transmitted over cellular networks, the command and control functions of a drone, along with other tasks, heavily rely on robust connectivity. The utmost concern for any UAV pilot is ensuring that the drone does not lose connection and become lost. Connectivity is important for the following reasons:

- Navigation: Remote pilots control the drone from the ground. Whether the drone follows a predefined flight pattern or is manually controlled, geographical systems such as GPS are needed to guide it to its destination.
- Location determination: The pilot needs to monitor the drone's location, especially if it is out of sight, to ensure it stays out of no-fly zones.
- Remote control: Connectivity is crucial for remote control, enabling the drone to receive commands from the ground.
- Report of drone conditions: Drones in Europe are governed by the Unmanned Aircraft Systems Traffic Management (UTM) system. Regulations require secure, reliable, and resilient connectivity, mandating that all drones operating in European airspace report their position, direction, and speed to conventional air traffic control via the UTM [57].
- Takeover of remote control: Other parties, such as official organizations monitoring drones, need the ability to take over drone control when necessary, including movement control and the use of onboard cameras.
- Sensing: While flying, drones may use radar systems to detect obstacles by sending and receiving waves. This sensing process also requires connectivity to ensure real-time data transmission and processing, enabling the drone to react promptly to detected obstacles.
- Notification of crowds within a geographical area: For safety purposes, the drone needs to be informed if it is flying over crowds, enabling it to avoid these areas and reduce the impact in the case of an incident.

These activities require information from the telecom network. Network capability exposure may form a tool to safely provide the needed information to third-party applications and enhance drone performance.

6.2. Use Case Description

This section describes a use case originally from an Ericsson blog [58] to demonstrate the usage of capability exposure in connected drones. Figure 6.1 illustrates this use case. In this scenario, a drone is connected to a 5G network through a specialized communication module embedded within the drone. The drone is then controlled remotely by a pilot using a drone management application. This use case is not specific to Ericsson but illustrates a general scenario applicable to any compatible 5G network setup. Network exposure in this use case is introduced as three APIs, temporarily referred to as API 1, API 2, and API 3. In the use case, API 1 and API 3 are implemented between the NEF and the third-party application over the N33 interface, while API 2 is implemented between the SCEF and the third-party application over the T8 interface.

The use case begins with the authentication and authorization of the drone and the pilot. The drone management application initiates this process by triggering API 1. API 1 communicates with the NEF within the core network. The NEF then notifies other network functions, such as the NRF, to initiate authentication using OAuth 2.0 [59], a protocol designed to authorize websites or applications to access resources hosted by other web applications on behalf of a user.

Once authenticated, a secured data channel is then established through 5G, ensuring reliable and efficient data transmission between the drone and the enterprise's systems. The pilot can now control the drone and receive real-time data, and defines the drone's mission and commands it to take off. The pilot specifies the flight pattern for the inspection of a tower and sends the mission details to the drone. Commands and flight patterns are transmitted through a secured data channel.

During the flight, the pilot commands the drone to zoom in for a closer inspection. Initially, the image quality is insufficient, with a QoS of 500 kb/s. To address this, the pilot uses API 2 to request an enhancement in video quality. Based on this request, the SCEF coordinates with the PCRF and PCEF to optimize the bandwidth of the data connection (specifically, the bearer in the PDN connection) and instructs the drone to increase the bitrate to 8000 kb/s in real time. This instruction is sent via the 4G network's control plane to the application in the drone, which then adjusts the bitrate. The resulting adjustment enables high-quality video suitable for detailed inspection.

API 2 manages the quality control index mechanism in 3GPP LTE networks, ensuring bearer traffic is allocated appropriate QoS levels. It guarantees a network bearer within each data communication session, applying lower latency, jitter, and maximum data burst volume.

Remote control of the drone requires seamless communication and low latency. Upon access request from the drone management application, API 3 is triggered to switch data traffic from the Broadband IoT slice to a low-latency Ultra-Reliable Low-Latency Communications (URLLC) slice. This ensures optimal real-time performance for remote control.

The remote pilot maneuvers the drone up the tower for further inspection. At a second inspection point, the pilot uses the drone management application to activate enhanced QoS settings, selecting high-quality video streaming. This action triggers a request via API 2 to the SCEF, which facilitates communication with the Serving/PDN Gateway (S/P-GW) through the Policy and Charging Rules Function (PCRF) to configure the necessary QoS policies.

Finally, upon completion of the mission, the pilot clicks the "return to base" button to end the drone mission.

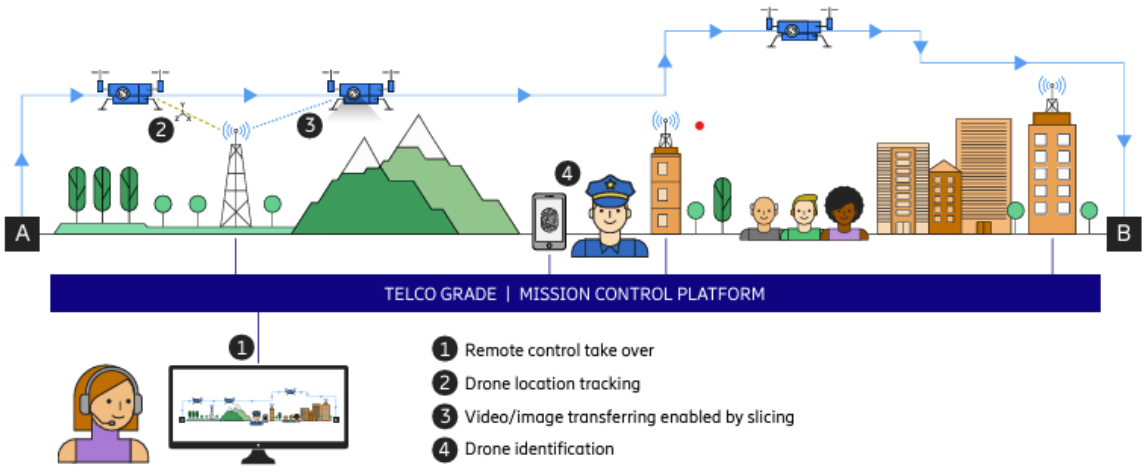


Figure 6.1: Remote Drone Control Use Case [60]

The use case demonstrates how APIs (API 1, API 2, and API 3) operate in the scenario of connected drones. API 1 handles authentication and authorization, API 2 manages dynamic QoS adjustments for video streaming, and API 3 ensures low-latency communication for effective remote control. The network diagram is shown in Figure 6.2 and Figure 6.3.

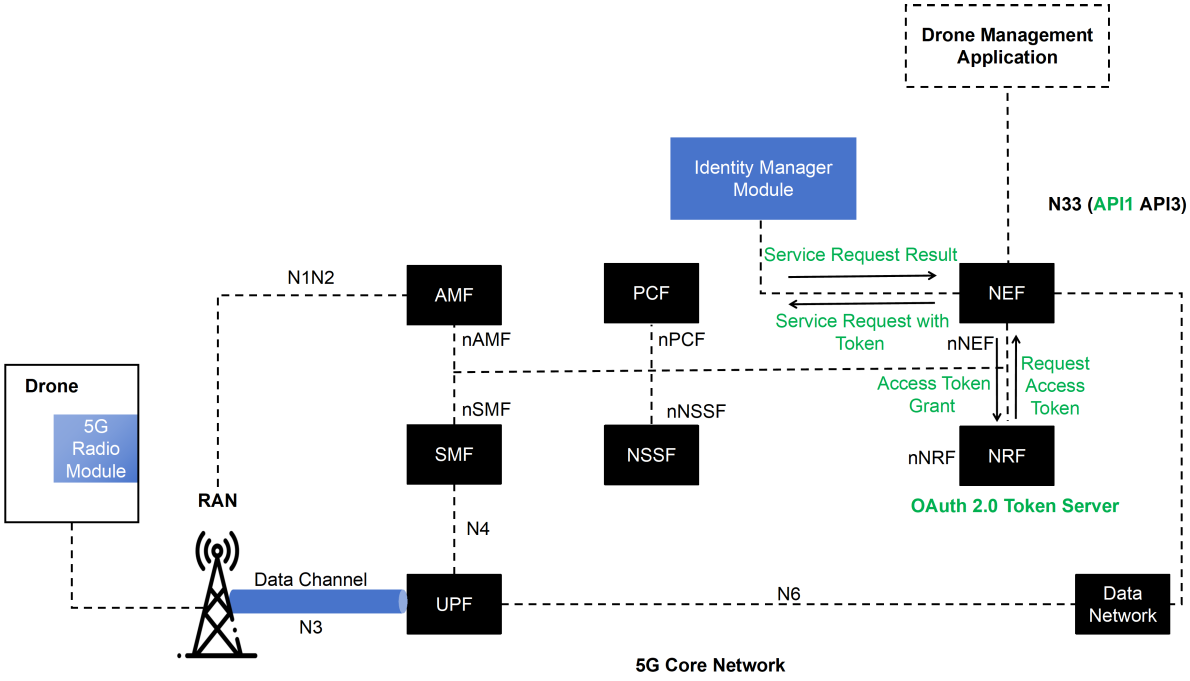


Figure 6.2: 5G network architecture in Use case

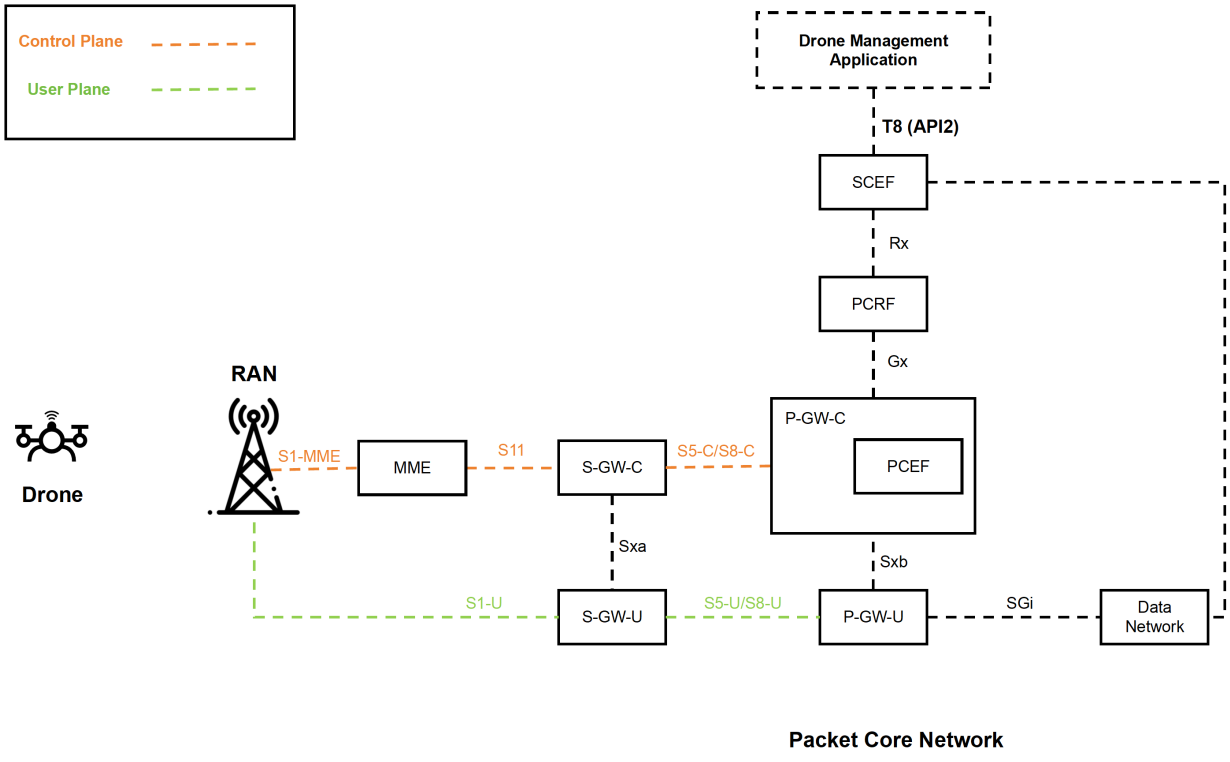


Figure 6.3: 4G Network Architecture in Use Case

6.3. Analysis of Network Capability Exposure in the Use Case

In the use case described in Section 6.2, several APIs (API 1, API 2, and API 3) demonstrate the network capability exposure of a drone application, each serving a critical role in drone connectivity. For instance, API 3 enables switching to a low-latency slice, essential for tasks like remote control and AR/VR services.

While these APIs showcase specific functionalities within the use case, they do not directly correspond to the standardized service APIs defined by the CAMARA project. CAMARA aims to ensure interoperability and consistency by standardizing network capability exposure across diverse platforms.

Ericsson's capability exposure solution integrates network exposure functions (SCEF/NEF) to process data from the network and translate application-level API instructions into network commands. This highlights that the exposed capabilities from network functions do not always directly mirror the APIs provided by network capability exposure platforms.

Although Ericsson's use case involves these APIs in both 4G and 5G contexts, this thesis will focus on analyzing network capability exposure in the 5G context.

Controlled Security

In the context of drone applications, API 1 plays a pivotal role in ensuring security control. It manages the authorization of drones using 3GPP-based mobile connectivity, enabling them to be accessed and controlled remotely via a drone management application. This process involves registering the drone for service and ensuring it utilizes a mobile network subscription enabled for drone operations, involving several key components.

The drone management application authorizes drones utilizing 3GPP-based mobile connectivity. Drones are registered for service within the drone management application and utilize a mobile network subscription enabled for drone usage by the service provider.

Service-Based Architecture (SBA) within the 5GC enhances communication within drone applications by structuring the architecture into modular microservices, each specializing in distinct functions such as API management, authentication, and data transmission. NEF acts as the intermediary for capability exposure, handling the secure exposure of network capabilities to external applications. NEF ensures that all interactions comply with security requirements by enforcing security policies and controlling access to network resources.

OAuth 2.0 [59] is employed to authenticate and authorize applications that invoke APIs, ensuring only authorized applications can access drones and their associated services. The NRF provides endpoints for OAuth 2.0 token issuance and validation as described in 3GPP TS 33.501 [61]. The NRF issues tokens to authorized applications, which are then used to access specific resources, ensuring each request is authenticated and traceable. The NEF uses these tokens to control access to network capabilities, ensuring that only authenticated and authorized entities can interact with network services and data.

Within the SBA framework, microservices manage API requests, apply security measures, and ensure compliance with service level agreements (SLAs). The drone application uses a secured data channel within the broadband IoT slice to transmit mission details and commands to the drone, with microservices handling encryption (e.g., TLS) to ensure data integrity and confidentiality. NEF exposes network capabilities using standardized APIs or APIs specifically designed for drones, such as API 1, API 2, and API 3, which form part of a specialized service package for NEF. This enables the drone management application to securely interact with drones through the NEF's modular and secure architecture.

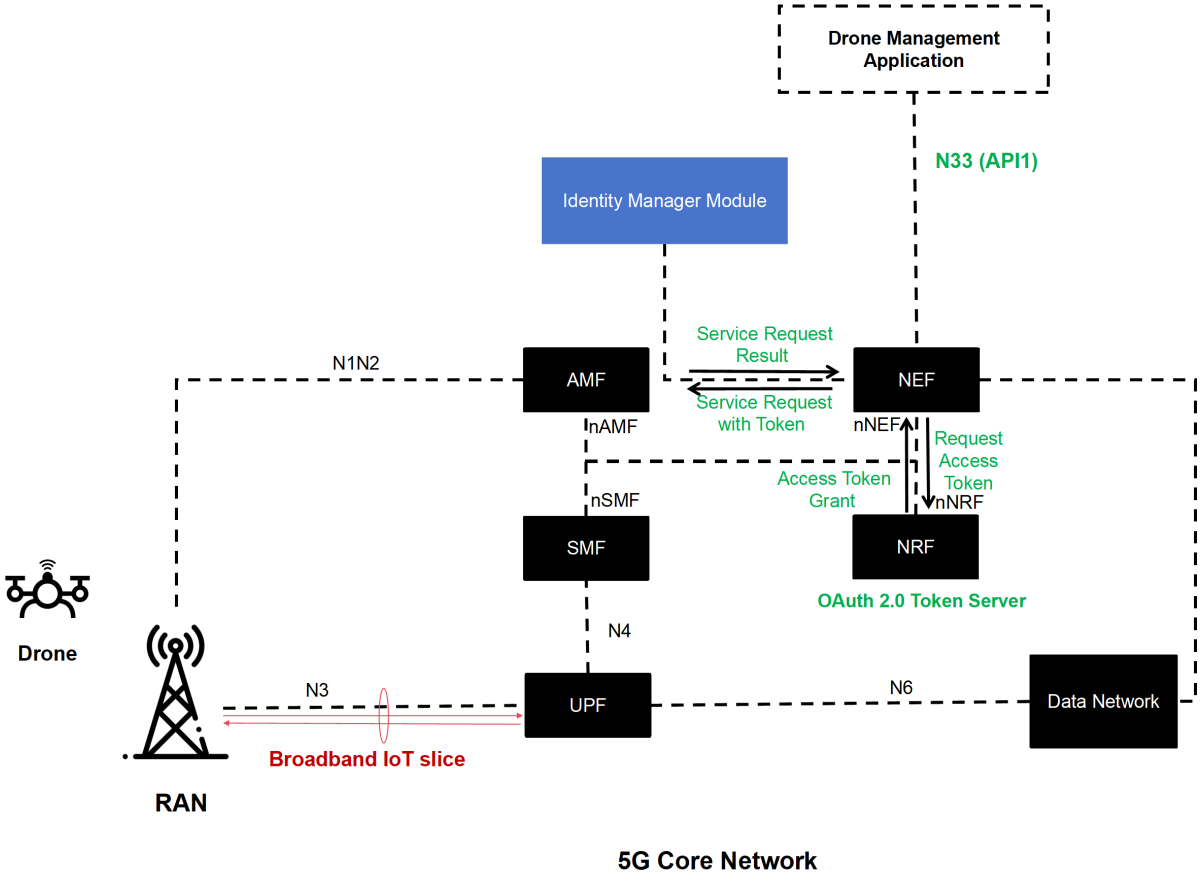


Figure 6.4: Architecture of API 1's usage

The architecture of API 1's usage is shown in Figure 6.4. It includes the following components:

- **Drone**: A connected device that, in this case, initially leverages the broadband IoT slice for communication. It utilizes the 5G network for data transmission and remote management tasks, requesting access to network services via API 1.
- **RAN**: Handles the 5G connection between the drone and the network. It manages the radio interface and forwards data packets from the drone to the core network via the UPF.
- **UPF**: Acts as a packet data gateway between the RAN and the data network, routing user data packets from the drone to external applications. The UPF terminates the PDU session and ensures that user traffic is delivered correctly.
- **SMF**: Oversees the establishment, modification, and termination of PDU sessions. It manages session details and ensures the drone's connectivity is maintained as required, while interfacing with the UPF to enforce session policies.
- **NEF**: Exposes 5G network capabilities to external applications like the Drone Management Application. Through API 1, the NEF secures and controls access to network services, ensuring that the application can authenticate and authorize drone requests using OAuth 2.0 tokens.
- **AMF**: Manages mobility and connection setup for the drone, including functions such as registration, authentication, and handovers. It ensures that the drone remains connected to the network while mobile.

- **NRF**: Serves as an OAuth 2.0 token server, validating and granting access tokens for secure API communication between the NEF and the Drone Management Application.
- **Identity Manager Module**: Stores authentication and authorization data, verifying the drone's identity before granting access to network services through API 1. This module interfaces with the NEF to confirm that only authorized devices can access network capabilities.

The process of API 1's usage can be summarized as follows: The drone establishes 5G connectivity through the RAN. RAN manages the connection and forwards data packets to UPF. UPF routes user data packets, and SMF manages session details. SMF ensures session management while NEF handles API exposure. NEF provides secure network capabilities to the drone management application. The application requests authentication and authorization via API 1 to NEF. NEF requests and obtains OAuth 2.0 tokens from NRF. The NRF validates these tokens to ensure they are authentic and valid. Once validated, the NEF uses the tokens to enforce secure access control, ensuring that only authorized entities can access the network and its services.

In summary, the NEF ensures that all interactions with the drone management application are authenticated and authorized, maintaining secure communication and access control within the 5G network.

The Number Verification API, as defined in CAMARA guidelines [62] [63], appears to be a suitable solution for API 1, which focuses on authenticating drones via mobile network connections.

The Number Verification API includes essential endpoints (/verify and /device-phone-number) that allow verification and retrieval of the authenticated user's device phone number. The /verify endpoint verifies whether a provided phone number matches the authenticated user's device phone number, directly meeting API 1's requirement for drone authentication. Meanwhile, the /device-phone-number endpoint retrieves the authenticated user's device phone number, enabling service providers to independently verify it.

API 1 specifies authentication via the mobile network, a requirement also addressed by the security framework of the Number Verification API. The use of OAuth 2.0 with the Authorization Code grant flow ensures robust authentication and authorization processes without requiring user interaction.

The Number Verification API implements security protocols, such as OAuth 2.0 framework integration, and three-legged authentication. These measures protect sensitive information and restrict access to authorized entities only, fulfilling API 1's security standards.

While API 1 specifically targets the authentication of drones, the Number Verification API verifies mobile phone numbers associated with authenticated users. API 1 necessitates seamless integration with mobile network authentication mechanisms for secure drone control, whereas the Number Verification API primarily ensures mobile number authenticity and user device associations.

To align the Number Verification API with API 1's functionalities, several customization pathways can be explored:

- Enhanced Authentication Mechanisms: Integrate additional authentication methods specific to drone identification and security protocols.
- Extended Endpoint Capabilities: Expand API endpoints to include functionalities for registering drones, managing permissions, and facilitating real-time data exchange.
- Integration with Drone Management Systems: To ensure effective integration with existing drone management systems, the Number Verification API should be adapted to support standardized

interfaces and protocols used by these systems. This integration will enable seamless data flow between the API and the drone management systems, facilitating real-time updates and streamlined workflows. By aligning with the security protocols of the management systems, such as enhanced authentication and authorization measures, the integration will strengthen security. This approach ensures that drone operations are efficiently managed and monitored while maintaining stringent security standards.

In conclusion, while the Number Verification API from CAMARA currently serves to verify mobile phone numbers, by evolving with enhanced authentication mechanisms and expanded functionalities mentioned above, the Number Verification API can effectively meet the evolving demands of drone management systems, ensuring secure and efficient operations in 3GPP-based mobile connectivity scenarios. CAMARA could explore the development of a dedicated API suite optimized for drone management, incorporating advanced security protocols, real-time data handling, and integration with 3GPP-compliant network exposure frameworks to ensure seamless, secure, and scalable operations within drone ecosystems.

Quality of Service

In the context of drone applications, API 2 ensures optimal QoS by maintaining high-quality real-time data transfer essential for tasks such as live video streaming and remote drone control. API 2 addresses challenges such as diminishing data quality over distances by leveraging 3GPP-defined mechanisms and integrating with policy and charging functions across both 4G and 5G networks.

API 2 achieves these objectives through its application-driven QoS API available within the SCEF. This API manages QoS index mechanisms in LTE networks to allocate bearer traffic to appropriate QoS levels. By leveraging the PCRF and the Policy and Charging Enforcement Function (PCEF) in 4G networks, API 2 ensures low network latency and optimal bandwidth allocation for specific UEs, such as drones, thereby allowing for a high-quality user experience [58]. For high-quality user experience, applications must also effectively utilize the allocated bandwidth.

API 2 seamlessly integrates with the SCEF, serving as a central component for exposing network capabilities and APIs to external entities, such as drone management applications. During drone missions, API 2 enables real-time adjustments to QoS parameters to meet specific operational demands. For instance, as a drone approaches an inspection target, the pilot dynamically uses API 2 to optimize the QoS settings of dedicated bearers. This involves adjusting the bandwidth of the bearer in the PDN connection to accommodate the new bitrate. A message is sent via the 4G control plane (SCEF → PCRF → PCEF → S-GW-C → MME → RAN → Drone), and the application within the drone adjusts the bitrate, ensuring high-definition video streaming capabilities.

Figure 6.5 illustrates the architecture of API 2. The components shown and their roles are listed below:

- Drone: A connected drone requiring high-speed data connectivity to transmit high-quality video.
- RAN: Provides wireless communication between the drone and the core network.
- S/P-GW (Serving/Packet Gateway): Acts as an interface between the RAN and the core network, responsible for routing and forwarding data packets.
- PCEF: Enforces QoS and charging policies in the data plane of the mobile network, ensuring these policies are applied to the drone's data flows.
- PCRF: Determines the QoS policies for the drone's data flows, specifying QoS parameters to ensure compliance with the required traffic standards.

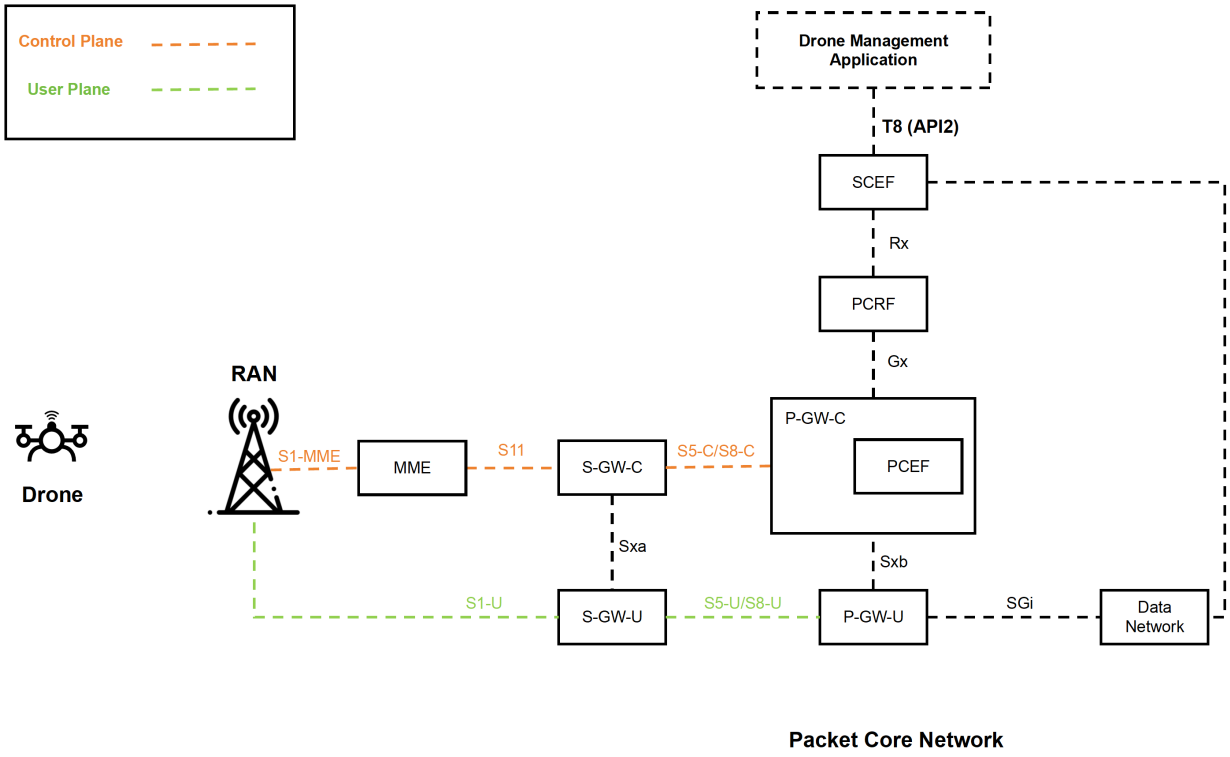


Figure 6.5: Architecture of API 2's usage

- A central hub for exposing network services and capabilities, enabling API 2 to dynamically adjust QoS parameters.
- API 2: The specific API within SCEF responsible for adjusting QoS parameters such as those required for high-quality video streaming, communicating with PCRF to ensure QoS adjustments are enforced.
- Drone Management Application: Interfaces with API2 to send user commands and requirements, ensuring the drone operates with the required QoS for optimal performance.

MME operates by receiving user commands from the drone management application, triggering events within the network. SCEF then adjusts QoS parameters dynamically to meet changing operational needs, ensuring that the drone maintains high-quality real-time data transfer for tasks like live video streaming and remote control.

Upon activation of enhanced QoS settings, the third-party application communicates with the SCEF via API 2. The SCEF sends QoS requests to the PCRF using the Rx protocol. The PCRF then pushes the updated QoS plan to the PCEF in the P-GW-C via the Gx protocol. The PCEF implements the plan by allocating resources and signaling the P-GW-U to adjust the bandwidth of the dedicated bearers managing the drone's data traffic as required.

As this thesis studies network capability exposure in 5G, the analysis of API 2 will be considered within the 5G context. Figure 6.6 illustrates the usage of API 2 in a 5G network.

The key components in this architecture diagram are as follows:

- Drone: The drone acts as the UE and connects to the RAN. The N1 interface is a direct connection between the drone and the AMF, handling control signaling between the UE and the 5G Core.
- RAN: The RAN provides the radio interface and manages communication between the drone and the core network. It is linked to the drone via the N1 interface and to the AMF via the N2 interface.

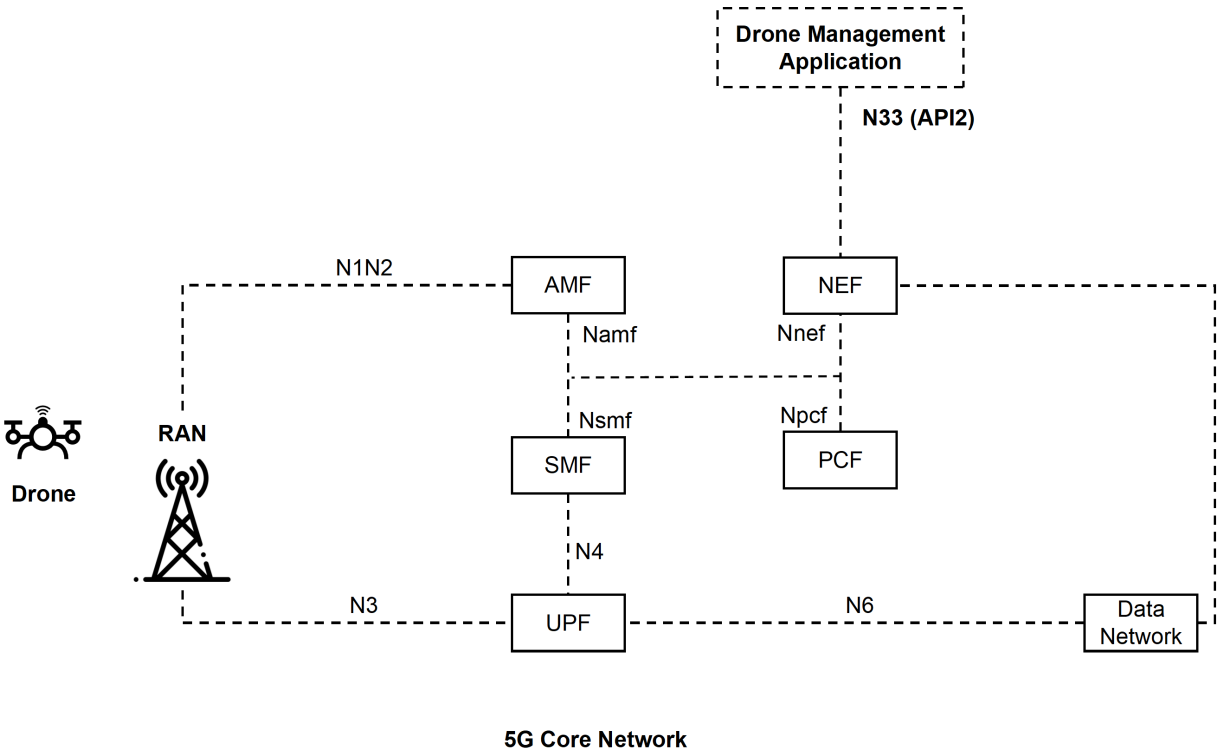


Figure 6.6: Architecture of API 2's usage in 5G

- AMF: The AMF manages UE mobility, registration, authentication, and session setup between the drone and the 5G core. The N2 interface connects the RAN (specifically the gNodeB) to the AMF for mobility management and signaling.
- SMF: The SMF is responsible for managing the drone's PDU sessions, including establishment, modification, and termination. It communicates with the UPF over the N4 interface and coordinates with the AMF via the Nsmf interface.
- UPF: The UPF handles data packet routing and forwarding and enforces QoS policies. It connects to the RAN via the N3 interface and to the Data Network via the N6 interface, facilitating data transmission between the drone and external applications.
- NEF: The NEF securely exposes network services and capabilities to external applications, such as the Drone Management Application. It communicates with the PCF through the Nnef interface to ensure the appropriate network policies are enforced.
- PCF: The PCF manages the enforcement of network policies for the drone's session, particularly QoS policies. It communicates with the NEF through the Npcf interface and with the SMF to apply dynamic QoS rules, ensuring the required low-latency and high-bandwidth connectivity for the drone.
- Drone Management Application: This external application interacts with the NEF via the N33 interface (API 2) to dynamically adjust QoS and other network parameters for the drone's mission, ensuring optimal performance based on specific requirements.

API 2 operates as follows: The drone management application sends user commands via API 2 to adjust the QoS parameters. The NEF processes these requests and forwards them to the PCF, which determines the appropriate QoS settings for the drone's traffic. The PCF then communicates the new QoS policy to the UPF through the SMF, which enforces the updated settings for the drone's data flows. Within

the control plane, commands to adjust the drone's bit rate are sent from the NEF to the AMF, then to the RAN, and finally to the application within the drone, which modifies the bit rate accordingly.

These actions ensure low-latency, high-bandwidth connections for real-time data transfer, such as video streaming, during drone operations.

Additionally, the Quality on Demand (QoD) API in CAMARA manages events to monitor and adjust QoS parameters across sessions or connections, enabling mobile applications to request improved network connection quality—such as increased bandwidth and reduced latency. The QoD API allows application developers to request stable latency (i.e., reduced jitter) or increased throughput for specific application data flows between application clients (within a user device) and backend Application Servers.

The sequence of events supported by the QoD API is shown in Figure 6.7 and described below:

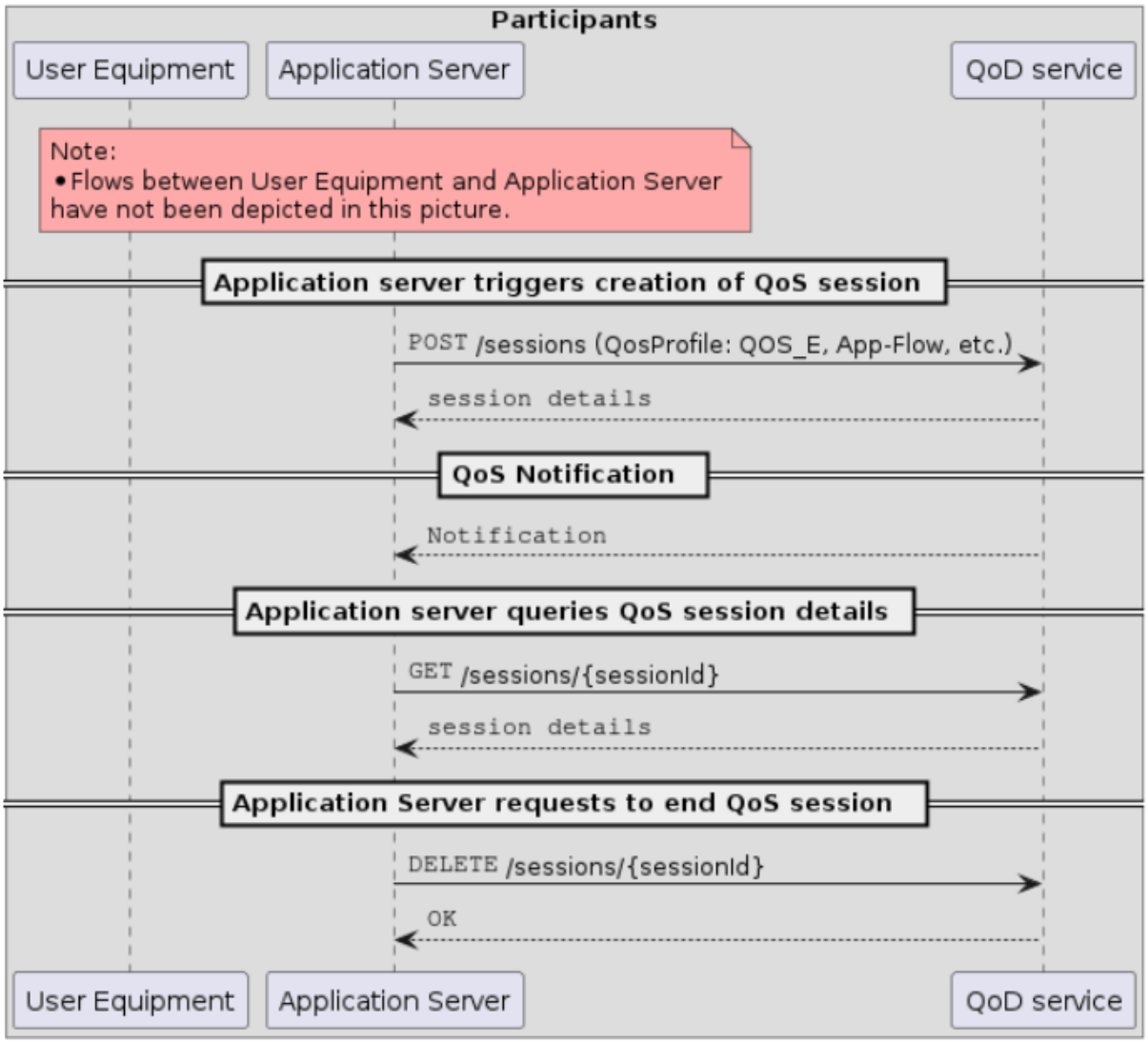


Figure 6.7: Events and processes supported by the QoD API [64]

1. Retrieve QoS Profiles: Obtain details about available QoS profiles managed by the API.
2. Request QoS Sessions: Initiate a quality-on-demand session from a mobile device to an application server by providing necessary parameters such as the desired QoS profile, duration, ports, and a

notification URL.

3. Session Management:
    - Retrieve details of an existing QoS session: Query the session details using the session ID.
    - Extend the duration of an existing QoS session: Request an extension of the current session using its session ID.
    - Terminate an existing QoS session: Request the end of the session using its session ID.
4. Notification Mechanism: Provide a URL to receive notifications about the session status, such as when a session is terminated.

The process of how QoD API works can be summarised as follows:

The application server first triggers the creation of a QoS session by sending a POST request with details such as the QoS profile and App-Flow. The QoD service then sends a notification to the provided URL regarding the session status. The application server queries the QoS session details using a GET request with the session ID. The application server requests to end the QoS session using a DELETE request with the session ID.

The example usage of the QoD API aligns with the fundamental requirements of API 2 in drone applications. Both emphasize real-time or dynamic adjustment of QoS parameters based on operational demands, particularly enhancing video quality during drone operations. API 2, like the QoD API, responds to specific events to initiate QoS adjustments, ensuring optimal performance during critical tasks.

Therefore, API 2 within drone applications can adopt a technical approach akin to the QoD API's example usage. Both manage QoS status changes effectively and facilitate dynamic adjustments to meet operational requirements, particularly in optimizing video streaming quality during drone inspections.

In summary, API 2 effectively mirrors the QoD API of CAMARA by focusing on dynamic, application-driven QoS control. This ensures consistent latency, minimized jitter, and prioritized throughput for critical data flows, essential for maintaining high-quality drone operations. The QoD API's capabilities in managing QoS profiles and session resources align seamlessly with the needs of drone applications, supporting real-time adjustments to uphold optimal network performance.

Remote Control of Drones

According to the use case description, API 3 is essential for enabling remote control of drones by orchestrating the seamless transition of data traffic from the Broadband IoT slice to the URLLC slice within the core network. This switch is triggered when flight control is transferred to remote pilots or when AR/VR services are required.

In the drone use case, the drone initially connects to the 5G network via the Broadband IoT slice. The remote pilot uses the Drone Management Application to control the drone. When low-latency communication is required, the pilot initiates a request through the application. This request, sent via API 3, is processed by the NEF, which forwards it to the AMF. The AMF interacts with the Network Slice Selection Function (NSSF) to determine the most suitable slice— in this case, the Ultra-Reliable Low Latency Communication (URLLC) slice. The SMF then configures the UPF to switch the drone's data traffic from the Broadband IoT slice to the URLLC slice. The UPF reroutes the data traffic to ensure low-latency communication, allowing the drone to operate under the URLLC slice and maintain real-time control.

While API 2 focuses on managing the quality of service for data transfer, particularly ensuring high-quality video streams by adjusting QoS parameters (such as bandwidth, latency, jitter, and data burst

volume), it does not handle the dynamic switching between different network slices. API 2 ensures that the specific data stream, like video, maintains its quality within the existing network slice.

API 3, on the other hand, is designed specifically to handle scenarios requiring ultra-low latency and high reliability, such as the remote control of drones and AR/VR services. Unlike API 2, which optimizes QoS within the current network slice, API 3 enables dynamic selection and switching to the appropriate network slice. This capability is crucial when the operational requirements change, for example, when a drone needs to be remotely piloted or when high real-time interaction is necessary for AR/VR applications.

By leveraging API 3, seamless, reliable, and high-performance drone operations are ensured, even when high-quality video streaming, managed by API 2, is required. The combination of both APIs allows for comprehensive and flexible management of network resources, ensuring both high-quality video and responsive control for connected drones.

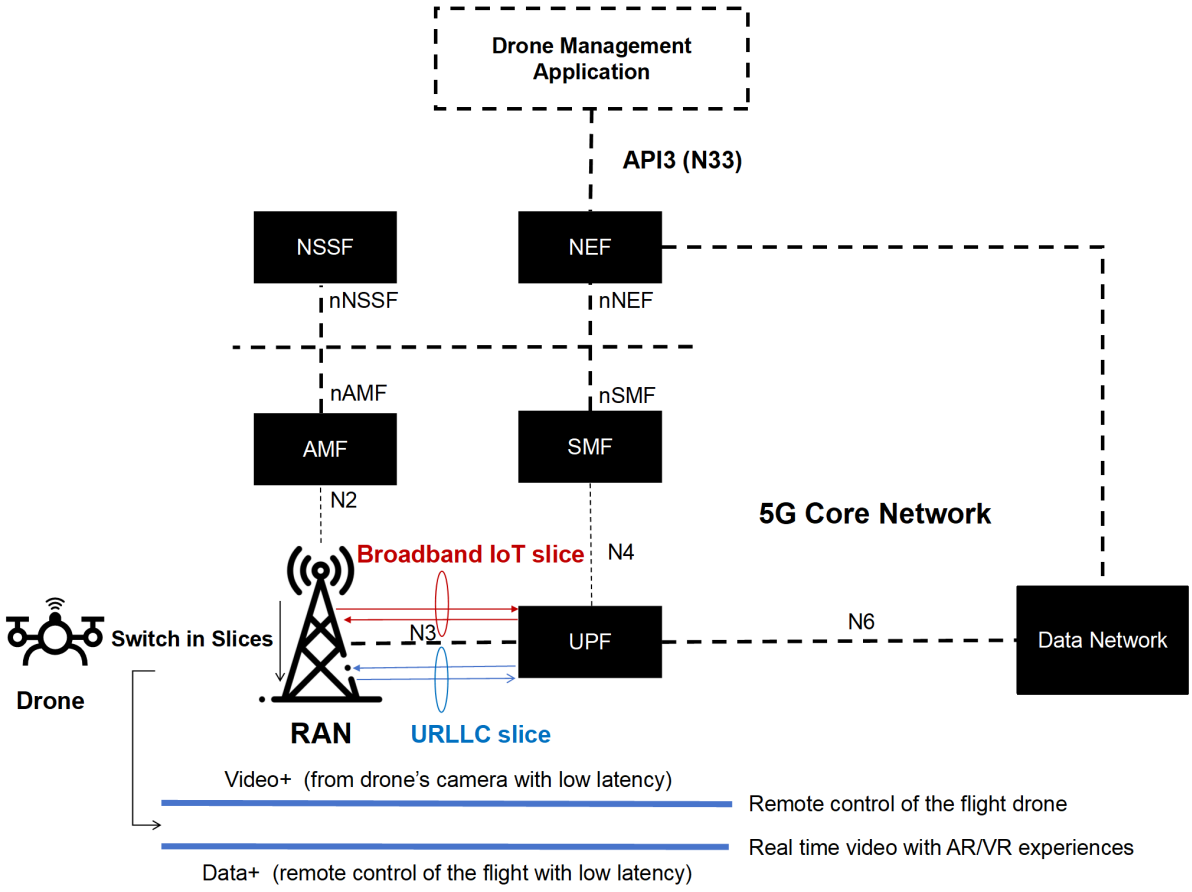


Figure 6.8: Architecture of API 3's usage

Figure 6.8 illustrates the architecture of API 3's usage. The roles of the components shown are detailed below:

- Drone Management Application: Used by the pilot to send commands and request low latency communication. It sends a request via API 3 to initiate the change of slices.
- NEF: Receives and processes the API request, interacting with the NSSF to determine the best slice for low latency.
- AMF: Supports network slice selection by interacting with the NSSF to determine the most appropriate slice based on the requirements.

- NSSF: Determines the appropriate network slice (URLLC slice) based on service requirements.
- SMF: Manages the session and configures the UPF to switch the traffic.
- UPF: Routes user data to the appropriate network slice, initially through the Broadband IoT slice and switching to the URLLC slice upon NEF instruction.
- Network Slices: The Broadband IoT Slice supports initial communication with higher latency, while the URLLC Slice provides ultra-reliable low latency communication for real-time control.
- Drone: Communicates with the 5G network through the appropriate slice managed by the UPF.

In essence, network slicing in drone operations optimizes communication efficiency by dynamically allocating resources tailored to specific operational demands. By ensuring low latency, high reliability, and optimized throughput, network slicing via API 3 supports uninterrupted remote drone control and enhances the quality of real-time AR/VR experiences essential for modern drone applications.

6.4. Enhancing Use Case Functionality with CAMARA APIs

The CAMARA project includes a variety of additional APIs beyond those highlighted in the previous section. Although not featured in the use case demo and description, these APIs are inferred to support background processes or have potential to enhance drone operations further. These additional APIs provide advanced capabilities for network optimization, resource management, and user experience enhancement. By integrating these CAMARA APIs, developers can achieve more robust and efficient drone operations, ensuring seamless connectivity and superior real-time control.

This section will explore these additional APIs and illustrate how they can be leveraged to further enhance the drone use case.

6.4.1. Location Retrieval API

The Location Retrieval API [65] allows third-party applications to retrieve the location of a specific user device, determining the area where a specific user device is situated. This location can be described in two ways: as a circle defined by coordinates (latitude and longitude) and a radius, or as a simple polygon outlined by segments connecting a series of coordinates to form a closed shape. The shape provided in the response depends on the network conditions at the subscriber's location, meaning that any supported shape could be returned.

Applications can optionally specify a maximum age for the location information, indicating the oldest acceptable age of the location data in seconds. This parameter serves to ensure that the retrieved location is sufficiently current for the client's needs. It allows clients to request location data that meets their freshness criteria, balancing real-time accuracy with network capabilities.

The maxAge parameter does not imply that location data is retrieved solely from storage. Instead, it guides the API to prioritize current location data if available. If the network can determine the device's location in real-time, the API will provide the most recent coordinates. However, depending on network conditions or caching mechanisms, the actual age of the location data provided may occasionally exceed the specified maxAge.

Alongside the location information, the response includes the timestamp when the location was determined. This timestamp informs clients of the data's freshness and supports applications where timely location updates are critical, such as real-time navigation or fleet management.

In summary, this service API will request and receive device information, such as a phone number, network access identifier, or IP address, as well as the maximum age of the location information.

An example of a Network Access Identifier is: 123456789@domain.com. This is a public identifier used to address a subscription in a mobile network. In 3GPP terminology, it corresponds to the GPSI formatted with the External Identifier (Local Identifier@Domain Identifier). Unlike a telephone number, the network access identifier is not subject to portability rules and is individually managed by each operator [66].

In return, the API will provide the area where the device is located and the last location time. The accuracy of the retrieved area is contingent on the network conditions at the subscriber's location, offering location services based on the mobile network as a complement to GPS [67].

This API is frequently used to enhance GPS tracking for critical location-based applications, particularly due to the susceptibility of GPS signals to spoofing on devices. GPS location information is obtained through the reception of signals from satellites in the Global Navigation Satellite System (GNSS). Each satellite broadcasts signals that include precise timing and orbital information. By receiving signals from multiple satellites simultaneously, GPS receivers calculate their own position through a process called trilateration, which measures the distances to the satellites based on signal travel time.

Drones, benefiting from line-of-sight connectivity with ground-based stations, require accurate location data for safe operations, task completion, and precise flight control. The integration of GNSS and emerging 5G technologies holds promise for achieving decimeter-level positioning accuracies. This advancement is crucial in navigating drones through complex environments with high precision [68].

Figure 6.9 illustrates the architecture through which the CAMARA API - Location Retrieval API facilitates the retrieval of a drone's location. Initially, a pilot initiates a location retrieval request through a third-party application, which acts as a CAMARA API invoker. This request is received by the CAMARA solution suite via the Location Retrieval API and subsequently forwarded to the NEF within the 5G core network using a 3GPP Network API. Within the CAMARA suite, a transformation function maps the service API (Location Retrieval API) to the corresponding network API (MoLcsNotify API).

NEF then communicates with the GMLC to initiate the location retrieval process. This journey passes through the AMF and the Location Management Function (LMF) before reaching the NG-RAN, where real-time measurements of the drone's location are performed.

Following the measurement process, the drone's location information is transmitted back through the sequence in reverse, ultimately reaching the pilot via the CAMARA API. This comprehensive process ensures the timely and accurate retrieval of the drone's location, enabling effective monitoring and control. This method of retrieving the drone's location within the communication network will be used as part of the solution for tracking the drone's location, as discussed in Section 6.5.

Detailed process for drone location retrieval using the CAMARA Location Retrieval API can be described as follows, referencing [66], [68], and [69].

1. Pilot Initiates Location Retrieval via Third-Party Application

1. The pilot opens the third-party application (drone management application) and requests the location of the drone.
2. The third-party application constructs a request with necessary parameters, including device identifiers and the desired `maxAge` for the location information.

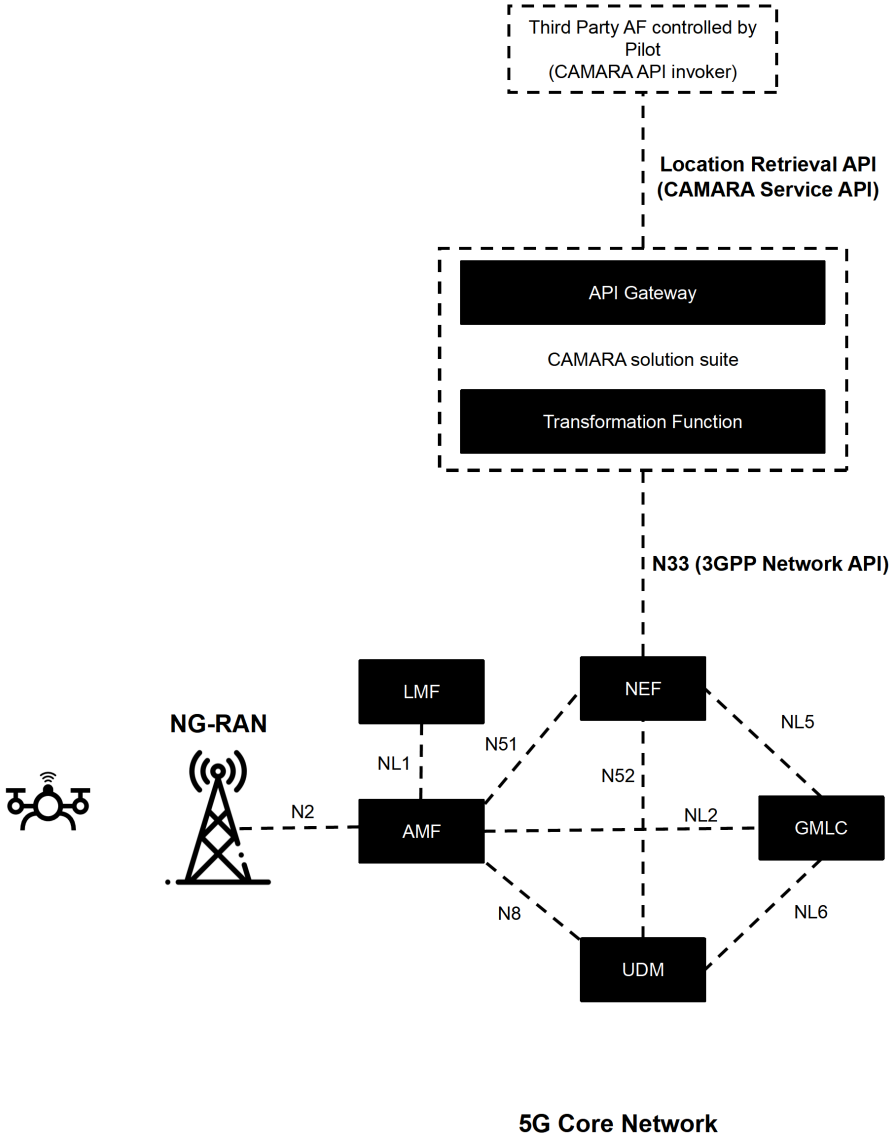


Figure 6.9: Location Retrieval API applied in a Drone Use Case

2. Third-Party Application Sends Request to CAMARA solution suite

1. The third-party applicationsends an `HTTP POST` request to the CAMARA Location Retrieval API's `/retrieve` endpoint with the following payload:

3. CAMARA Solution Suite Processes the Request

1. The CAMARA solution suite receives the request and verifies the authentication and authorization of the request.
2. Upon successful validation, the CAMARA solution suite utilizes the Transformation Function to determine the network in which the subscriber is located. The Transformation Function maintains information on correspondences between service APIs and network APIs.
3. Once the subscriber's network is identified, the CAMARA solution suite maps the service API to the corresponding 3GPP network API(s) and forwards the location retrieval request to the NEF within the appropriate 5G core network.

4. **Core Network Processing (NEF-GMLC-AMF-LMF-AMF)**

    1. **NEF (Network Exposure Function)**:
        - NEF receives the request from the CAMARA solution suite and translates it into appropriate network commands.
        - NEF communicates with the GMLC to initiate location retrieval.
    2. **GMLC (Gateway Mobile Location Center)**:
        - GMLC is responsible for managing location services within the network.
        - GMLC forwards the location request to the AMF.
    3. **AMF (Access and Mobility Management Function)**:
        - AMF manages the mobility and connection states of the UE (drone).
        - AMF forwards the request to the LMF to compute the location.
    4. **LMF (Location Management Function)**:
        - LMF is responsible for computing the precise location of the device (drone) using various measurements and reference signals.
    5. **AMF**:
        - AMF interacts with the NG-RAN to obtain necessary measurements.

5. **RAN (Radio Access Network) and Drone Interaction**

    1. **NG-RAN**:
        - NG-RAN configures the drone (UE) using the Radio Resource Control (RRC) protocol to enable positioning updates.
        - NG-RAN transmits positioning reference signals (PRS) and collects Positioning Measurements.
    2. **UE (Drone)**:
        - The drone receives the PRS and may transmit sounding reference signals (SRS) in the uplink.
        - Positioning measurements are collected and sent back to NG-RAN, which forwards them to AMF. These measurements may be done by a single antenna with beamforming, multiple antennas, or one antenna without beamforming (with a drop in accuracy).

6. **Location Computation and Response**

    1. **AMF and LMF**:
        - AMF forwards the measurement to LMF, LMF computes the drone's location using collected measurements.
        - The location is determined in the form of a circle or polygon based on accuracy and network conditions.
    2. **AMF and GMLC**:
        - The computed location is sent back from LMF to AMF, then to GMLC.
    3. **NEF**:
        - GMLC sends the location data back to NEF.
        - NEF formats the data and sends it back to the CAMARA solution suite.

7. CAMARA API Sends Response to Third-Party App

1. The CAMARA solution suite receives the location data from NEF.
2. The CAMARA API sends a response back to the third-party app in the requested format (circle or polygon).

8. Third-Party App Displays Drone Location

1. The third-party app receives the location data and presents it to the pilot.
2. The pilot can now see the drone's location on the app interface, either as a circle with coordinates and radius or as a polygon defined by multiple coordinates.

While the CAMARA Location Retrieval API effectively defines areas using latitude and longitude coordinates, along with a radius for circles or an array of points for polygons, it provides only a 2D representation of spatial data. This 2D approach captures geographic location but does not encompass the full address of a drone. In drone operations, altitude is a crucial component that significantly impacts the drone's position relative to other objects and ensures safe flight paths. Without integrating altitude, the API's 2D model falls short in addressing the spatial needs of drone navigation and collision avoidance.

6.4.2. Population Density Data API

The Population Density Data API, developed by CAMARA [70], is a specialized tool designed for drone operations, especially where real-time population density assessment is crucial for safe and efficient flights. This API allows drone operators to access population density insights for specific areas at future dates and times. These insights are derived from anonymized data collected from network-connected devices within the designated flight zone, enabling pilots to avoid congested areas and reduce the risk of drone incidents.

The API works by gathering anonymized data from network-connected devices, such as smartphones and IoT devices, within the flight zone. This data is then processed using historical information and predictive models to estimate population density. The estimates are calculated for equally sized grid cells within the specified area.

Drone pilots can retrieve processed data from the API, which provides population density estimates for future dates and times, expressed in people per square kilometer. The API offers density estimates for each grid cell and time slot within the specified range, including a minimum and maximum value to represent the estimation range. These estimates are based on historical data and predictive algorithms.

Key applications of the API include supporting Beyond Visual Line of Sight (BVLOS) operations by providing real-time data that helps operators assess intrinsic Ground Risk Class (iGRC) and make informed flight path decisions, ensuring safety and compliance with regulatory standards such as SORA 2.5 [71].

In addition, the API's population density estimates are valuable for urban planning and managing large-scale events. Urban planners can use the data to predict future population distribution for better infrastructure and resource planning. Event organizers can leverage these forecasts to effectively manage crowds and ensure public safety during major gatherings, including those involving drones [71].

Example Usage

To better illustrate how this API can be utilized in a drone use case, consider the following scenario: A drone is tasked with monitoring and collecting environmental data across urban and suburban areas. To optimize flight paths and ensure efficient data collection, the remote pilot wants to leverage population

density data to avoid densely populated areas where interference or privacy concerns may arise. This example will focus on route planning based on population density data retrieved from the Population Density API.

The Population Density API integrates into the drone management system using HTTP/REST protocols. The message flow involves the following steps:

1. API Integration Setup:

- Integrate the Population Density API into the drone management system.
- Set up authentication and API key management as required by the API provider.

2. Fetching Population Density Data: The drone management system sends a POST request to the Population Density API with the required geographic area, time frame, and precision. This request is sent over HTTP with JSON-formatted data, including geohash coordinates and a time window.

- Use the Population Density API to fetch population density data for the geographical areas where drones will operate.
- Specify the geographical boundaries (polygon) or specific points (geohashes) for which the pilot needs population density information.
- Include the required time period (start and end dates) for which the data is needed.

Example API Request (modified from the Population Density YAML file [72]):

```
POST /population-density HTTP/1.1
Host: population-density-api.com
Authorization: Bearer your_access_token
Content-Type: application/json

{
    "area": {
        "type": "Polygon",
        "coordinates": [
        [
            { "latitude": 34.12345, "longitude": -118.12345 },
            { "latitude": 34.67890, "longitude": -118.67890 },
            { "latitude": 34.54321, "longitude": -118.54321 }
        ]
    ]
},
    "startTime": "2024-07-17T10:00:00Z",
    "endTime": "2024-07-17T11:00:00Z",
    "precision": 7
}
```

3. Processing and Analyzing Data: Once the API responds with population density data, the drone management system parses the response to identify areas with high population density that may require avoidance during drone operations.

- Upon receiving the API response, the drone management system processes the population density data for the specified area.

- The system identifies areas with high population density that should be avoided during drone operations.

Example API Response (modified from the Population Density YAML file [72]):

```
{
    "status": "SUPPORTED_AREA",
    "timedPopulationDensityData": [
        {
            "startTime": "2024-07-17T10:00:00Z",
            "endTime": "2024-07-17T11:00:00Z",
            "cellPopulationDensityData": [
            {
                "geohash": "001",
                "populationDensityData": {
                "dataType": "DENSITY_ESTIMATION",
                "maxPplDensity": 150,
                "minPplDensity": 30,
                "pplDensity": 60
                }
            },
            {
                "geohash": "002",
                "populationDensityData": {
                    "dataType": "DENSITY_ESTIMATION",
                    "maxPplDensity": 100,
                    "minPplDensity": 40,
                    "pplDensity": 90
                    }
                }
            ]
        }
    ]
}
```

4. Optimizing Flight Paths: Using the population density data from the API, the drone management system dynamically adjusts flight paths. The system can reroute the drone to avoid high-density areas and prioritize low-density areas for more efficient data collection.

- Based on the population density data received, the drone management system implements a dynamic route planning algorithm.
- If a geohash area has a population density exceeding a predefined threshold (e.g., 100 people/km²), the drone reroutes to avoid interference or privacy concerns.
- Areas with lower population density are prioritized to ensure smooth and efficient operations.

Example Route Planning Logic:

- If the population density in a geohash area exceeds a certain threshold (e.g., 100 people/km²), the drone's path is adjusted to avoid densely populated zones.
- Areas with lower population density are prioritized for data collection, ensuring more efficient and smooth operations.

5. Real-Time Updates and Feedback: To maintain up-to-date operational data, the drone management system periodically fetches updated population density information, adjusting its flight path in real-time as conditions change.

- Periodically fetch updated population density data from the API to ensure the drone's operations stay aligned with real-time conditions.
- Integrate real-time feedback loops that adjust the drone's mission dynamically based on population density changes or new data received.

By integrating the Population Density API, the drone system can avoid densely populated areas and adjust its flight path in real-time based on changing conditions, optimizing efficiency and enhancing overall operational performance.

6.5. Collision Avoidance API

This section introduces the Collision Avoidance API, a new specialized network API designed to integrate with the 5G core network for real-time drone collision detection and avoidance. The API collects information from various NFs, such as drone location data and the positions of other objects, and provides this data to the drone's control application, enabling effective collision avoidance.

Unlike existing standard APIs, this API leverages advanced network capability exposure, granting precise control over drone operations and safety mechanisms. It serves as an intermediary between the NEF in the 5GC and Unmanned Aircraft System Traffic Management (UTM) systems, aggregating critical data from multiple sources to detect and prevent potential collisions. The API integrates data from radar, GPS, and 5G positioning systems, along with real-time updates from the RAN and the 5GC, ensuring comprehensive situational awareness for drones.

As outlined in [73], long-range radar (LRR) operating in the 77 GHz band is widely deployed in automotive applications, such as adaptive cruise control and forward collision warning. Automotive millimeter-wave (mmWave) radar technology, particularly at 77 GHz, has become a key enabler for vehicular functions like collision avoidance and autonomous driving systems. The integration of Joint Communication and Sensing (JCAS) systems in vehicular platforms has gained significant interest, driven by the rise of autonomous vehicles. In this context, IEEE 802.11ad-based mmWave technology at 77 GHz has been explored as a promising candidate to unify both communication and radar capabilities on the same platform.

With the advent of 6G technology, the convergence of radio and radar transmissions is expected to bring further innovation. Radar systems operating at this frequency can detect nearby objects and their velocities with sub-centimeter accuracy, a crucial factor for enhancing collision avoidance. This high-precision radar data can be seamlessly integrated into existing APIs, providing real-time, precise information on the location, velocity, and trajectory of surrounding objects, thus significantly improving collision detection and avoidance algorithms.

The NEF exposes the following real-time data via the Collision Avoidance API:

Drone Data:

- GPS Coordinates: Latitude, longitude, and altitude.
- Flight Path: Trajectory and waypoints.
- Velocity: Real-time speed and direction vectors.
- Status: Battery levels, sensor health, and connectivity metrics.

RAN Data:

- Network Quality Metrics: SNR, RSRP, RSRQ, and handover status.
- Cell Load: Congestion in serving and adjacent cells.

5GC Data:

- Precise Location: Using 5G positioning methods like Observed Time Difference of Arrival (OTDOA).
- Mobility Events: Handover and movement between network slices.
- Session QoS: Latency, jitter, and packet loss data.

External applications, such as UTM systems, use this integrated dataset for collision detection, analyzing the drone's position, velocity, and proximity to other objects. These systems then make informed decisions regarding trajectory adjustments and avoidance maneuvers, ensuring safe drone operations in complex airspaces.

6.5.1. Parameters Exposed by the API

Key parameters exposed by the API include:

Parameters from Drone:

- LRR Data parameters [74]: Frequency band, update rate in Hz, and detection range in km.
- GPS Coordinates: Geospatial information indicating the drone's precise latitude and longitude, as well as altitude.

Location Information from the 5GC-enabled *Nlmf_Location Service* API [75] encompasses:

- Geographic Area: Represented in various formats such as points, circles, ellipses, polygons, and arcs, each with associated uncertainty parameters for accurate positioning.
- Velocity Estimate: Speed and direction vectors providing real-time information on the drone's movement.
- Altitude: Vertical distance above a reference point, crucial for precise three-dimensional positioning.
- Positioning Methods: Utilizes techniques supported by the *Nlmf_Location* API such as Global Positioning System (GPS), Global Navigation Satellite System (GLONASS), Wireless Local Area Network (WLAN), and others, ensuring diverse and accurate location determination.
- Accuracy and Uncertainty: Metrics indicating the reliability and precision of location data obtained through the LMF API.

These parameters, exposed to third-party applications, enable comprehensive collision analysis, risk assessment, and avoidance decision-making, ensuring safer and more efficient drone operations across various environments.

6.5.2. Architecture for Collision Avoidance API

The architecture diagram of the Collision Avoidance API is shown in Figure 6.10. The roles of the components within the architecture are summarised as follows:

Drone: The drone is equipped with various sensors (e.g., cameras, radar, GPS) that continuously collect data related to its position, velocity, altitude, and environmental conditions. This data is transmitted to the UTM.

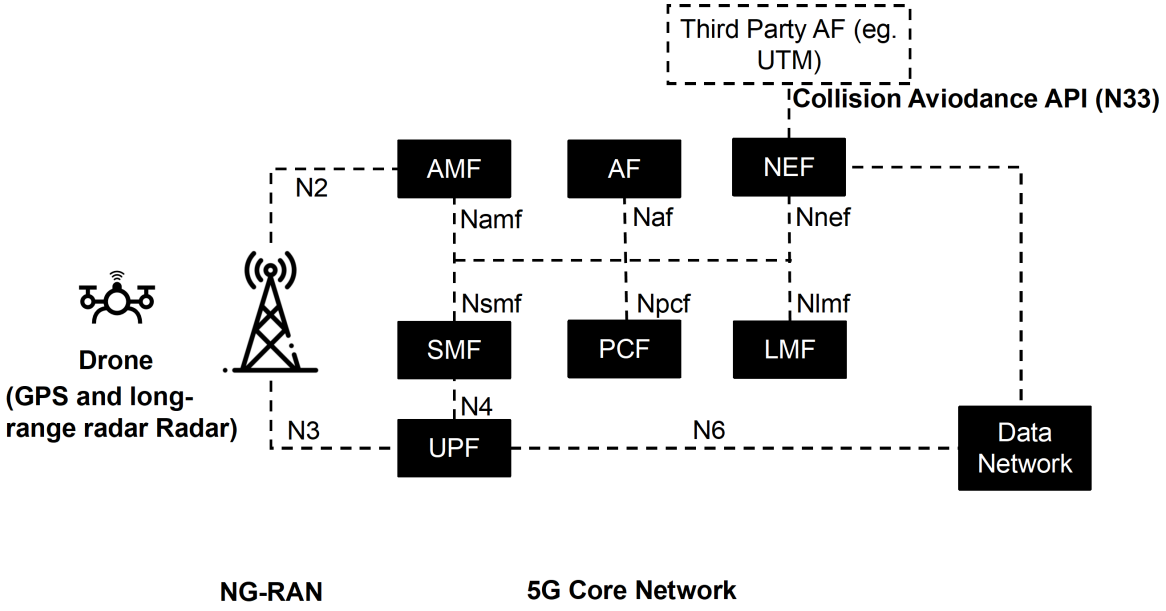


Figure 6.10: Architecture Diagram for Collision Avoidance API

NG-RAN: The NG-RAN facilitates the initial communication between the drone and the core network. It handles both user plane data (sensor data) and control plane messages (commands and acknowledgments).

UPF: The UPF is responsible for routing user plane data. It forwards the drone's sensor data to the NEF and ensures secure and efficient data transmission.

AMF: The AMF manages control plane operations such as registration, connection management, and mobility. It ensures that the drone remains connected to the network and can communicate effectively.

NEF: The NEF serves as an interface for exposing network capabilities to third-party applications. It houses the Collision Avoidance API, which relays sensor data from the UPF to the third-party application and the Collision Analysis Engine. It also handles control plane signaling related to the Collision Avoidance API.

Third-Party Application Function: The third-party application utilizes the data provided via the Collision Avoidance API. It conducts real-time analysis and decision-making to adjust the drone's flight pattern based on the processed data from the Collision Analysis Engine. It sends back control messages to the drone via the API.

Control Plane Message Flow:

- The drone sends control plane messages to the RAN, which are then forwarded to the AMF.
- The AMF communicates with the NEF to manage session and mobility aspects by exchanging signaling messages and data through defined APIs. During the drone's registration, the AMF establishes a session context and may request network resource information, such as QoS parameters. As the drone moves, the AMF monitors its location and connectivity, notifying the NEF of any handovers or operational changes, which enables timely updates to third-party applications like UTM systems.
- Control messages are exchanged between the NEF and third-party applications to ensure proper management and operation of the drone.

User Plane Data Flow:

- The drone transmits application data (e.g., GPS coordinates, velocity, flight path) through its 5G radio interface to the gNB.
- The gNB forwards the user plane data over the 5G network to the UPF.
- The UPF routes and forwards the payload directly to the NEF.
- The LMF sends the drone's location data to the NEF via a service-based interface (Nlmf).
- The NEF, after receiving location data from the LMF and the UPF, exposes the data to authorized external applications via the Collision Avoidance API.

6.5.3. Collision Avoidance Process

The Collision Avoidance API follows a multi-step process to provide the required information:

1. Third-Party Authentication and Authorization: The third-party application undergoes authentication and authorization using OAuth 2.0 to ensure secure access to the Collision Avoidance API.
2. Third-Party Interaction with NEF: The third-party application provides information to the NEF regarding the identifier for the drone, which could be an IPv4 address, IPv6 address, phone number, or Network Access Identifier.
3. NEF Interaction with AMF and RAN: The NEF passes this control plane message to the AMF via a service-based interface within the core network. The AMF communicates with the RAN to identify the drone based on the provided identifier.
4. Signaling Message to Notify Application: Once the drone is identified, a signaling message is sent back through the network to notify the third-party application that the drone can be monitored for collision prevention.
5. Drone Data Transmission: The drone collects and transmits user plane data (e.g., GPS coordinates, velocity) through the RAN to the UPF for forwarding.
6. NEF Receives Data from LMF: The NEF receives data related to the drone (e.g., location data) from the LMF via a service-based interface.
7. NEF Interaction with Collision Avoidance API: The NEF combines control plane data (e.g., location data from the LMF) and relevant information from the data network and sends this data to the Collision Avoidance API for further analysis.
8. Data Relay to Third-Party App: The Collision Avoidance API relays the analyzed data (e.g., positioning updates, mobility events) to the third-party application for collision avoidance purposes.

The process using JSON coding is outlined as follows:

Preamble: OAuth 2.0 Authentication between NEF and UTM

Before any communication occurs between the NEF and UTM, the NEF must first obtain an access token from the UTM using OAuth 2.0. This token will be used to authenticate subsequent requests. This applies to all API usage with the UTM.

Step P1: NEF Requests Token from UTM NEF sends an HTTP POST request to UTM's token endpoint with the following JSON data element:

```
1 {
```

```
"grant_type": "client_credentials",
"client_id": "NEF_client_id",
"client_secret": "NEF_client_secret",
"scope": "api_access"
}
```

Step P2: UTM Responds with Access Token UTM responds with HTTP 200 OK and the following payload:

```
{
"access_token": "eyJhbGciOiJSUzI1NiIsInR...",
"token_type": "Bearer",
"expires_in": 3600
}
```

The NEF stores this token and uses it for authenticating all subsequent requests from this application.

Step 1: NEF Registers with UTM

UTM sends an HTTP POST request to NEF's `/register` endpoint with the following payload:

```
{
"utm_id": "UTM123",
"utm_api_endpoint": "https://utm.example.com/api", "utm_credentials": "
    base64_encoded_credentials"
}
```

NEF responds with HTTP 200 OK and the following payload:

```
{
"nef_id": "NEF456",
"nef_api_endpoint": "https://nef.example.com/api",
"nef_credentials": "base64_encoded_credentials"
}
```

UTM and NEF validate the provided credentials using stored public keys or certificates and store each other's endpoints for future secure communication.

Step 2: Drone Registers with Core Network (UPF)

Drone Initiates Registration The drone initiates registration using NAS (Non-Access Stratum) signaling: NAS Registration Request.

The AMF authenticates the drone and responds with NAS Registration Accept.

Drone Sends Registration Information The drone sends its own IP address and port via HTTP POST to the AMF:

```
{
"drone_id": "Drone789",
"drone_ip": "192.168.1.10",
"drone_port": 5000
}
```

AMF Provides NEF Information to Drone The AMF sends an HTTP POST to the drone's registration endpoint, providing NEF's IP address and port:

```
{
"nef_ip": "203.0.113.10",
"nef_port": 8080
}
```

Step 3: AMF Notifies NEF of Drone Registration
The AMF sends an HTTP POST request to NEF's `/notifyRegistration` endpoint with the following payload:

```
{
"drone_id": "Drone789",
"drone_ip": "192.168.1.10",
"drone_port": 5000,
"session_id": "PDU12345"
}
```

NEF creates a new process instance for the drone identified by `Drone789` and acknowledges the notification with HTTP 200 OK:

```
{
"status": "registered",
"process_instance_id": "Instance123"
}
```

Step 4: Data Transmission Path
UTM Requests Collision Avoidance Data UTM sends an HTTP POST request containing the drone's ID and IP address to NEF for collision avoidance information via the collision avoidance API:

```
{
"drone_id": "Drone789",
"drone_ip": "192.168.1.10"
}
```

NEF Interaction with LMF NEF interacts with the LMF to retrieve location data about the drone, using the provided drone ID.

NEF Forwards Request to UPF NEF forwards the collision avoidance request to the the UPF via HTTP POST to the UPF.

UPF Acknowledges Request and Prepares for Data Transmission The UPF acknowledges the collision avoidance request by sending an HTTP 200 OK response to the NEF and prepares for user plane data transmission by configuring the necessary routing and processing rules.

UPF Notifies Drone to Send Data UPF sends an HTTP POST request to the drone to notify it to start sending radar and GPS data:

```
{
"message": "start data transmission",
"target_ip": "203.0.113.10",
"target_port": 8080
}
```

Drone Sends Data to UPF The drone sends UDP/TCP data packets (e.g., radar data and GPS data) to the UPF via its IP address and port (192.168.1.10:5000).

UPF Handles Data Transmission The UPF forwards the data packets to the NEF using the established process instance via HTTP POST:

```
{
"process_instance_id": "Instance123",
"data": "base64_encoded_location_data"
}
```

NEF Relays Control Plane Data to UTM The NEF relays the control plane data (e.g., location updates) to the UTM via the collision avoidance API at UTM's `/receiveData` endpoint:

```
{
"drone_id": "Drone789",
"data": "base64_encoded_location_data"
}
```

In summary, the Collision Avoidance API functions within the network infrastructure, facilitating the exchange of information between the drone's sensors and third-party applications or systems involved in collision analysis and avoidance.

6.6. Conclusion

This chapter presents a connected drone use case to illustrate the utilization of network capability exposure for data transmission, navigation, and collision avoidance. While analyzing the application of network APIs in this scenario, it became evident that the current implementation does not fully address all aspects of network capability exposure. To fill this gap, Section 6.4 introduces and details two existing CAMARA APIs, demonstrating how their integration could enhance the use case. Furthermore, based on this investigation, a new network capability exposure API — Collision Avoidance API is proposed. This API improves drone safety by providing the UTM with location data from the drone, information on the location and velocity of other objects, and relevant data from the 5G network, thereby enhancing collision avoidance measures.

7

Conclusion and Future Work

This concluding chapter summarizes the key findings of this thesis and offers recommendations for future research. Section 7.1 presents the general conclusions derived from the previous chapters of the thesis. Section 7.2 discusses recommendations and potential directions for further research.

7.1. Conclusion

This thesis investigates network capability exposure in 5G, focusing on two use cases introduced in chapters 5 and 6. It explores how and where network capability exposure can be applied to enhance these scenarios.

The use case in Chapter 5 involves an enhanced call using AR/VR technology, where a field technician utilizes 5G while a remote expert is connected via wireline. The objective is to investigate how call control capability exposure can be employed to integrate commands into a video replica. Among the four presented implementation options, a media server within the IMS network is chosen for this purpose.

However, current capability exposure is insufficient for this implementation. To address this, a new capability exposure—call control capability exposure—is proposed, operating between a third-party application and the NEF. After receiving instructions from UE-A/UE-B via WebRTC, the third-party application first directs the media server to allocate resources for video editing. It then instructs the MMTEL-AS in the control plane to route the replica video from UE-A to the selected media server in the user plane.

In general, call control capability can be used by third-party applications to manage user plane processing. For example, in the call recording scenario introduced in Chapter 5, a third-party application could use this capability exposure to route voice data to a media server for recording and later retrieve the recorded audio.

Chapter 6 introduces a specialized API: the Collision Avoidance API, designed to prevent drone collisions with other drones or objects. This API provides location and velocity data from the drone's radar, detailing the positions of both the drone and any detected objects. This process begins with interactions between the NEF and third-party applications such as UTM. First, an authentication and authorization process takes place. Once the UTM is verified, the NEF registers with it, and the specific registration occurs with the UPF. The UPF then provides the NEF with the drone's IP address.

When the UTM requests collision avoidance data, the NEF forwards the request via the 5GC to the drone. The drone then sends its radar information and the positions and velocities of detected objects

through the RAN to the 5GC. In conjunction with other control plane network functions, such as the SMF and PCF, this data is transmitted to the NEF for exposure. Additionally, location information regarding the drone is sent from the LMF to the NEF via the service-based interfaces, providing an extra source of location data. These two sets of information are consolidated in the NEF and exposed to third-party applications, such as UTM, via the Collision Avoidance API for analysis and decision-making.

In conclusion, Chapter 6 presents a detailed theoretical framework for the Collision Avoidance API, emphasizing its architecture, workflow, and JSON coding for practical implementation. By facilitating the exchange of location and velocity data from drone radars to third-party applications like UTM, the API enhances collision detection and avoidance mechanisms.

7.2. Future Work

A logical next step for future work may be to expand the types and sources of location data for the Collision Avoidance API. By exposing additional location data and improving the reliability of sources, the accuracy of the drone's situational awareness could be significantly enhanced. Furthermore, optimizing the collision avoidance process to reduce response time is essential, as minimizing delays directly impacts drone safety and operational efficiency.

In terms of call control capability exposure, further exploration of its potential use cases is needed, particularly in how controlling user plane processing through network capability exposure can be applied to specific scenarios. Identifying additional applications where this capability could be leveraged, and understanding whether network elements can be adapted for various use cases, will be important steps in expanding its functionality.

Additionally, a more comprehensive analysis of Network Capability Exposure use cases across different standards—such as 3GPP, O-RAN, and GSMA—could provide valuable insights. Investigating how these standards interact and complement each other will contribute to a better understanding of their potential to enhance user experience and improve network performance.

References

[1] Ericsson. *5G WIRELESS ACCESS: AN OVERVIEW*. Tech. rep. Telefonaktiebolaget LM Ericsson, (2023).

[2] Huawei. *5G Network Architecture-A High Level View*. Tech. rep. HUAWEI TECHNOLIGIES CO., LTD., (2016).

[3] M. Intelligence. *Telecom API Market Size Share Analysis - Growth Trends Forecasts (2024 - 2029)*. 2023. URL: https://www.mordorintelligence.com/industry-reports/telecom-api-market.

[4] I. O. W. Group. *OAuth Core 1.0*. URL: https://oauth.net/core/1.0/.

[5] 3GPP. *Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Architecture enhancements to facilitate communications with packet data networks and applications (Release 15)*. Tech. rep. 3GPP TS 23.682 v15.5.0 (2018.07). 3GPP.

[6] 3GPP. *Universal Mobile Telecommunications System (UMTS); LTE; 5G; T8 reference point for Northbound APIs (Release 17)*. Tech. rep. 3GPP TS 29.122 v17.5.0 (2022.05). 3GPP.

[7] M. Starsinic et al. “An Overview of 3GPP Exposed Services for IoT Service Platforms”. In: *GetMobile: Mobile Comp. and Comm.* 22.2 (Sept. 2018), pp. 16–21. DOI: 10.1145/3276145.3276153.

[8] K. Samdanis et al. “From network sharing to multi-tenancy: The 5G network slice broker”. In: *IEEE Communications Magazine* 54.7 (2016), pp. 32–39. DOI: 10.1109/MCOM.2016.7514161.

[9] 3GPP. *5G; 5G System; Network Exposure Function Northbound APIs; Stage 3 (Release 15)*. Tech. rep. 3GPP TS 29.522 v15.3.0 (2019.04). 3GPP.

[10] M. Gramaglia et al. “A unified service-based capability exposure framework for closed-loop network automation”. In: *Transactions on Emerging Telecommunications Technologies* 33.11 (2022), e4598. DOI: https://doi.org/10.1002/ett.4598. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.4598.

[11] 3GPP. *Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; IP Multimedia Subsystem (IMS); Stage 2 (Release 16)*. Tech. rep. 3GPP TS 23.228 v16.5.0 (2020.10). 3GPP.

[12] L. Lin et al. “A Novel 5G Core Network Capability Exposure Method for Telecom Operator”. In: *2020 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/-SustainCom)*. 2020, pp. 1450–1454. DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00217.

[13] Y. Yu. “The Mobile Network Capability Exposure Friendly to the Mobile Internet Applications”. In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. 2017, pp. 1–6. DOI: 10.1109/WCNC.2017.7925816.

[14] P. Li et al. “Capability Exposure Vitalizes 5G Network”. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. 2021, pp. 874–878. DOI: 10.1109/IWCMC51323.2021.9498666.

[15] L. Lin et al. “5G-A Capability Exposure Scheme based on Harmonized Communication and Sensing”. In: *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 2022, pp. 1036–1041. DOI: 10.1109/TrustCom56396.2022.00141.

[16] D. Fragkos et al. “NEFSim: An open experimentation framework utilizing 3GPP’s exposure services”. In: *2022 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*. 2022, pp. 303–308. DOI: 10.1109/EuCNC/6GSummit54941.2022.9815829.

[17] Open5GS. *Open5GS: Open source 5G core network stack*. https://github.com/open5gs. 2024.

[18] G. Makropoulos et al. “5G and B5G NEF exposure capabilities towards an Industrial IoT use case”. In: *2023 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*. 2023, pp. 647–651. DOI: 10.1109/EuCNC/6GSummit58263.2023.10188241.

[19] G. C. Inc. *GL Communication Introduces 5G Protocol Analyzer for Network Monitoring*. 2023. URL: https://www.gl.com/press-release/5g-protocol-analyzer-press-release.html.

[20] 3GPP. *System architecture for the 5G System (5GS) (Release 16)*. Tech. rep. 3GPP TS 23.501 v16.6.0 (2020-10). 3GPP.

[21] 3GPP. *5G; NG-RAN; Architecture description (Release 16)*. Tech. rep. 3GPP TS 38.401 v16.3.0 (2020-11). 3GPP.

[22] M. Säily et al. *5G-Xcast: RAN Logical Architecture and Interfaces for 5G-Xcast*. Tech. rep. 5G-Xcast Project, 2019.

[23] 3GPP. *Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Network architecture (Release 15)*. Tech. rep. 3GPP TS 23.002 version 15.0.0 (2018.07). 3GPP.

[24] 3GPP. *Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 16)*. Tech. rep. 3GPP TS 24.229 v16.6.0 (2020.07). 3GPP.

[25] T. communications. *Voice Over 5G | The 5G Zone*. URL: https://the5gzone.com/index.php/voice-over-5g/.

[26] Ericsson. *Ericsson Virtual Multimedia Telephony Application Server*. URL: https://marketplace.cloud.vmware.com/services/details/ericsson-virtual-multimedia-telephony-application-server-1-7/?slug=true.

[27] 3GPP. *Procedures for the 5G System (5GS) (Release 15)*. Tech. rep. 3GPP TS 23.502 v15.4.1 (2019-3). 3GPP.

[28] 3GPP. *5G ; 5G System; User Plane Function Services; Stage 3 (Release 17)*. Tech. rep. 3GPP TS 29.564 v17.1.0 (2022.07). 3GPP.

[29] 3GPP. *5G; 5G System Enhancements for Edge Computing; Stage 2 (Release 17)*. Tech. rep. 3GPP TS 23.548 v17.2.0 (2022.05). 3GPP.

[30] EVOLVED-5G. *Deliverable D4.1 5G Exposure Capabilities for Vertical Applications (Intermediate)*. Tech. rep. EVOLVED-5G, (2021).

[31] 3GPP. *5G; 5G System; Binding Support Management Service; Stage 3 (Release 17)*. Tech. rep. 3GPP TS 29.521 v17.6.0 (2022.09). 3GPP.

[32] 3GPP. *5G; 5G System; Policy Authorization Service; Stage 3 (Release 17)*. Tech. rep. 3GPP TS 29.514 v17.5.0 (2022.06). 3GPP.

[33] 3GPP. *5G; 5G System; Policy and Charging Control signalling flows and QoS parameter mapping; Stage 3 (Release 16)*. Tech. rep. 3GPP TS 29.513 v16.8.0 (2021.08). 3GPP.

[34] O-RAN.WG1. *O-RAN Architecture Description*. Tech. rep. O-RAN-Architecture-Description-v06.00. O-RAN.

[35] 3GPP. *5G; NR; Overall description; Stage-2 (Release 15)*. Tech. rep. 3GPP TS 38.300 v15.3.1 (2018-10). 3GPP.

[36] 5gworldpro.com. *O-RAN : 3GPP Vs O-RAN Alliance*. 2021. URL: https://www.5gworldpro.com/blog/2021/08/08/o-ran-3gpp-vs-o-ran-alliance/.

[37] O.-R. Alliance. *O-RAN Architecture Overview*. 2019. URL: https://docs.o-ran-sc.org/en/latest/architecture/architecture.html#components-definition.

[38] VMware. *Envisioning Service Management and Orchestration for 5G Toward a Modular Multi-Vendor, Multi-Cloud SMO Spanning Beyond RAN to Core and Edge*. Tech. rep. VMware, Inc., (2022).

[39] R. W. News. *Non-real time RIC vs. near-real time RIC*. 2021. URL: https://www.rcrwireless.com/20211129/fundamentals/non-real-time-ric-vs-near-real-timeric#:~:text=The%20near%20RT%20portion%20manages,or%20on%20the%20network%20edge..

[40] O.-R. Alliance. *O-RAN Architecture Overview*. 2019. URL: https://docs.o-ran-sc.org/en/latest/architecture/architecture.html.

[41] O-RAN. *O-RAN.WG1.Use-Cases-Detailed-Specification-R003-v12.00*. Tech. rep. O-RAN ALLIANCE, 2023.

[42] O-RAN. *O-RAN Work Group 1 (Use Cases and Overall Architecture) O-RAN Architecture Description O-RAN.WG1.OAD-R003-v12.00*. Tech. rep. O-RAN ALLIANCE, 2024.

[43] O-RAN.WG3. *Near-Real-time RAN Intelligent Controller Use Cases and Requirements*. Tech. rep. O-RAN.WG3.UCR-R003-v04.00. O-RAN.

[44] GSMA. *Operator Platform Concept – Phase 1: Edge Cloud Computing*. Tech. rep. (2020-01). GSMA.

[45] GSMA. *Telco Edge Cloud: Edge Service Description Commercial Principles Whitepaper*. Tech. rep. v1.0 (2020-10). GSMA.

[46] GSMA. *Operator Platform Telco Edge Proposal*. Tech. rep. v1.0 (2020-10). GSMA.

[47] GSMA. *Operator Platform Telco Edge Requirements*. Tech. rep. v3.0 (2022-10). GSMA.

[48] CAMARA. *The Linux Foundation Projects - CAMARA*. 2023. URL: https://camaraproject.org/.

[49] Orange. *CAMARA 1.0*. 2024. URL: https://developer.orange.com/apis/camara.

[50] GSMA. *The Ecosystem for Open Gateway NaaS API development*. Tech. rep. GSMA, 2023.

[51] GSMA. *Operator Platform Group*. URL: https://www.gsma.com/solutions-and-impact/technologies/networks/operator-platform-hp/.

[52] CAMARA. *CAMARA Project - Scope*. 2023. URL: https://camaraproject.org/scope/.

[53] J. Ordonez-Lucena et al. “Pathways towards network-as-a-service: the CAMARA project”. In: *Proceedings of the ACM SIGCOMM Workshop on Network-Application Integration*. 2022, pp. 53–59.

[54] GSMA. *IMS Data Channel White Paper*. Tech. rep. v1.0 (2021-12). GSMA.

[55] 3GPP. *Universal Mobile Telecommunications System (UMTS); LTE; 5G; IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction (Release 16)*. Tech. rep. 3GPP TS 26.114 v16.6.1 (2020.09). 3GPP.

[56] GSMA. *NG.134 IMS Data Channel v1.0*. Tech. rep. GSMA, (2023-04).

[57] N. Elron. *Unlocking the Sky: Cellular Connectivity for Commercial Drones*. 2024. URL: https://webbingsolutions.com/unlocking-the-sky-cellular-connectivity-for-commercial-drones/.

[58] Ericsson. *Network exposure and the case for connected drones*. 2020. URL: https://www.ericsson.com/en/blog/2020/6/network-exposure-and-the-case-for-connected-drones.

[59] I. O. W. Group. *OAuth 2.0*. URL: https://oauth.net/2/.

[60] Ericsson. *Network exposure APIs in 5G expand connected drone capabilities*. Tech. rep. Telefonaktiebolaget LM Ericsson, (2023).

[61] 3GPP. *5G; Security architecture and procedures for 5G System (Release 17)*. Tech. rep. 3GPP TS 33.501 v17.5.0 (2022.05). 3GPP.

[62] CAMARA. *CAMARA APIs access and user consent management*. 2024. URL: https://github.com/camaraproject/IdentityAndConsentManagement/blob/main/documentation/CAMARA-API-access-and-user-consent.md.

[63] Orange. *CAMARA - Number Verification - France*. 2024. URL: https://developer.orange.com/apis/camara-number-verification-france/api-reference.

[64] CAMARA. *QoD for enhanced communication (0.10.1)*. 2024. URL: https://github.com/camaraproject/QualityOnDemand/tree/main.

[65] CAMARA. *DeviceLocation*. 2024. URL: https://github.com/camaraproject/DeviceLocation.

[66] CAMARA. *Location retrieval API (0.1.0)*. 2024. URL: https://editor.swagger.io/?url=https://raw.githubusercontent.com/camaraproject/DeviceLocation/release-v0.2.0/code/API_definitions/location-retrieval.yaml.

[67] GSMA. *GSMA Open Gateway API Descriptions*. 2024. URL: https://www.gsma.com/solutions-and-impact/gsma-open-gateway/gsma-open-gateway-api-descriptions/.

[68] Ericsson. *5G positioning: What you need to know*. 2020. URL: https://www.ericsson.com/en/blog/2020/12/5g-positioning--what-you-need-to-know.

[69] 3GPP. *5G; 5G System (5GS) Location Services (LCS); Stage 2 (Release 16)*. Tech. rep. 3GPP TS 23.273 v16.7.0 (2021.07). 3GPP.

[70] CAMARA. *Population Density Data*. 2024. URL: https://github.com/camaraproject/PopulationDensityData/tree/main.

[71] CAMARA. *Population Density Data (0.1.0-wip)*. 2024. URL: https://redocly.github.io/redoc/?url=https://raw.githubusercontent.com/camaraproject/PopulationDensityData/main/code/API_definitions/population-density-data.yaml&nocors.

[72] CAMARA. *Population Density Data API YAML Definition*. 2024. URL: https://raw.githubusercontent.com/camaraproject/PopulationDensityData/main/code/API_definitions/population-density-data.yaml.

[73] P. Kumari et al. "IEEE 802.11ad-Based Radar: An Approach to Joint Vehicular Communication-Radar System". In: *IEEE Transactions on Vehicular Technology* 67.4 (2018), pp. 3012–3027. DOI: 10.1109/TVT.2017.2774762.

[74] J. Gong et al. "Detection of Micro-Doppler Signals of Drones Using Radar Systems with Different Radar Dwell Times". In: *Drones* 6 (Sept. 2022), p. 262. DOI: 10.3390/drones6090262.

[75] 3GPP. *5G; 5G System; Location Management Services; Stage 3 (Release 16)*. Tech. rep. 3GPP TS 29.572 v16.6.0 (2021.04). 3GPP.