6GSNS

Co-funded by the European Union

SAFE-6G

SAFE-6G

A Smart and Adaptive Framework for Enhancing Trust in 6G Networks

Deliverable D3.3: User-centric Distributed 6G Core

Date: 17/07/2026

Version: V1.0

DISCLAIMER

Grant Agreement	101139031
Document number	D3.3
Document title	User-centric Distributed 6G Core
Lead Beneficiary	CMC
Editor(s)	Jose Costa-Requena (CMC) Alejandro Fornés (UPV) Spyridon Georgoulas (NCSRD)
Author(s)	Jose Costa-Requena (CMC) Panagiotis Pavlidis (NCSRD) Spyridon Georgoulas (NCSRD) Dimitrios Roidis (NCSRD) Harilaos Koumaras (NCSRD)
Dissemination level	Public
Contractual date of delivery	30/06/2026
Status	Final
File name	SAFE-6G_D3.3_V1.0.pdf

Revision History

Version	
V0.1	TOC proposal, guidelines and templates
V0.2	Contributions from CMC
V0.3	Second round of contributions from CMC after WP3 leader feedback
V0.4	Contributions from NCSRD
V0.8	Internal Review comments (UPV, EBOS) addressed
V0.9	Additional review from TM (NCSRD), for alignment purposes
V1.0	Final version ready for submission

GLOSSARY

Abbreviations/Acronym	Description
5GC	5G Core
5QI	5G QoS Identifier (5QI)
AF	Application Function
AMF	Access and Mobility Function
AS	Application Session
AUSF	Authentication Server Function
AVX	Advanced Vector Extensions
CAPIF	Common API Framework
CNC	Cumucore Network Configuration
CN-NSSMS	Core Network Slice Subnet Management System
CNW	CMC Network Wizard
CPE	Customer Premises Equipment
DN	Data Network
DS-TT	Device-Side TSN Translator
eBPF	Extended Berkeley Packet Filter
EPC	Evolved Packet Core
FQDN	Fully Qualified Domain Name
gNB	5G NR Node B
GTP-U	GPRS Tunnelling Protocol User Plane
GUI	Graphical User Interface
LAN	Local Area Network
LTE	Long Term Evolution
NEF	Network Exposure Function
NETCONF	Network Configuration Protocol
NIC	Network Interface Card
NF	Network Function
NRF	Network Repository Function
NSACF	Network Slice Access Control Function
NSMF	Network Slice Management Function
NSMS	Network Slice Management System
NSSF	Network Slice Selection Function
NWDAF	Network Data Analytics Function
NW-TT	Network TSN Translator
O&M	Operation and Management
PCF	Policy Control Function
PF	Policy Function
PFCP	Packet Forwarding Control Protocol
QoS	Quality of Service
RAN	Radio Access Network
RAN-NSSMS	RAN Network Slice Subnet Management System

SA	Standalone
SBA	Service-Based Architecture
SEPP	Security Protection Proxy
SM	Session Management
SMF	Session Management Function
SUPI	Subscriber's Permanent Identifier
TP-NSSMS	Transport Network - Slice Subnet Management System
TSN	Time Sensitive Networking
TSN-AF	TSN Application Function
UDM	Unified Data Management
UDR	Unified Data Repository
UE	User Equipment
UPF	User Plane Function

EXECUTIVE SUMMARY

This document is the deliverable of WP3 - D3.3: *User-centric Distributed 6G Core*. This deliverable provides a technical documentation of the user-centric distributed 5G/6G core, detailing the implementation details, as well as providing the software components needed for validating the SAFE-6G solution with the Metaverse use cases.

Specifically, D3.3 describes the technologies, modules and interfaces that are part of the 6G core design as part of the SAFE-6G system to provide the expected trustworthy-related features. This entails the core, and the Metaverse modules that will be used to construct the use cases to validate the SAFE-6G system. Thus, the document defines how all building blocks of the SAFE-6G packet core architecture are integrated together to deliver 6G core to support Metaverse use cases.

KEYWORDS

5G, 6G, User Centric distributed 6G Core, CAPIF, Open5GS

TABLE OF CONTENTS

List of FIGURES

List of TABLES

1 INTRODUCTION

Figure 1 highlights in green the building blocks of the SAFE-6G architecture implemented in WP3. This deliverable focuses on the yellow blocks; this is, the developments and implementations performed on the **core network** plane, one of the three underlying layers in which the SAFE-6G framework acts upon (jointly with the services and continuum planes). It should be reminded that this plane is not part of the SAFE-6G framework but key part of the 5G and beyond ecosystem, exposing a set of APIs and endpoints that Trust Functions from WP4, coordinated by the Cognitive Coordinator (CoCo), can consume in order to increase the trustworthiness of the system, based on users' requests.

This deliverable reports the final design choices and the features implemented for Cumucore's private network core – not publicly available since it has a commercial license. For the selected open-source implementation (i.e., Open5GS), the documentation and code of the 2 APIs developed to be exposed through its Network Exposure Function (NEF) – namely NEF Monitoring Event & Application Session (AS) With QoS – are publicly available at https://gitlab.com/safe-6g/development/nef-monitoring-event and https://gitlab.com/safe-6g/development/nef-qos, respectively.

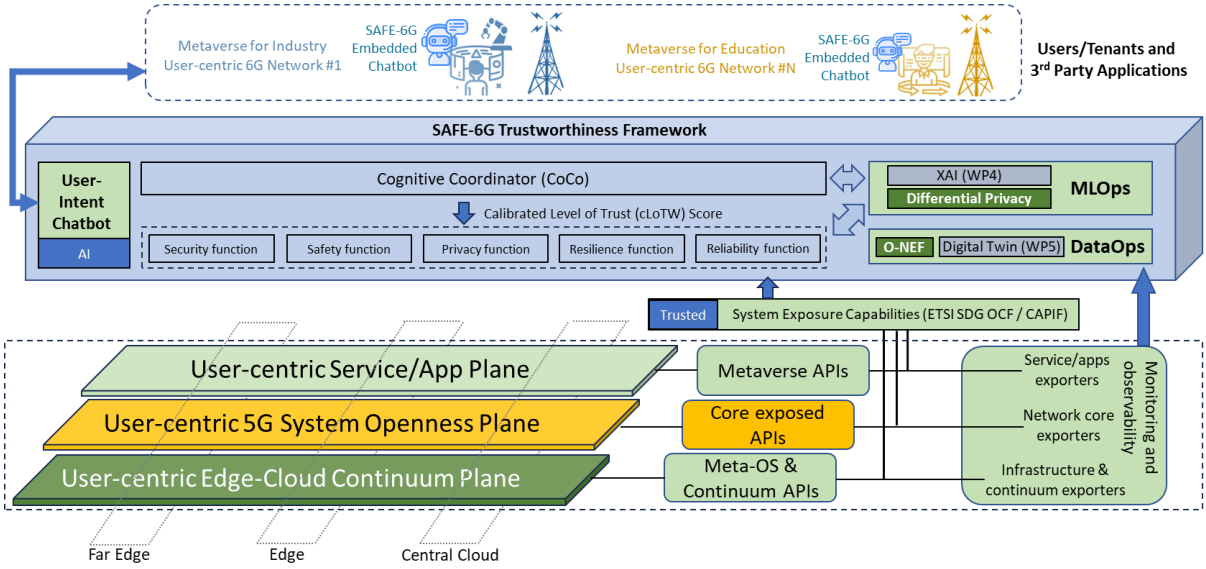

Figure 1: Building blocks and components of the SAFE-6G architecture

1.1 5G MOBILE PACKET CORE PLATFORM

This section serves as an overview of the baseline functionality of the 5G mobile packet core. 5G core evolved from a monolithic architecture in Long Term Evolution (LTE)/4G into a more distributed approach called Service Based Architecture (SBA). The objective is to facilitate the extension of services and functionality by adding new Network Functions (NFs) that will be integrated without major changes. The SBA architecture includes the network functions included in the following diagram which consists of: Access and Mobility Function (AMF), Session Management Function (SMF), User Plane Function (UPF), Network Repository Function (NRF), Authentication Server Function (AUSF), Unified Data Repository (UDR), Network Slice Selection Function (NSSF), Network Slice Access Control Function (NSACF), Network Data Analytics Function (NWDAF), Policy Control Function (PCF), NEF and Security Protection Proxy (SEPP). The NFs relevant for delivering user centric architecture with the

integration of new service network functions are described in Table 1, extending the list with respect to D3.1 as described in following sections, to support WP4 modules and bring additional functionality for Common API Framework (CAPIF) integration.

Component	*Description*	*Interactions*
NRF	Network Repository Function (All NF register and inform available interfaces).	The NRF is the key function to support SBA and register user centric network functions. It also serves as index that can be consulted by the rest of NFs.
AMF	Access and Mobility Function (gNB connection, UE registration & authentication and handover management).	The AMF interacts with security function to support trustworthy modules. It works with AUSF to manage user access and authentication, with AMF managing access and mobility.
SMF	Session Management Function (PDU setup, IP address allocation, UPF location and policy checking).	It receives policy information from PCF to determine if a session should follow any specific requirements in terms of security or Quality of Service (QoS).
UPF	User Plane Function (Data plane, GTP dec/ecap, data usage and QoS collection).	The User Plane Function interfaces with the Control Plane (SMF) with the N4 interface. It receives data from the RAN and can forward it to the Data Network, or to other UEs of the network.
UDM	User Data Management (User data profile management, access user credentials).	The UDM stores the user specific policy requirements for the mobility or session profile. Other functions (e.g., AUSF, AMF, SMF) can request subscriber data to it.
AUSF	Authentication Service Function (Interacts with UDM for authenticating UE).	It allows adding new security algorithms required by the trust functions. It is in charge of authentication and security procedures.
PCF	Policy Control Function stores user specific mobility and session policies.	The PCF checks the policy subscription information stored in the UDM to provide policy rules to network functions (SMF/AMF).
NWDAF	Network Data Analytics Function (collects NF status from NRF, data usage and QoS from UPF and exposes to other NFs).	It provides the interface for collecting data statistics from the UE sessions and the rest of network functions.
NSSF	Network Slice Selection Function (accesses network slice information available and checks with slice information registered to UE).	It provides the list of authorized slices that will be allocated to the UE. It can be modified to restrict access to certain slices based on input of external security network functions.
NEF	Network Exposure Function exposes interfaces from internal NFs to external AF.	NEF is the key function that allows internal and external APIs to apply different QoS or policies.
O&M CNC	Cumucore Network Configuration (stores OSS; BSS information and exposes to NFs).	Implementation of O&M function which is not defined in 5G specifications but is required to interact with the core functions.

Table 1: 5G core network functions

Figure 2 shows the main NFs including the network slicing features included in 5G core Release 16 used as baseline. The above figure also represents the integration of CAPIF to expose the core functionality to external Application Functions (AFs) as proposed in D3.1. In the next section, the updated packet core modules are presented including the usage of NEF to provide external AF access through CAPIF module to the core functionality.

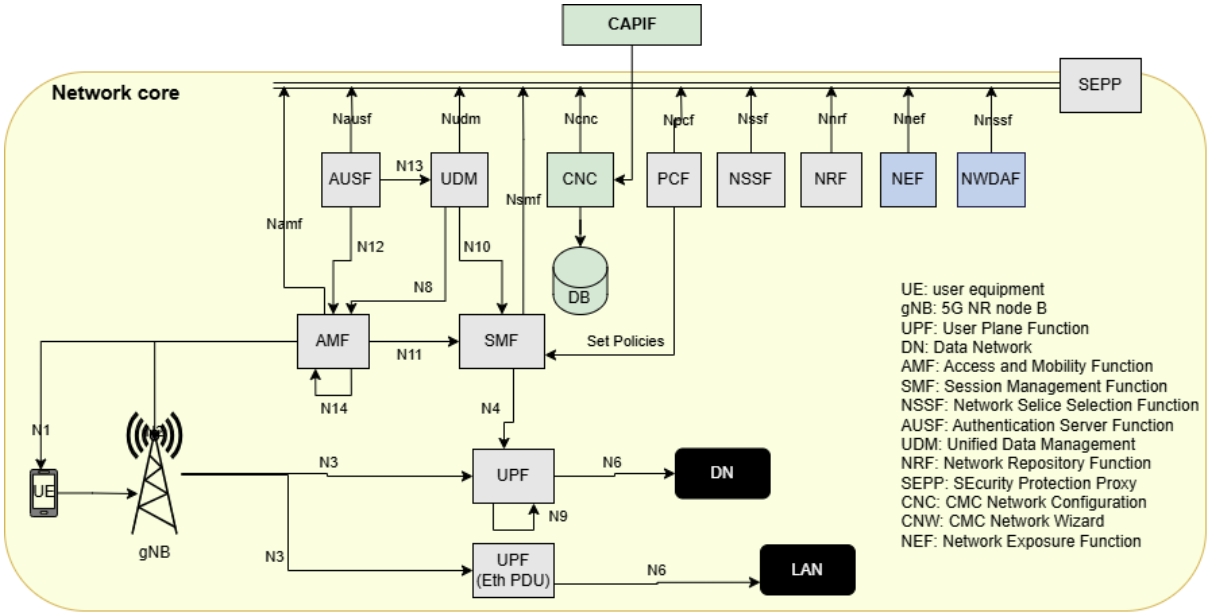

Figure 2: 5G Service Based Architecture blocks and components

1.2 THE RATIONALE BEHIND THE STRUCTURE

The structure of this deliverable is as follows. Introduction has presented the basic architecture of the mobile packet core including the Network Functions (NFs) implemented as part of 5G core, focusing on the WP3-related building blocks for supporting CAPIF and the SAFE-6G components. Section 2 presents the additional NFs defined as part of the 6G core and describes the technical specifications of the addressed NFs. Particularly, Section 2 targets the advanced features currently under design at 3GPP to be part of Rel. 19 and 20. The evolution of mobile packet core to be part of 6G is also outlined. Section 3 presents the open-source mobile packet core Open5GS also utilized in SAFE-6G project extended with additional interfaces for supporting WP4 modules deployed in SAFE-6G. Section 4 presents the detailed implementation of NF interfaces to be exposed for WP4 modules to support the Metaverse use cases. This section focuses on the north and bound main interfaces of the packet core to build up the SAFE-6G platform. Finally, conclusions are drawn in Section 5.

2 6G CORE PLATFORM

Mobile packet core is evolving with 3GPP Rel. 19 and Rel. 20 under design phase towards the 6G core. It will continue with the architecture from 5G, allowing the deployment of user or service specific instances of the 5G/6G network or allocate some of the NFs to different services such as best effort or TSN. The SBA consists of several NFs, all of them interconnected through well-defined REST API interfaces and service endpoints defined in 3GPP. The next core architecture is taking a more distributed approach moving towards a swiss-knife type of solution to support user centric services but also machine type communications.

2.1 6G PACKET CORE SYSTEM

The 6G core modules consist of the standard ones, along with components that will be added as part of the SBA architecture. The 6G core is more flexible to include new NFs to support user centric services, and therefore it should be able to provide the required functionality for consumer data located on public Internet or cloud services. Moreover, the 6G core shall support network slicing and deployment of additional modules, as well as connectivity for future robotics or machine learning applications that require more deterministic and time sensitive communications. The specific development and implementation activities related to the core have been guided by the following requirements stated in D2.1:

- **REQ-COM-ETH-NF-M-03:** Compliance with 5G and industry standards.
- **REQ-COM-DES-F-M-04:** Following Cloud Native paradigm.
- **REQ-COM-DES-F-M-06:** Functions Virtualization.
- **REQ-COM-DES-F-M-19:** Access to performance metrics from 5G/6G core infrastructure.
- **REQ-COM-DES-F-R-24:** Fault Tolerance.
- **REQ-COM-DES-F-M-27:** Easy Maintainability.
- **REQ-COM-DES-F-M-08:** APIs interaction through Open CAPIF.
- **REQ-COM-DES-F-M-19:** Access to performance metrics from 5G/6G core infrastructure.
- **REQ-UCEN-DES-F-M-01:** User centric functionality description.
- **REQ-UCEN-DES-F-M-02:** Standalone NF.
- **REQ-UCEN-DES-F-R-03:** Each NF should be containerized.
- **REQ-UCEN- DES-F-R-04**: Core network shall be able to support auxiliary NFs.
- **REQ-UCEN-COMP-F-M-05**: Core network exposure must be performed via the Open CAPIF.
- **REQ-UCEN-COMP-F-M-06:** The core network must comply with the 3GPP Standardization.
- **REQ-UCEN-COMP-F-M-07:** Core network functions and Trust function interaction must comply with SA6 3GPP specifications.

As a result of the defined requirements and the evolution of the project, the following **features** have been designed and implemented in Task 3.2 (apart from supporting the typical core functions). These are related to new functionalities, openness and exposure, and Cloud Native adaptations:

- Signalling and user plane in containerised, Kubernetes environment.

- Cumucore's core NFs implemented for Fully Qualified Domain Name (FQDN) support to support Cloud deployment.
- Design UPF to use IP, 3GPP compliant FQDN and Kubernetes.
- Dedicated interfaces for integration of AI/ML through NWDAF.
- NEF extension for implementing interfaces for AI/ML.
- Internal API available at CNC exposed to be used by internal AF/Orchestrator.
- NEF API extended for supporting external SAFE-6G AF.
- Development of TSN UPF, TSN-AF and customer UPF.
- Design interface to deploy and register multiple instances of UPF with different features: UPF 1 (IP PDU), UPF 2 (Ethernet PDU), etc.

Moreover, the 6G Core will expose through the NEF the additional functionalities and thus will grant access to the new modules integrated in the 6G network (see next subsection). The 6G core will be oriented to end user considering both personal communications but also machine learning applications that require access to internal information from the core about connectivity levels, delay, throughput, etc. Figure 3 shows the multi-facet of the core that is exposing the internal functionality to consumer applications through the NEF to "External Application Function". In the same figure one can see that the 6G core is also exposing an interface to an external machine controller, i.e. TSN controller, to request reliable deterministic communications with other devices connected to the on-premises industrial local area network (LAN).

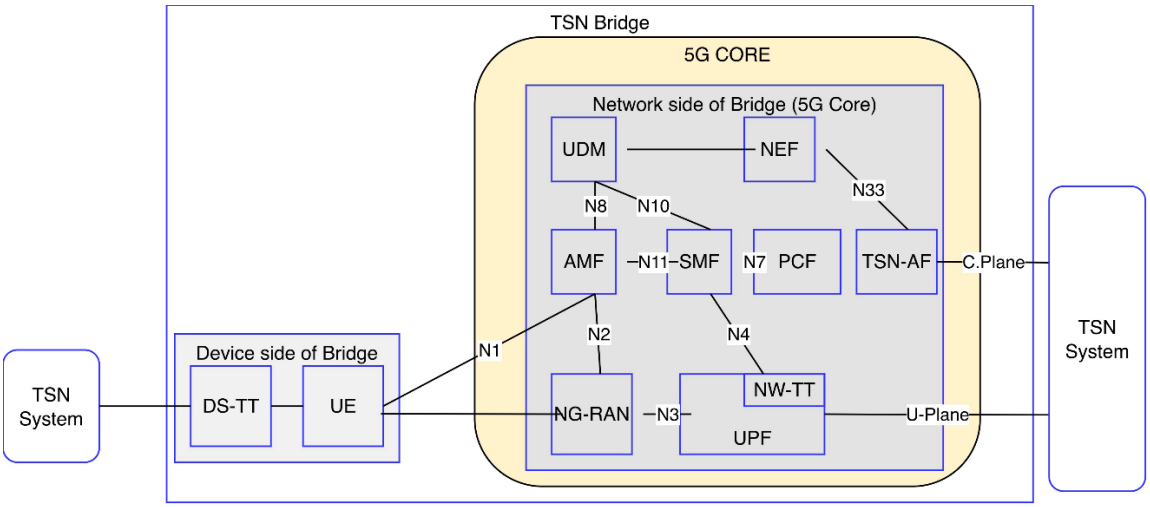

Figure 3: 6G core architecture with NF for industrial and service centric applications through NEF

Table 2 below shows a set of new NFs defined in 3GPP Rel 17 and Rel 18 in order to utilize the 6G core for service-oriented applications in private industrial networks where Time Sensitive Networking is required for deterministic communications.

Component	***Description***	***Interactions***
NW-TT	Network TSN Translator	It is the key function in the 5G core to support the integration with industrial LAN that support Time Sensitive Networking communications.
DS-TT	Device Side TSN Translator	Is the function required in the device side to support TSN communications of mobile devices

		connected to 5G Customer Premises Equipment (CPEs). The DS-TT must be connected with the core function NW-TT for end-to-end TSN support over the 5G system.
TSN-AF	TSN Application Function	This AF exposes the 5G system as native TSN bridge as specified by the 3GPP compliant with TSN specifications. The external TSN controllers will interact with the TSN-AF as if the 5G was native TSN bridge.
CNC	Centralized Network Controller	TSN controller that collects information from all the TSN devices in the network and configures them to setup TSN flows.
NSACF	Network Slicing Access Control Function	Network function that controls the access to the network slices based on number of authorized devices.
UPF-Eth PDU	User Plane Function with Ethernet PDU support	Standard UPF that includes support for Ethernet PDU. The UPF requires integration with low level network interfaces in order to access Hardware based time stamping required for TSN NW-TT but also to transparently transfer L2 packets from the fixed LAN to the connected UEs.

Table 2: Additional 6G Core network functions

A major effort was required to migrate the user plane function (i.e., the UPF) from fully Hardware integrated to support time sensitive features into virtualization platform such Kubernetes. The latest changes of Linux OS resulted in major impact on the performance of the UPF which was designed for high optimization and integration with Network Interface Card (NIC) driver. An Extended Berkeley Packet Filter (eBPF)-based IP handling software is failing during kernel upgrades because the Linux kernel does not guarantee a stable internal API/ABI for internal structures and verifier logic. Over the past five years (2021-2026), the Linux networking stack and eBPF subsystem have evolved rapidly, introducing breaking changes in memory layouts, verification strictness, and packet-processing pipelines. This has resulted in problems when migrating the UPF from its hardware-optimized implementation to Kubernetes-based virtualization. The core data structure for network packets, *struct sk_buff (skb)*, has been modified in almost every major kernel version. This leads to significant performance drops since the UPF code was accessing packet offsets, structural padding changes or the rearrangement of fields (like mark, priority, or internal flags) that was causing Cumucore's UPF based eBPF program to read corrupted data.

Category	***Status***
Version	1.0
Open API Swagger	https://gitlab.com/safe-6g/development/cmc-nef/-/blob/main/cmc-nbi-oam-openapi.yaml?ref_type=heads
Deployment requirements	Heavily depending on the scale of the system. As guiding figures: • Signalling: 32 GB RAM, 1TB storage, 16 vCPU • User pane: 8 GB RAB, 250GB storage, 4 vCPU, NIC i226 E801 • Dedicated port per UE for Ethernet PDU
Dependencies	Linux Kernel 6.14 with eBPF XDP maps

	Advanced Vector Extensions (AVX) CPU Support for MongoDB NIC with XDP Hardware time stamping for TSN support e.g. NIC i226, E801
License	Cumucore commercial license

Table 3: 6G core final release summary

2.1.1 6G Core Extensions

The 6G core is being extended to include full end to end control of the 6G system including RAN, Transport and Core. The core adds the new controllers following the SBA paradigm with new network functions, moving towards more a versatile solution where the different resources are managed through dedicated controllers to deliver an end-to-end solution based on the user or service requirements. The controllers will be defined by different vendors based on whether they manage the radio resources, i.e., RAN Network Slice Subnet Management System (RAN-NSSMS), transport i.e., Transport Network Slice Subnet Management System (TP-NSSMS) or Core Network Slice Subnet Management System (CN-NSSMS), as shown in the next Figure.

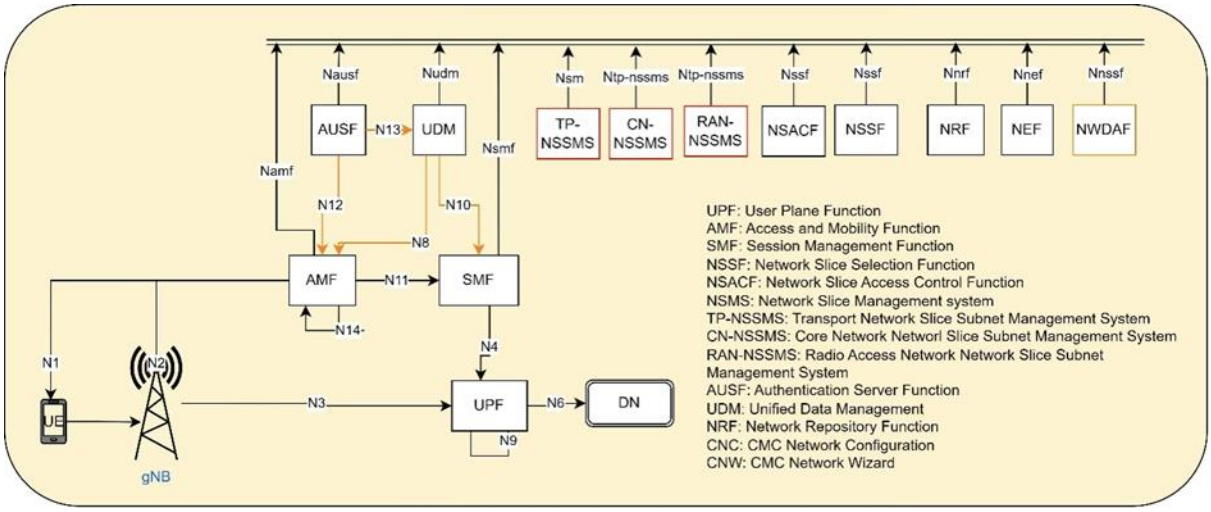

Figure 4: 6G core network controller functions

The controller functions can be integrated in the 6G core network for managing the resources and allocating them to different network slices. Thus, the controllers would be part of the network slice management. Figure 5 shows how the integration of new controller functions would be associated with the network slice assignment. It shows the user to request a dedicated slice from the Graphical User Interface (GUI) through external application represented as Operation and Management (O&M) module which could be exposed following the NEF through Open CAPIF.

After the user has created the slice with specific service requirements the core functions will first check the device has access through the NSACF and after that the Network Slice Management Function (NSMF) will interact with RAN, Core and Transport controllers to allocate the necessary resources. Those controllers might be vendor specific and support different technologies such as Network Configuration Protocol (NETCONF), which is based on standard protocols. Other vendors might provide different controllers that should follow the SBA to register as a core network function and provide the control functionality to the NSMF for managing the resources for creating the network slices.

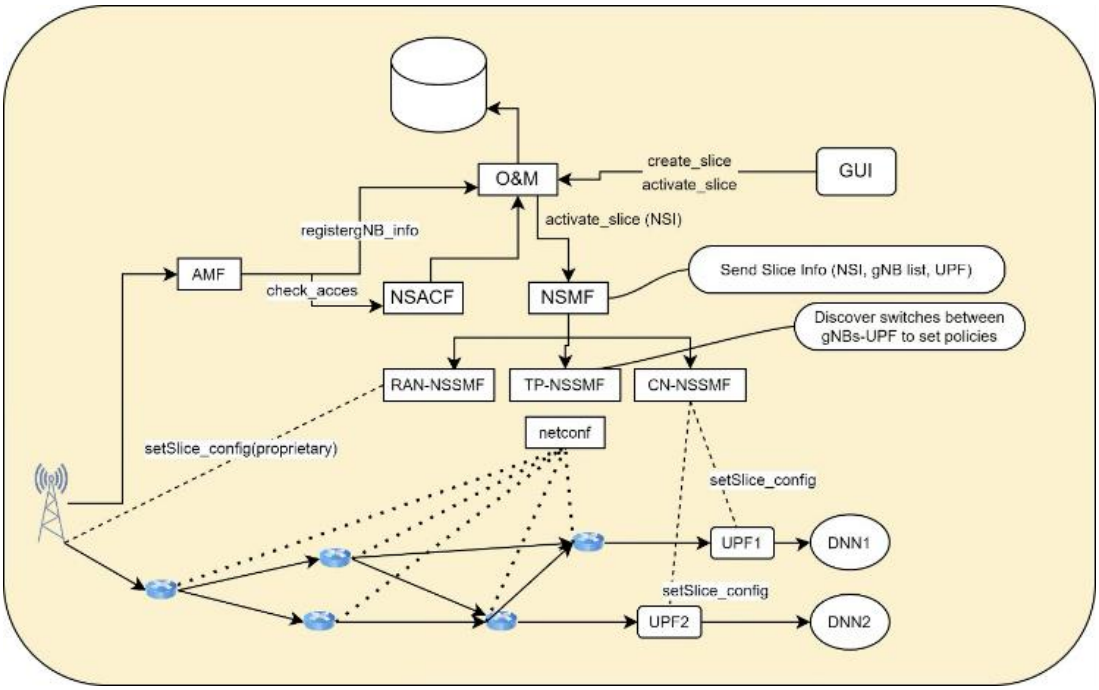

Figure 5: Network slice management with 6G core controller functions.

3 6G OPEN-SOURCE CORE

In addition to the commercial 6G core platform described in the previous section, SAFE-6G also considers an open-source mobile core implementation to evaluate the proposed framework across heterogeneous deployment environments. Assessing the SAFE-6G architecture on both commercial and open-source platforms demonstrates that the proposed trust mechanisms, AI-driven functions and network exposure capabilities are not tied to a specific vendor implementation but can be integrated into standards-compliant 5G/6G core networks. This approach also enables rapid prototyping, experimentation and validation of new functionalities within an open and extensible software ecosystem.

For this purpose, SAFE-6G utilizes Open5GS[1], a production-grade open-source implementation of the 3GPP Evolved Packet Core (EPC) for LTE networks and the 5G Core (5GC) for Standalone (SA) deployments. Open5GS implements the 3GPP SBA, where core NFs operate as independent software services communicating through standardized service-based interfaces and RESTful APIs. The platform provides the complete set of fundamental network functions required for 5G operation, including the AMF, SMF, UPF, NRF, AUSF, UDM and PCF, offering a standards-compliant baseline suitable for research and advanced service development.

During network operation, the AMF manages UE registration, mobility procedures and NAS signalling, while the SMF establishes and controls PDU sessions and configures forwarding rules within the UPF through the Packet Forwarding Control Protocol (PFCP). The UPF provides high-performance user-plane forwarding between the Radio Access Network (RAN) and external data networks using the GPRS Tunnelling Protocol User Plane (GTP-U). Internally, Open5GS employs an event-driven and asynchronous processing architecture that efficiently handles signalling procedures while maintaining compliance with 3GPP specifications for authentication, authorization, mobility management, policy control, security and service discovery.

The selection of Open5GS within SAFE-6G is motivated by several characteristics that closely align with the objectives of future 6G systems:

- **3GPP Service-Based Architecture compliance**, ensuring interoperability with standardized 5G network functions and facilitating the integration of future 6G capabilities.
- **Extensible and modular architecture**, enabling the implementation of additional network functions and project-specific extensions without modifying the overall system architecture.
- **Cloud-native deployment support**, allowing deployment of containerized environments through platforms such as Docker and Kubernetes and facilitating distributed edge-cloud execution.
- **Support for distributed user-plane deployments**, enabling localized UPFs and low-latency service provisioning, consistent with edge-cloud continuum architectures envisioned for 6G.
- **Open implementation**, providing an ideal platform for research, rapid prototyping and validation of novel AI-driven networking and trustworthiness mechanisms.

[1] Open5GS documentation: https://open5gs.org/

Within SAFE-6G, the baseline Open5GS platform has been extended to support the project requirements and to enable seamless integration with the SAFE-6G framework. The implemented enhancements (features) include:

- **Comprehensive operational monitoring**, through centralized logging, Prometheus-based metrics collection and Grafana dashboards, providing continuous visibility of network operation and facilitating performance analysis and troubleshooting.
- **Controlled network exposure**, achieved through the integration of a custom NEF, allowing fine-grained control over the capabilities and network information exposed to external applications while introducing an additional layer of security.
- **Support for 3GPP-compliant northbound APIs**, enabling the implementation and exposure of project-specific APIs through the NEF for interaction with external applications and orchestration components.
- **Integration with SAFE-6G intelligent components**, allowing interoperability with AI-driven analytics, trust functions and orchestration services while preserving compliance with the SBA.

The baseline Open5GS deployment was extended within SAFE-6G to support the project's requirements for secure network exposure, AI-driven orchestration and interoperability with external applications. In addition to integrating monitoring and analytics capabilities, the implementation introduces a set of 3GPP-compliant northbound APIs exposed through the NEF, enabling controlled access to selected network services and information. The following subsections describe the NEF APIs developed in SAFE-6G, their functionality, and their role in enabling secure interaction between the 5G/6G core and external AFs.

3.1.1 NEF Monitoring Event API

The NEF Monitoring Event API provides external authorized applications with controlled access to network monitoring information exposed by the 5G/6G core through the NEF. Its objective is to allow external AFs and SAFE-6G trust components to subscribe to, retrieve and monitor network events without requiring direct access to internal network functions, thereby preserving the SBA principles and the security boundaries of the core network.

Within SAFE-6G, this API acts as the primary interface for exposing operational information required by AI-driven analytics, trustworthiness assessment and network-aware applications. The implementation follows the 3GPP NEF exposure model, where monitoring events generated by internal network functions are securely made available through standardized northbound REST interfaces. In particular, the NEF Monitoring Event API enables the exposure of location area information for a specific User Equipment (UE) through the 5G Core Network, allowing authorized AFs to receive monitoring event reports related to the UE's location. This capability supports network-aware services and context-aware applications while ensuring controlled access, authentication and authorization through the NEF.

The NEF Monitoring Event API allows external applications to:

- **Subscribe** to monitoring events such as Location Reporting for UE’s **current location**.

- **Receive reports** (via callback URL) based on event occurrences or configured intervals.
- **Query the last known location** for a specific UE.

The implementation follows the **3GPP TS 29.122** specification for MonitoringEvent API and supports integration within the CAPIF security framework. NEF retrieves location information for a UE based on an AMF exposed API that extracts related information and writes it in a MongoDB. Thus, the NEF API fetches the related location data from the MongoDB. There are two distinct flows according to the main operation of that API. The first one is the flow when a CURRENT_LOCATION request has been performed for a certain UE (see Figure 6). The other is for a LAST_KNOWN_LOCATION Request (Figure 7).

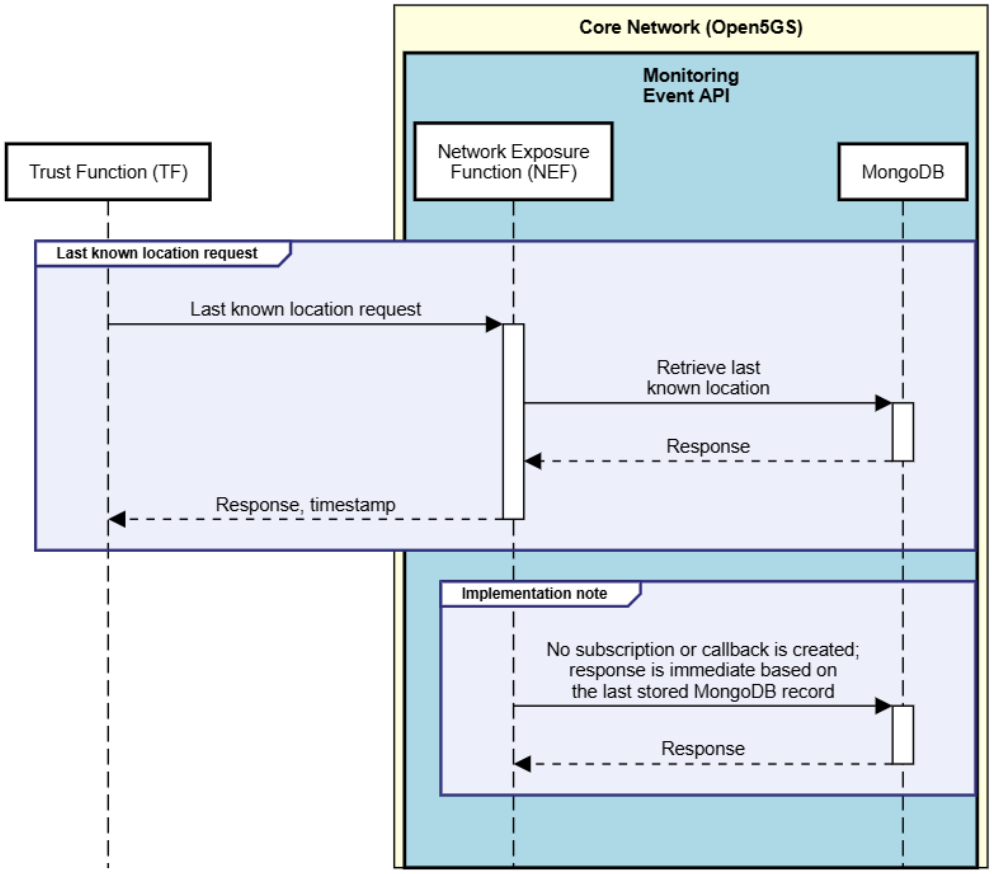

Figure 6: Current Location Request flow

The following steps are made in the sequence flow of Figure 6:

1. AF sends POST subscription request for current location.
2. NEF acknowledges subscription creation (201 Created).
3. Request for AMF's infoAPI is made and then location updates are stored in MongoDB.
4. NEF retrieves the latest UE location record from MongoDB.
5. NEF sends a MonitoringNotification to AF's callback URL with a MonitoringEventReport message encapsulated.
6. AF acknowledges that a Monitoring Notification is received.

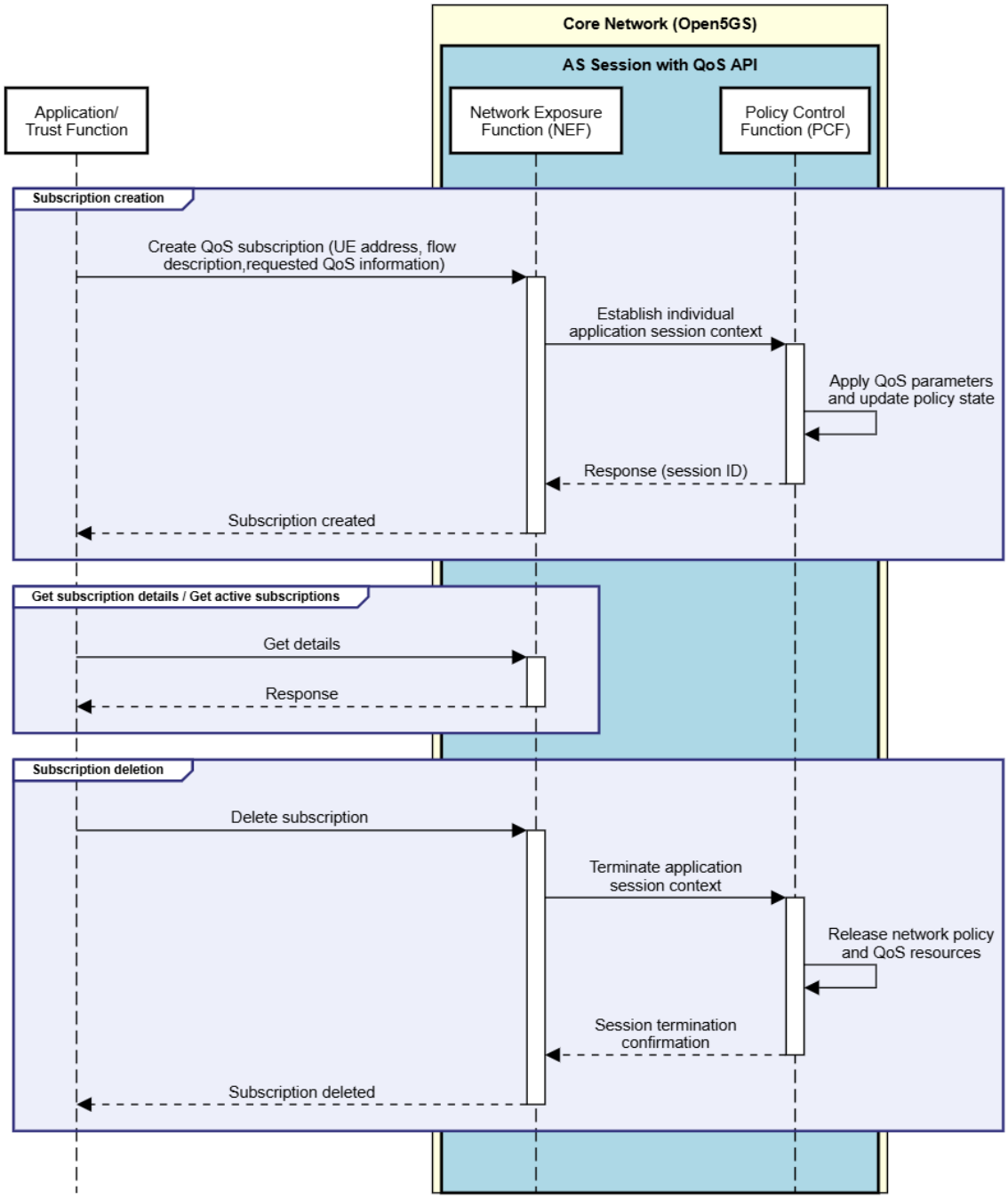

Figure 7: Last Known Location Request flow

The following steps are made in the sequence diagram of Figure 7:

1. AF requests last known location for a certain UE.
2. NEF fetches the last known location of the UE from MongoDB.
3. If any data entry is found in the database, the last known location returned to NEF.
4. NEF immediately returns a 200 OK with the MonitoringEventReport.

The URI of MonitoringEvent resources consist of the following format: “apiRoot/3gpp-monitoring-event/v1/”, where *apiRoot* corresponds to the API’s endpoint. In Table 4, an overview of the resources and methods of that API is provided.

Category	*Status*
Version	1.0.1
Documentation	Provides the complete installation, configuration, API usage instructions, and deployment guide for the NEF Monitoring Event API. https://gitlab.com/safe-6g/development/nef-monitoring-event/-/blob/main/README.md
Source code	Contains the full open-source implementation of the NEF Monitoring Event API, including the REST endpoints and supporting modules. https://gitlab.com/safe-6g/development/nef-monitoring-event
Docker images	Provides the pre-built Docker image required to deploy the NEF Monitoring Event API. https://hub.docker.com/r/safe6g/nef-monitoring-event
Helm charts	Not applicable, as no Kubernetes Helm chart is used for this component.
Open API Swagger	Provides the OpenAPI (Swagger) specification describing all available REST endpoints, request parameters, and response formats. https://gitlab.com/safe-6g/development/nef-monitoring-event/-/blob/main/monitoring_event_openapi.yaml
Demo	No public demonstration is currently available for this component. It will be provided during integration period for better clarity.
Deployment requirements	• 500 MB RAM • 1 vCPU
Dependencies	Open5GS core network
License	Apache 2.0

Table 4: NEF Monitoring Event API final release summary

3.1.2 NEF AsSessionWithQos API

The AsSessionWithQoS API enables an AF to request from the network to provide specific QoS to an application session for one UE or for a list of UEs, in accordance with 3GPP TS 29.122 and 3GPP TS 23.502. The AF typically specifies the required QoS level by indicating a 5QI (5G QoS Identifier) value or specific QoS parameters such as guaranteed and maximum bit rates. This RESTful API defines the data models, resources and procedures for creating and managing Application Server (AS) sessions with required QoS (3GPP TS 29.122, subclause 5.14.1). The session setup and policy enforcement are implemented via the Policy Control Function (PCF)'s Npcf_PolicyAuthorization standard call, with the NEF mediating the request to the Policy Function (PF). The AsSessionWithQoS API developed in the project complies with 3GPP NEF compliant (3gpp-as-session-with-qos) specification. It particularly exploits 3GPP TS 29.122 V17.2.0 T8 reference point for Northbound APIs[2], 3GPP TS 29.571 Common Data Types for Service Based Interfaces, version 17.2.0[3] and 3GPP TS 29.514 V17.1.0; 5G System; Policy Authorization Service; Stage 3[4]. Figure 8 illustrates the sequence diagram of the implemented API.

[2] https://www.3gpp.org/ftp/Specs/archive/29_series/29.122/
[3] http://www.3gpp.org/ftp/Specs/archive/29_series/29.571/
[4] http://www.3gpp.org/ftp/Specs/archive/29_series/29.514/

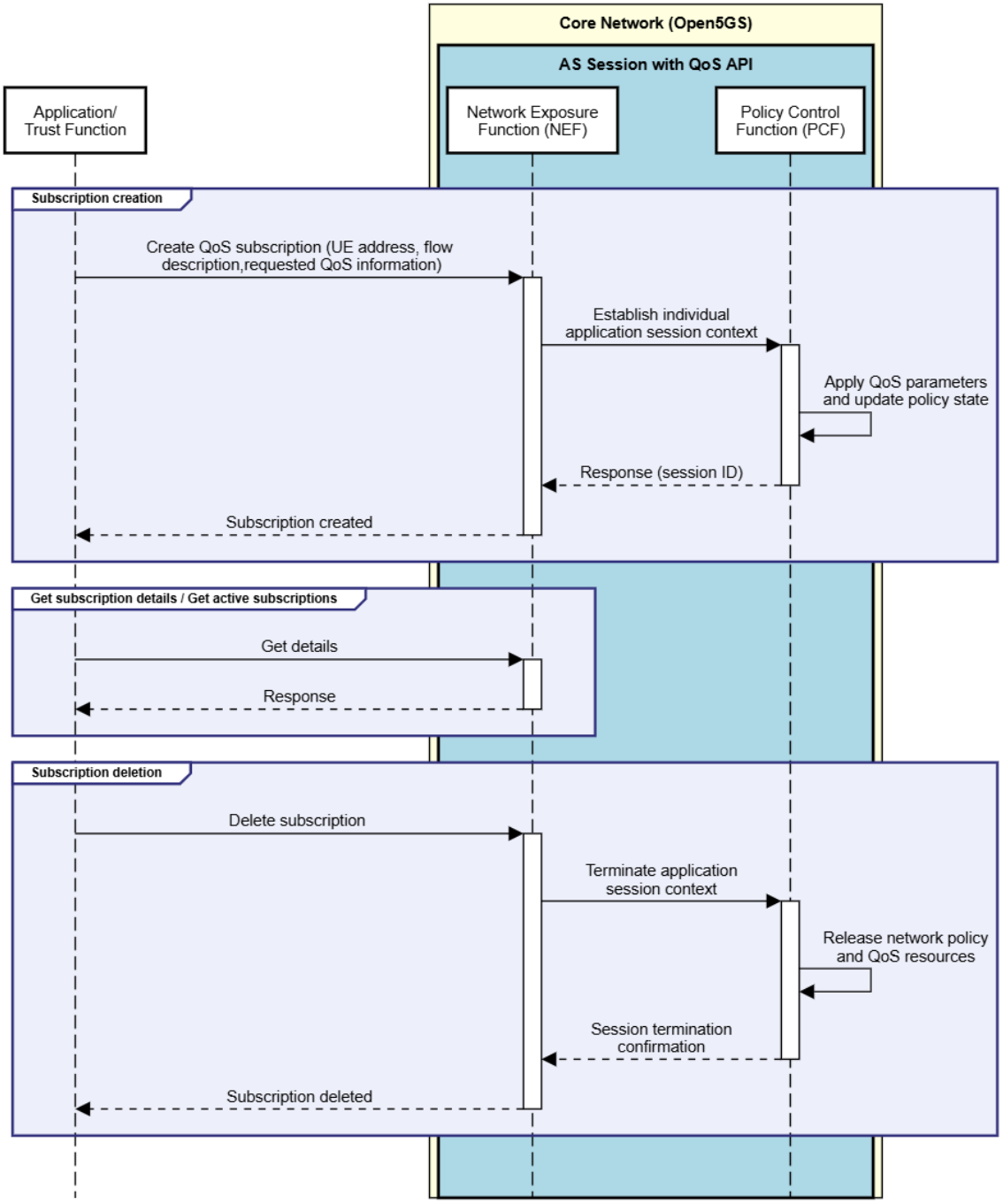

Figure 8: Sequence Diagram AsSessionWithQoS API

The following explanation corresponds to the sequence diagram shown in the figure above:

1. An external AF or service (e.g., a Trust Function) shall send a POST request to the NEF to create a QoS subscription. It contains a UE address or a list of UE addresses, plus flow description information specifying the associated Application Session and the requested QoS information, e.g., a 5G QoS Identifier (5QI) value.
2. NEF sends a policy authorization request to PCF and establishes an individual application session context. Normally, the NEF uses the UE address in step 1 to discover the PCF associated with the UE. Still, this is not required here since the network contains only one PCF.

- At this point, the PCF updates its policies and initiates a Session Management (SM) Policy Association Modification procedure as defined in TS 23.502 clause 4.16.5.2 to notify the SMF about the modification of policies. In turn, the SMF initiates the PDU Session Modification procedure as defined in TS 23.502 clause 4.3.3 to provide the requested QoS information and flow description to the RAN and UPF. This enforces the UPF to apply the requested QoS information for the associated session.

3. PCF returns the Session Context Response (session ID) to the NEF.
4. NEF replies to the AF with "201 Created", confirming the QoS subscription.
5. Optionally, the external AF may consume specific endpoints to retrieve either information related to a specific subscription or the entire list of all active ones.
6. If the AF wants to remove an existing individual subscription, it shall send a DELETE request to remove it. The NEF would then instruct the PCF to terminate the Application Session Context. The PCF returns session termination confirmation to the NEF, which would confirm the deletion to the AF.

Throughout this flow, the NEF mediates all AF interactions with the PCF, exposing simple create, read and delete operations for managing QoS application sessions. In Table 5, an overview of the resources and methods of that API is provided.

Category	***Status***
Version	Latest
Documentation	Provides the installation, configuration, deployment and API usage guide for the NEF AsSessionWithQoS API. https://gitlab.com/safe-6g/development/nef-qos/-/blob/main/README.md?ref_type=heads
Source code	Contains the complete open-source implementation of the NEF AsSessionWithQoS API and its supporting modules. https://gitlab.com/safe-6g/development/nef-qos/-/tree/main?ref_type=heads
Docker images	Provides the pre-built Docker image required to deploy the NEF AsSessionWithQoS API. https://hub.docker.com/r/safe6g/nef-qos
Helm charts	Not applicable, as no Kubernetes Helm chart is used for this component.
Open API Swagger	Provides the OpenAPI (Swagger) specification describing the available REST endpoints, request parameters and response messages. https://gitlab.com/safe-6g/development/nef-monitoring-event/-/blob/main/monitoring_event_openapi.yaml
Demo	No public demonstration is currently available for this component. It will be provided during the integration period for better clarity.
Deployment requirements	• 500MB RAM • 1 VCPU
Dependencies	Open5GS core network, Docker, Python
License	Apache 2.0

Table 5: NEF AsSessionWithQoS API final release summary

4 6G CORE INTERFACES

4.1 INTERNAL API

The 6G core allows to have internal API that is accessible for other network functions deployed together with the main 6G core network functions. The internal API has the basic TLS connections to secure the integration between NFs.

API prefix: /api/v1.0		
Endpoint	***Method/s***	***Description***
/cnc-subscriber-management?order={int=1 or -1}	GET	Subscribers profiles
/cnc-subscriber-management/by-group?gname={{USER_GROUP_ID}}&order={{1 or -1}}	GET	Subscribers info by group
/smf-cnc/qos-monitoring-report/	GET	QoS monitoring report from the UPF
/smf-cnc/qos-monitoring-report?report-type=delay&tgt_ue=\"{{TARGET_SUPI}}\"&time=10	GET	UE delays measurement
/cnc-configuration/sm-policy-config?ueId={{TARGET_ SUPI}}&snssai={sst:{{TARGET_SST}},sd:"{{TARGET_SD}} "&dnn="{{TARGET_DNN}}"}	GET	SM policy of UE (SUPI)
/cnc-nwdaf/analytic-info-request?event-id={{event type}}	GET	NF list with features at NRF
/nnwdaf-analyticsinfo/v1/analytics?event-id=LOAD_LEVEL_INFORMATION&event-filter={"snssais":"[{"sst":{{TARGET_SST}},"sd":"{{TARGET_SD}}"}]}	GET	Slice usage statistics
/cnc-slice-capacity/per-interface-stats/{{sst+sd}}	GET	Slice usage capacity
/cnc-tac-geolocatiom/Polygon	GET	Polygons and slices in TA
/cnc-config/af-qos-profile	GET	QoS profile templates

Table 6: CNC API

API prefix: /nwdaf-analyticsinfo/v1/analytics?eventid=		
Endpoint	***Method/s***	***Description***
LOAD_LEVEL_INFORMATION&eventfilter={"snssais":"[{"sst":{{TARGET_SST}},"sd":"{{TARGET_SD}}"}]}	GET	Analytics of network slice
UE_COMM&tgtue={"supi":["{{TARGET_SUPI}}"]}	GET	Analytics of UE (TS 23.288)
NF_LOAD&tgtue={"anyUe":true}	GET	NF load analytics predictions
UE_MOBILITY&tgt-ue={"supi":["{{TARGET_ SUPI}}]}	GET	UE mobility analytics (SUPI)
USER_DATA_CONGESTION&tgt-ue={{TARGET_ SUPI}}	GET	UE data congestion analytics/ predictions (SUPI)
NETWORK_PERFORMANCE&tgt-ue={"anyUe": true}	GET	GNB Analytics/predictions

Table 7: NWDAF API

API prefix: /3gpp-as-session-with-qos/v1/0001/subscriptions		
Endpoint	***Method/s***	***Description***
/	POST	Add secondary flow with QOS on top of default flow with best effort
/{{FLOW_ID}}	DELETE	Delete the secondary flow

Table 8: QoS session API

4.2 EXTERNAL API

Besides the internal API, the 6G Core also expose a subset of functionality to external AFs that can access through the NEF which also expose them as part of Open CAPIF.

API prefix: /		
Endpoint	***Method/s***	***Description***
/oam-data/operator-info	POST	Create operator
/oam-data/operator-info?id={id}	PUT	Update operator
/oam-data/operator-info	GET	Get operators
/oam-network-slice/slice-instance	POST	Create a slice meta data
/oam-configuration/ue-subscription-profile	GET	Get Slice
/oam-network-slice/slice-instance/{sliceName}	DELETE	Delete a slice
/oam-configuration/ue-subscription-profile	POST	Create qos profile
/oam-configuration/ue-subscription-profile/{profile_id}	PUT	Update qos profile
/oam-configuration/user-group	POST	Create Group
/oam-configuration/user-group/{groupId}	GET	Get Group Info
/oam-configuration/user-group/{groupId}	PUT	Update Group Info
/oam-network-slice/upf-dnn-list	GET	List available UPFs
/oam-config/upf-profile-update/{nfInstanceId}	PUT	Update UPF configuration
/oam-subscriber	POST	Create subscriber
/oam-subscriber	PUT	Update subscriber
/oam-preemption/preemption-config	POST	Update Qos on the run
/oam-settings/single-ue-am-status/{gpsi}	GET	Get gNB IP
/oam-settings/connectivity-per-gnb?gnb-ip={ip_address}	GET	Get gNB load
/oam-monitoring-info/core-capacity/interface-stats?id=goupf	GET	Get user plane load in Bare metal deployment
/oam-settings/core-capacity	GET	Get signalling load
/oam-data/{nfType}/nf-analytics-request	GET	Get computing platform load
/oam-data/v1/oam-monitoring-info/core-capacity/k8-interface-stats?instanceName={UPF}&namespace={5g}	GET	Get Computing platform load
/3gpp-analyticsexposure/v1/{afId}/fetch	POST	Get UE Usage
/oam-preemption/preemption-config	GET	Subscribe to alarm events
/oam-alarms/v1/alarm-event/history-information/time-range?start-time={start-time}&end-time={end-time}	GET	Get history of alarms
/api/v1/alarm-event/history-information?eventId=latest	GET	Get the latest specific alarm item

Table 9: NEF API exposed to external AFs

5 CONCLUSION

This document reported the outcomes of T3.2, mostly related to the final release of the packet core software including the following work items:

- Migrating the user plane to the cloud addressing the performance limitations due to the limitation of virtualisation platforms.
- Extending NEF with additional API to support SAFE-6G functions to access the functionality of the core.
- Extending NWDAF to provide performance and Key Performance Indicators through the NEF for external SAFE-6G functions implement the logic required to improve system trustworthiness.
- The SAFE-6G framework has been realized through both a commercial (Cumucore) and an open-source (Open5GS) implementation, demonstrating that the proposed architecture, interfaces, and network exposure mechanisms can be integrated across heterogeneous 5G/6G core platforms, improving interoperability, portability, and future adoption.

This document includes the evolution from state-of-the-art 5G core towards 6G core with AI-enabled functionality and fully distributed controller-based architecture. The NEF implementation and APIs are also documented.