

SAFE-6G

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

Deliverable D4.3: User-centric Safety Trust Function

Date: 30/09/2026

Version: V1.0

DISCLAIMER

This document contains confidential information reserved for the partners of the SAFE-6G (“A Smart and Adaptive Framework for Enhancing Trust in 6G Networks”) Consortium and is subject to the confidentiality obligations set out in the Grant Agreement 101139031 and to the SAFE-6G Consortium Agreement.

This document contains information, which is proprietary to the SAFE-6G Consortium that is subject to the rights and obligations and the terms and conditions applicable to the Grant Agreement. The action of the SAFE-6G Consortium is funded by the European Commission.

Neither this document nor the information contained herein shall be used, copied, duplicated, reproduced, modified, or communicated by any means to any third party, in whole or in parts, except with prior written consent of the SAFE-6G Consortium. In such case, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced. In the event of infringement, the consortium reserves the right to take any legal action it deems appropriate.

This document reflects only the authors’ view and does not necessarily reflect the view of the European Commission. Neither the SAFE-6G Consortium as a whole, nor a certain party of the SAFE-6G Consortium warrant that the information contained in this document is suitable for use, nor that the use of the information is accurate or free from risk and accepts no liability for loss or damage suffered by any person using this information.

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Grant Agreement	101139031
Document number	D4.3
Document title	User-centric Safety Trust Function
Lead Beneficiary	eBOS
Editor(s)	Theocharis Saoulides - eBOS
Author(s)	Theocharis Saoulides - eBOS Andreas Sousanis - eBOS
Dissemination level	Public
Contractual date of delivery	30/09/2026
Status	Final
File name	SAFE-6G_D4.3_V1.0.docx

Revision History

Version	
V0.1	Initial ToC
V0.2	First draft
V0.3	Consolidated draft
V0.4	Contributions with feedback from WP4 partners
V0.6	Consolidated contributions added
V0.7	Homogenization of content and final draft
V0.8	Internal review-ready version (revision made by TID and CMC)
V1.0	Final version

GLOSSARY

Abbreviations/Acronym	Description
3GPP	3rd Generation Partnership Project
5G	Fifth Generation mobile network
5G-AKA	5G Authentication and Key Agreement
5GS	5G System
6G	Sixth Generation mobile network
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
API	Application Programming Interface
AUSF	Authentication Server Function
CAPIF	Common API Framework
CI/CD	Continuous Integration / Continuous Deployment
cLoTw	Calibrated Level of Trustworthiness
CoCo	Cognitive Coordinator
CP	Control Plane
CPE	Customer Premises Equipment
CPU	Central Processing Unit
DataOps	Data Operations
DDoS	Distributed Denial of Service
DN	Data Network
DNN	Data Network Name
DoS	Denial of Service
DTaaS	Digital Twin as a Service
eSMF	Enhanced Session Management Function
ETSI	European Telecommunications Standards Institute
EU	European Union
eUPF	Enhanced User Plane Function
gNB	next Generation NodeB
GTP-U	GPRS Tunnelling Protocol User Plane
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IKEv2	Internet Key Exchange version 2
IP	Internet Protocol
IPsec	Internet Protocol Security
KPIs	Key Performance Indicators
MEC	Multi-access Edge Computing
ML	Machine Learning
ML-KEM	Module-Lattice-Based Key-Encapsulation Mechanism
MLOps	Machine Learning Operations
mTLS	Mutual Transport Layer Security
nApp	Network Application
NAS	Non-Access Stratum
NAT	Network Address Translation
NEF	Network Exposure Function
NF	Network Function
NFV	Network Functions Virtualisation
NGSI-LD	Next Generation Service Interfaces - Linked Data

NIST	National Institute of Standards and Technology
nLoTw	Non-calibrated Level of Trustworthiness
NRF	Network Repository Function
NSN	Network Service Node
NWDAF	Network Data Analytics Function
Open5GS	Open 5G System
OpenCAPIF	Open Common API Framework
OpenVPN	Open Virtual Private Network
PCC	Policy and Charging Control
PCF	Policy Control Function
PDU	Protocol Data Unit
PFCP	Packet Forwarding Control Protocol
PLMN	Public Land Mobile Network
QoS	Quality of Service
RAN	Radio Access Network
REST	Representational State Transfer
SBA	Service-Based Architecture
SBI	Service-Based Interface
SCP	Service Communication Proxy
SDN	Software Defined Networking
SDP	Software Defined Perimeter
SDPCF	Software Defined Perimeter Controller Function
SDPGW	Software Defined Perimeter Gateway
SIM	Subscriber Identity Module
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
SPA	Single Packet Authentication
TLS	Transport Layer Security
ToC	Table of Contents
TOSCA	Topology and Orchestration Specification for Cloud Applications
TS	Technical Specification
TUN	TUNnel Network Interface
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UP	User Plane
UPF	User Plane Function
URSP	UE Route Selection Policy
USN	User Service Node
vApp	Vertical Application
vCPU	virtual Central Processing Unit
VGW	VPN Gateway Function
VPN	Virtual Private Network
XR	Extended Reality

EXECUTIVE SUMMARY

This deliverable presents the User-centric Safety Function of SAFE-6G, detailing its design, architectural realisation, and implementation as a dynamic, adaptive, and trust-aware mechanism for strengthening safety in future 6G networks. The function serves as a key enabler for translating user-centric trust requirements into concrete network-level enforcement actions, principally through the integration of Zero-Trust principles and Software Defined Perimeter (SDP) technology. In doing so, SAFE-6G advances beyond conventional perimeter-based security by adopting an authenticate-then-connect paradigm, whereby protected services remain concealed from unauthorised entities until authentication, authorisation, and trust evaluation have been successfully completed.

The Safety Function is structured around three complementary layers, in alignment with the common SAFE-6G Trust Function model. The Network Application (nApp) layer gathers the available resources and the current topology of the edge-cloud continuum, providing the network context on which safety decisions depend. The Local Artificial Intelligence (AI) Agent combines this context with the trust input received from the SAFE-6G trust management framework to select one of three safety flavours, Low, Medium, or High, with the enforced decision derived from the calibrated level of trustworthiness computed across all Trust Functions. The vApp layer then realises this decision, deploying the SDP Controller Function (SDPCF), the SDP Gateway (SDPGW), and the Virtual Private Network (VPN) Gateway Function (VGF) in the selected domain of the continuum and provisioning the corresponding policy rules in the 5G core. Through this layered design, the function delivers individualised security perimeters, dynamic micro-segmentation, and encrypted Protocol Data Unit (PDU) session connectivity tailored to the user, the requested service, and the prevailing operational context.

The deliverable also details the implementation of the function. The SDPCF operates as an enhanced Session Management Function (SMF), interoperating with an Open5GS-based 5G core, whilst the SDPGW operates as an enhanced User Plane Function (UPF) that restricts protected traffic exclusively towards the VGF. The VGF provides WireGuard-based encrypted tunnels between the User Equipment (UE) and the domain hosting the protected service. All components are containerised, built through a GitLab CI/CD pipeline, and deployed on demand across the continuum through the aerOS platform using TOSCA service templates.

This design reflects a trade-off inherent to SDP-based approaches: whilst enhanced concealment and isolation improve resilience against port scanning, denial-of-service (DoS) exposure, and unauthorised discovery, they introduce additional setup time associated with trust calibration, component deployment, and session establishment. Within SAFE-6G, this overhead is managed as part of a calibrated enforcement process, in which encrypted PDU-session connectivity is established only when the trust context and available infrastructure capacity justify it. The Safety Function therefore constitutes not merely a security component, but an adaptive trustworthiness mechanism that aligns user intent, network telemetry, resource availability, and encrypted PDU-session enforcement across the edge-cloud continuum. The deliverable concludes with the research questions and integration aspects to be carried forward into WP5 pilot integration and validation.

KEYWORDS

Safety Function, Zero Trust, Software Defined Perimeter (SDP), Micro-segmentation, Trustworthiness, Local AI Agent, Encrypted PDU Session, 5G Core, Edge-Cloud Continuum

TABLE OF CONTENTS

1	<i>Introduction</i>	<i>1</i>
2	<i>Related Work</i>	<i>4</i>
2.1	Safety Function Foundations and Relevance to SAFE-6G	5
2.2	Integration of SDP with Programmable Networks.....	6
3	<i>Trust Function Definition and Role in SAFE-6G</i>	<i>8</i>
3.1	Requirements Mapping	9
4	<i>Reference Architecture</i>	<i>13</i>
4.1	Component Interaction and Message Flow.....	14
4.2	Deployment Assumptions and Design Constraints	17
5	<i>Trust Function Layers.....</i>	<i>19</i>
5.1	Local AI Agent.....	19
5.1.1	Agent Inputs, Data Consumption and Output	19
5.2	Vertical Applications (vApps)	20
5.2.1	Implementation of SDP Controller Function (SDPCF)	22
5.2.2	Implementation of SDP Gateway (SDPGW)	23
5.2.3	Implementation of VPN Gateway Function (VGF)	24
5.2.4	Interfaces and Integration Points.....	26
5.2.4.1	5G Core Integration	26
5.2.4.2	User-Plane and Tunnel Integration.....	27
5.2.4.3	Intelligence and Orchestration Integration	28
5.3	Network Applications (nApps)	29
5.3.1	Core Utility Modules	29
6	<i>Trust Evaluation & Enforcement Logic</i>	<i>33</i>
6.1	Encrypted PDU-Session Enforcement Mapping	33
6.2	Attack Handling and Resilience Logic.....	35
7	<i>Future Research Pathways</i>	<i>37</i>
7.1	Architectural Integration and Operation.....	37
7.1.1	Coordination with 3GPP Procedures.....	37
7.1.2	Multi-Function Coordination and Decision Precedence	38
7.1.3	Scalability and Orchestration	38
7.1.4	Mobility and Perimeter Relocation	40
7.2	Trustworthiness and Security Resilience.....	40

7.2.1	Data Availability and AI Reliability	40
7.2.2	Explainability and Regulatory Alignment	41
7.2.3	Protection of Control and Enforcement Points.....	41
7.2.4	Cryptographic Agility and Post-Quantum Readiness	41
8	Conclusions	43
9	References	45

List of FIGURES

Figure 1: Safety Trust Function – High Level Architecture view	13
Figure 2: Safety Trust Function – Sequence Diagram	17
Figure 3: Safety Trust Function – Deployment view	18
Figure 4: Safety Trust Function – Orchestration flows across vApps and SDP components	22
Figure 5: Safety Trust Function – Local AI Agent interactions with the system components	31

List of TABLES

Table 1: Safety Trust Function Requirements.....	10
Table 2: Core Network Interfaces & Protocols	27
Table 3: User Plane Interfaces & Protocols.....	28
Table 4: aerOS APIs	30
Table 5: Open5GS APIs.....	31
Table 6: Logical VPN-like domains and corresponding safety flavours	35
Table 7: Safety Perimeter Granularity levels	39
Table 8: Safety Function Setup Time Optimization view	39

1 INTRODUCTION

The evolution towards sixth generation (6G) networks introduces unprecedented levels of service flexibility, network disaggregation, edge-cloud integration, and user-centric service provisioning. These characteristics are expected to underpin highly dynamic communication environments, in which services are instantiated, adapted, and relocated across heterogeneous infrastructures in accordance with user requirements, application demands, and prevailing network conditions. At the same time, this evolution substantially expands the network attack surface, as network functions, user-plane resources, control-plane interfaces, edge nodes, cloud domains, and vertical applications (vApps) become progressively more distributed and programmable. Consequently, the task of preserving trustworthiness, security, privacy, and resource isolation becomes considerably more complex than in preceding network generations, where the infrastructure perimeter was more clearly delineated and security decisions remained largely operator-centric.

Traditional security approaches, predicated largely upon static perimeter defences, fixed firewall rules, and connect-then-authenticate access models, are proving increasingly inadequate for the dynamic and distributed nature of future 6G ecosystems. Experience gained from contemporary cloud-based, Software Defined Networking (SDN)-enabled, and Network Functions Virtualisation (NFV)-based networks demonstrates that virtualised, programmable infrastructures introduce a range of novel security and privacy challenges, including weak authentication and trust models, insufficient access control, compromised data integrity, and heightened exposure to network intrusion, distributed denial-of-service (DDoS), and port-scanning attacks. These findings are directly pertinent to 6G, where disaggregated network functions, edge-cloud resources, multi-stakeholder service provisioning, and user-centric orchestration necessitate security mechanisms that are adaptive, dynamic, and context-aware, rather than static and perimeter-bound [1]–[4].

Within the SAFE-6G framework, the Safety Function is introduced as a key trustworthiness enabler, tasked with enforcing user-centric safety and resource-isolation policies. Its principal objective is to translate the user trust requirements received from the Cognitive Coordinator (CoCo) [5] into concrete network-level enforcement actions, through the dynamic establishment of individualised operational boundaries. By applying Zero-Trust and SDP principles, the function renders network resources invisible to unauthorised entities until authentication and authorisation have been successfully completed. This "dark infrastructure" paradigm substantially reduces the exposure of critical network assets, precluding unauthorised entities from discovering services, ports, and network paths before access has been explicitly granted, an effect corroborated by SDP and SDN-SDP evaluations as well as by formal SDP specification guidance [1]–[3].

The SDP paradigm aligns closely with the SAFE-6G vision, insofar as it operates on a need-to-know model, verifying the identity of a User Equipment (UE) or application prior to granting visibility and connectivity. Importantly, the Safety Function does not supersede the standard 3GPP access procedure: the UE continues to attach to the mobile network via the Access and Mobility Management Function (AMF) through the conventional 3GPP registration, authentication, mobility, and session-management procedures. The Safety Function thus constitutes an additional trust-enforcement layer,

applying SDP principles once the UE has entered the 3GPP system, such that protected services are exposed only after the relevant trust and policy verifications have been satisfied. Within SAFE-6G, this is realised through the interaction between the SDPCF, the SDPGW, and the VGF, which collectively establish individualised micro-perimeters and encrypted PDU-session connectivity for authorised users and services. By way of illustration, consider a user accessing an immersive industrial-training application hosted at the edge. Under a conventional model, the service endpoint may be visible to the device before the trust decision has been finalised, thereby permitting an unauthorised device to probe ports or make repeated connection attempts. Under the Safety Function, by contrast, the service remains concealed until the UE identity and the requested service have been verified and the calibrated trust context permits access. Only then is the protected path established: the SDPGW restricts the session to a narrow access path towards the VGF, which establishes the corresponding secure tunnel. The service consequently remains hidden from the public network, whilst nonetheless remaining reachable by an authorised user through a controlled and temporary perimeter [3], [4].

In practical terms, the adoption of SDP within SAFE-6G enables the Safety Function to bridge abstract trust requirements and concrete network enforcement. It assesses whether access should be granted, determines the extent to which a service should be exposed, and selects the isolation, monitoring, and tunnelling appropriate to the prevailing context, anchored within the core Service-Based Architecture (SBA) and extending across the edge-cloud continuum. Such adaptability is essential for 6G systems, in which trustworthiness cannot be treated as a binary condition: a user, device, application, or slice may be sufficiently trusted for one service, whilst simultaneously requiring stronger isolation for another subject to higher safety or availability requirements. The Safety Function consequently calibrates protection to user-centric and service-centric needs, whilst remaining aligned with the operational state of the underlying infrastructure [4]–[6], [8].

This design matters most in scenarios where services are instantiated dynamically, Network Functions (NFs) are decomposed into cloud-native components, and user equipment interacts with multiple service domains within a single session. In such cases, the network must enforce trust, alongside the other Trust Functions of the SAFE-6G framework, not only at connection setup but throughout the full service lifecycle. This spans the initial interpretation of user intent, the deployment of the relevant SDP and VPN functions, the establishment of the secure path, and the revocation or adaptation of access as the trust context changes [9]–[11].

This deliverable therefore presents the design, architectural realisation, and implementation of the Safety Function within the SAFE-6G ecosystem. It first explains in further detail why conventional connect-then-authenticate mechanisms fall short in dynamic, distributed 6G environments, motivating the shift towards an authenticate-then-connect model. It then defines the role of the Safety Function within the SAFE-6G trust management framework, including its interaction with CoCo, its mapping to the project requirements, and its use of telemetry from the core network and the edge-cloud continuum. The architectural description follows the common SAFE-6G Trust Function structure: the nApp layer gathers the topology and resource information on which decisions depend, the Local AI Agent selects the safety flavour on the basis of the calibrated trust context, and the vApp layer deploys and configures the SDPCF, SDPGW, and VGF across the continuum. The deliverable then

details the implementation of these components, including their technologies and interfaces. Finally, it discusses the open research questions and integration aspects to be carried forward into WP5, relating to scalability, performance trade-offs, AI-driven trust assessment, and security resilience, before concluding with a summary of the function's role in enabling secure, user-centric 6G services.

The deliverable also clarifies how the Safety Function complements the broader SAFE-6G trustworthiness framework. CoCo interprets user intent and produces trust-related decisions at a higher semantic level, whilst the Safety Function focuses on the feasibility and enforcement of those decisions at the network level. This distinction matters because a trust request can be semantically valid yet technically infeasible if the available infrastructure cannot support the required isolation, tunnel capacity, or gateway placement. For this reason, the Safety Function participates in a two-stage calibration process: it receives a non-calibrated level of trustworthiness (nLoTw) from CoCo, reports the safety flavour it can feasibly enforce, and bases its final enforcement on the calibrated level of trustworthiness (cLoTw) that CoCo computes across all Trust Functions. By introducing the notion of Trust Function Capacity into this process, the Safety Function keeps user-centric trust requirements aligned with the actual state of the infrastructure [5], [8].

To maintain a focused design, the scope of this deliverable is defined as follows:

- In scope: The architectural design and integration of the SDP stack into the 6G Packet Core, specifically the definition of functionalities across the three operational layers, the Local AI Agent (decision-making), the vApps (deployment and orchestration), and the nApps (network context acquisition), together with the interaction between the core safety components, the SDPCF, SDPGW, and VGF; the mapping of the function to the project requirements; and the implementation of these components, including their technologies, interfaces, CI/CD pipeline, deployment through aerOS, and code repositories.
- Out of scope: The specification of cryptographic primitives, which are provided by the underlying tunnelling technology; low-level protocol specifications beyond the 3GPP-defined interfaces employed; production-grade hardening; and pilot integration and complete validation results.

2 RELATED WORK

Traditional network security architectures are primarily operator-centric, relying on static identities, perimeter-based mechanisms, and pre-configured security zones. These "castle-and-moat" approaches were designed for environments in which the private network boundary could be clearly separated from the public network, and in which internal users, applications, and servers were largely confined to a controlled domain. This model is increasingly inadequate for 5G and 6G environments, where services and resources are distributed across a heterogeneous edge-cloud continuum, users connect from multiple locations and devices, and the network itself is programmable, virtualised, and dynamically orchestrated. A single static perimeter is no longer sufficient, because protected assets may move, scale, or be instantiated on demand across multiple administrative and technological domains. For instance, in a traditional enterprise network, once a user connects through a VPN, that user may obtain visibility of a relatively broad internal address space, even if only one application is required. In a 6G edge-cloud environment, this approach becomes risky, since services may be distributed across several edge nodes, slices, and domains; a compromised device could exploit this broad visibility to scan internal services, identify exposed interfaces, or attempt lateral movement. SAFE-6G avoids this exposure by revealing only the specific service tied to an approved request, keeping the remaining infrastructure dark [1], [3], [4].

The SDP literature highlights that modern networks face combined safety, security, privacy, and availability concerns, including authentication, access control, data integrity, and data availability. These concerns are amplified in SDN-based networks, where the separation between control and data planes introduces additional exposure points. SDN controllers, for example, can become attractive targets because of their centralised role, whilst flow-table limitations and programmable interfaces may increase susceptibility to DDoS attacks, malicious flow manipulation, or unauthorised use of network resources. Existing solutions such as access control lists, static firewalls, honeypots, machine-learning-based traffic classification, and TLS-secured controller-switch channels offer partial protection, but often add operational complexity, processing overhead, and scalability limitations in virtualised, dynamic environments [1], [2].

The current 5G system also presents limitations from a Zero-Trust perspective. Once a UE has registered and established a PDU session, the services reachable through the corresponding data network are accessible on the basis of network-level connectivity rather than a per-service trust evaluation [4], [7]. Such services may therefore remain discoverable to any attached device, creating opportunities for reconnaissance, denial-of-service, and man-in-the-middle attacks, whilst conventional VPNs and firewalls rely on static policies that lack the flexibility required by highly dynamic, intent-driven 6G services. To address these limitations, SAFE-6G integrates the SDP paradigm, the core idea behind the Safety Function, across both the Control Plane (CP) and the User Plane (UP), without altering standard 3GPP access procedures. By adopting a dark-infrastructure approach, protected services remain invisible to unauthorised entities and are exposed only after successful authentication, authorisation, and trust evaluation, enabling a strict authenticate-then-connect model at the service level [3], [4].

The SDP literature provides useful evidence for this architectural direction. An experimental evaluation of an SDP client–gateway framework shows that an SDP-secured network can remain resilient against port scanning and denial-of-service attacks, since unauthorised traffic is rejected before services become visible or accessible [1]. In the evaluated scenarios, the SDP-enabled architecture maintained substantially higher throughput under attack than a non-SDP configuration, whilst preventing port scanning from revealing open services. At the same time, SDP introduces a longer initial connection setup time, since identity verification, authorisation checks, controller–gateway coordination, and temporary access-rule creation must complete before the protected service becomes reachable. Classical SDP deployments typically realise these steps through Single Packet Authorization (SPA) and Mutual Transport Layer Security (mTLS) [3]. In SAFE-6G, equivalent guarantees are provided by standard 3GPP authentication of the UE and by the mutually authenticated WireGuard handshake between the UE and the VGF, so that no additional mechanism is introduced into the UE-to-AMF interaction [12], [13]. This trade-off matters for SAFE-6G, since strict encrypted PDU-session enforcement can strengthen protection and isolation, but must be balanced carefully against 3GPP procedures and the stringent 6G requirements for low latency, high availability, and service continuity [1], [2], [8].

Dynamic micro-segmentation sits at the centre of the Safety Function, where the SDPGW and VGF create individualised micro-perimeters that grant access only to the services required by a specific user context. Rather than exposing a broad network segment once a user has been authenticated, SAFE-6G aims to provide the minimum visibility and connectivity needed for a given session, service, and trust level. In the classic SDP model, access provisioning is governed by a central controller that authorises resource access only after mutual verification of device and user credentials, keeping critical network components hidden from external entities [2]–[4], [6]. SAFE-6G retains this principle but distributes the responsibilities: the Local AI Agent determines whether, and at which safety level, access should be granted on the basis of near-real-time telemetry and the calibrated trust context, whilst the SDPCF establishes the corresponding protected session. The Safety Function thereby replaces static, rule-based enforcement with adaptive, trust-aware protection. This shift matters for 6G, where the disaggregation of network functions and the involvement of multiple stakeholders call for more resilient, adaptive, and user-centric safety mechanisms.

2.1 SAFETY FUNCTION FOUNDATIONS AND RELEVANCE TO SAFE-6G

The Safety Function rests on the principle that services should not be visible until identity, authorisation, and policy conditions have been verified. In the classic SDP architecture, the SDP controller acts as the central element responsible for authentication, certificate handling, policy distribution, and the coordination of secure connectivity between initiating and accepting hosts. The initiating host is the authorised client requesting access to a protected service, whilst the accepting host, or gateway, is the enforcement point that keeps the service hidden until access is explicitly allowed [1], [3]. SAFE-6G maps this tripartite structure onto its own components whilst distributing the responsibilities of the controller. The decision on whether, and at which safety level, access is granted is taken by the Local AI Agent on the basis of the calibrated trust context; the SDPCF manages the resulting protected session; the UE acts as the initiating host; and the SDPGW and VGF jointly form

the accepting side, with the SDPGW restricting the session to the authorised path and the VGF terminating the encrypted tunnel.

A central element of the SDP model is that a protected service remains hidden until the initiating host provides a valid authorisation signal and the controller approves the corresponding access policy. In classical SDP terminology, this is associated with SPA; in SAFE-6G, it serves as a conceptual reference for default-deny service exposure rather than as a replacement for 3GPP UE registration or AMF-based access procedures. This behaviour matters for SAFE-6G because it allows the network to maintain a default-deny posture whilst still supporting dynamic, on-demand access for legitimate users: the protected service does not remain open whilst filtering unwanted traffic, but stays concealed and opens only narrowly defined paths once trust has been established. Consider a UE requesting access to a protected virtual-reality service after it has already attached to the 3GPP network. Once the AI Agent has determined that access should be granted, the SDPCF establishes the protected session and installs on the SDPGW, over PFCP, forwarding rules that permit traffic only towards the VGF. If another UE attempts to reach the service without valid tunnel credentials, its traffic is either discarded at the SDPGW or silently ignored by the VGF, so that the existence of the service is never revealed [1], [3], [12].

The SDP protection model is further strengthened by mutual authentication and by binding the access path to the authorised service. Conventional SDP deployments commonly rely on mTLS for this purpose [3]. In SAFE-6G, both ends of the protected path are authenticated through complementary mechanisms: the UE is authenticated by the 5G core through standard 3GPP procedures, whilst the WireGuard handshake mutually authenticates the UE and the VGF on the basis of the credentials provisioned for the session [12], [13]. The encrypted PDU-session path is thereby bound to the authorised service context. These mechanisms support fine-grained trust enforcement, in which user identity, the requested application, the required level of trustworthiness, and permitted tunnel use all form part of the safety decision [3], [4], [6].

2.2 INTEGRATION OF SDP WITH PROGRAMMABLE NETWORKS

Integrating SDP with software-defined networking offers a strong conceptual reference for SAFE-6G, since both environments rely on programmability, the logical separation of functions, and policy-driven control [2]. In SDN, the controller manages forwarding behaviour whilst switches execute forwarding rules. This programmability, however, also introduces risks related to controller compromise, malicious flow manipulation, table exhaustion, and the exposure of services through insufficiently protected forwarding paths. SDP complements SDN by adding an explicit security orchestration layer that restricts which clients can even become visible to protected services, allowing SDP to act as a security perimeter built from logical components rather than fixed hardware appliances [2]–[4]. In SDN environments, this is achieved by placing gateway functions close to SDN switches and using the controller and gateway logic to authorise flows before granting service access, which hides network resources behind controlled gateways in a scalable manner [2]. SAFE-6G follows the same conceptual direction, but extends it to a 6G context in which trust, intent, AI-driven evaluation, and cloud-native orchestration jointly determine when, where, and how a secure perimeter is established.

Previous SDP work also highlights the importance of gateway placement [2], [3]. A client-to-server deployment can reduce routing delay because the accepting host sits directly on the server, but this weakens concealment because the protected service remains directly exposed. A client-to-gateway deployment adds a gateway element that may introduce some delay, but provides stronger protection, since the service stays hidden behind a controlled enforcement point. SAFE-6G adopts the same security-oriented reasoning: the SDPGW acts as a specialised gateway that can protect one or more services or network functions, whilst the SDPCF manages the protected sessions and the VGF provides secure tunnel establishment. The number and placement of gateways therefore become design parameters that must balance security, delay, resource consumption, and scalability. A single SDPGW protecting all applications in a regional edge domain reduces orchestration cost, since only one gateway needs to be managed, but it may become a bottleneck during high-traffic periods or when many secure tunnels are requested simultaneously. Conversely, dedicating an SDPGW to each sensitive service strengthens isolation and reduces the failure domain, but consumes more compute and memory resources. SAFE-6G therefore treats gateway placement as a resource-aware decision rather than a fixed design choice. The decision is taken by the Local AI Agent, which sits above the SDP stack as the policy decision point of the Safety Function, on the basis of the target application requested through CoCo and the topology and resource information supplied by the nApp layer, as detailed in Section 5.3.

The SDN-SDP integration results are consistent with the SDP evaluations discussed above. In the SDN-SDP case, the protected architecture retained a significant share of legitimate traffic during a denial-of-service bandwidth attack and blocked port-scanning attempts through its default-deny gateway policy [2]. Together with the setup-time overhead observed in [1], these results support the decision of SAFE-6G to use SDP for safety enforcement, whilst highlighting the need to manage explicitly the performance cost of strict encrypted PDU-session enforcement.

3 TRUST FUNCTION DEFINITION AND ROLE IN SAFE-6G

The Safety Function is formally defined as the entity responsible for enforcing resource-isolation policies based on user intent within the SAFE-6G framework. Its primary role is to bridge high-level trustworthiness requirements and the concrete network configurations needed to achieve them: the function determines not only whether access can be granted at the requested level of trustworthiness, but also how that decision translates into practical enforcement actions, such as controlled visibility, secure tunnelling, gateway activation, micro-segmentation, and resource-aware isolation. As a reactive and adaptive component, it evaluates both user requirements and network conditions to determine the most appropriate enforcement strategy, preserving safety objectives without compromising the operational efficiency of the underlying infrastructure. This role follows the SDP principle that access should be granted on a need-to-know basis, only after authentication, authorisation, and policy checks have been completed. In SAFE-6G, the Safety Function applies this principle in a user-centric, service-aware manner. Rather than relying on a single static access decision, it evaluates the requested service, the current trust level, available resources, and operational risk before selecting the appropriate safety flavour and the corresponding encrypted PDU-session enforcement mode. The result is a granular, adaptive form of trust enforcement, in which the network can offer different degrees of isolation and tunnelling depending on the user context and the current state of the infrastructure [3], [4], [6].

To perform this role, the Safety Function draws on inputs from both CoCo and the network environment. The high-level semantic intents interpreted by CoCo provide context for the requested level of trustworthiness [5], whilst operational telemetry from the core network and the edge-cloud continuum supplies the evidence needed to assess feasibility. This includes session-level information from the SMF, mobility and access information from the AMF, policy context from the PCF, user-plane performance indicators from the UPF, and resource status from aerOS, collected and exposed through the DataOps platform and retrieved by the nApp layer at configurable intervals. Combining intent information with this near-real-time telemetry gives the Safety Function a practical view of the user request, the service context, and the infrastructure conditions under which encrypted PDU-session enforcement must be applied [7], [10], [14].

A key responsibility of the Safety Function is its contribution to trust computation and to the assessment of enforcement feasibility. The function receives the nLoTw, a multi-dimensional trust vector generated by CoCo through its AI-driven trust evaluation mechanisms [5]. Rather than enforcing this trust level directly, the Safety Function assesses its Trust Function Capacity, that is, the extent to which the requested trust requirements can be supported by the available infrastructure, considering factors such as processing capacity, memory availability, bandwidth consumption, and the operational status of deployed network functions. On this basis, it reports the safety flavour it can feasibly enforce, which CoCo combines with the intended actions of the other Trust Functions to derive the cLoTw. The cLoTw thus reflects both the desired trust level and the practical limitations of the execution environment, keeping trust enforcement realistic, sustainable, and aligned with overall network stability. Consider a user request that calls for a High Safety flavour because the service handles sensitive industrial control data. If the nearest edge node has insufficient VGF capacity, or the

available SDPGW instance is already close to its processing threshold, the Safety Function may determine that the requested level cannot be fully enforced at that location. In that case, the request may be redirected to another edge domain with sufficient resources, or, where policy permits, the cLoTw may be reduced. SAFE-6G therefore does not enforce trust decisions blindly, but calibrates them against the reality of the infrastructure.

The Safety Function operates within a closed-loop interaction model that enables continuous adaptation to changing conditions. Communication with CoCo takes place asynchronously over a Kafka-based message broker, on the `safety_trust_score` topic, through which the Safety Function receives the nLoTw and cLoTw values and publishes its intended actions and execution reports. Through this channel, it reports the status of deployed safety measures, the effectiveness of enforcement actions, and any resource constraints affecting the realisation of the requested trust level. This feedback allows CoCo to revise trust assessments and adjust safety requirements where needed, creating an adaptive trust-management process that responds to fluctuations in network load, mobility conditions, service demands, and security threats. In this way, the Safety Function becomes a critical enabler of the SAFE-6G vision of intelligent, user-centric, context-aware trustworthiness management across the edge-cloud continuum.

3.1 REQUIREMENTS MAPPING

The requirements defined in D2.1 [15] establish the architectural, functional, and communication capabilities that the SAFE-6G Safety Function shall provide, with particular emphasis on SBA compatibility, integration with 5G/6G networks, dynamic deployment, secure tunnel establishment, UE authentication and authorisation, and dynamic policy enforcement. This deliverable translates these requirements into the Safety Function architecture, based on the Local AI Agent, the vApp and nApp layers, and the SDPCF, SDPGW, and VGF components. Table 1 summarises the mapping, which is detailed in the remainder of this section.

Requirement ID	Requirement	Realisation in the Safety Function	Section
REQ-FSAF-DES-F-M-01	SBA-compatible design	SDPCF operating as an enhanced SMF, registered with the NRF and interacting with the AMF, PCF, and UDM over the SBI	4, 6.1.1, 6.2.1
REQ-FSAF-DES-F-M-02	Hosting infrastructure, SBA	Components deployed as containerised network functions in the SBA-based core and edge-cloud continuum	4.2, 6.1
REQ-FSAF-DES-F-M-03	Interconnection with 5G/6G networks	PCC rule provisioning, N4/PCF control of the SDPGW, CAPIF-authorized API invocation	4.1, 6.2
REQ-FSAF-DES-F-M-04	Dynamic function creation	On-demand deployment of the SDPCF, SDPGW, and VGF through aerOS and TOSCA, triggered by the Local AI Agent	4.1, 6.2.3
REQ-FSAF-COM-F-M-05	VPN technology	WireGuard selected as the supported VPN technology, the enforcement path is protocol-	6.1.3

		independent, with other protocols supportable through VGF variants	
REQ-FSAF-COM-F-M-06	SDP tunnel establishment	Encrypted tunnel from the UE through the SDPGW, restricted to VGF-bound traffic, terminated at the VGF	6.1.2, 6.1.3, 6.2.2
REQ-FSAF-COM-F-M-07, -08	UE-specific authorised access	Individualised perimeters, SDPGW egress restriction, tunnel scoped to the authorised service	6.1, 7.1
REQ-FSAF-COM-F-M-09	Device and UE authentication	A4 server functions distributed across existing components: 5G-AKA through the AUSF and UDM, UE context from the UDM, target application from the SAFE-6G framework, WireGuard tunnel authentication	2.1, 6.2
REQ-FSAF-COM-F-M-10	Dynamic policies	Local AI Agent decisions based on the cLoTw, using models updated through the MLOps pipeline, propagated by the vApp layer and applied through PFCP session modification	4.1, 5.1, 6.2.3, 7

Table 1: Safety Trust Function Requirements

Regarding the SBA-compatible design (REQ-FSAF-DES-F-M-01), the Safety Function operates alongside the 5G/6G core and interacts with it through the SBA. In particular, the SDPCF operates as an enhanced SMF: it registers its NF profile with the NRF and interacts with the AMF, PCF, and UDM over the standard service-based interfaces, whilst the SDPGW provides the corresponding user-plane enforcement as an enhanced UPF [7], [16]. The Safety Function therefore complements the native 3GPP core procedures rather than replacing them. The hosting infrastructure requirement (REQ-FSAF-DES-F-M-02) is addressed by positioning the Safety Function within the SBA-based core and the edge-cloud continuum: its components are deployed as containerised network functions within the underlying infrastructure, and interact with the other core and edge components according to the service-based operational model.

The requirement for interconnection with 5G/6G networks (REQ-FSAF-DES-F-M-03) is addressed through the integration of the Safety Function with the core network and its policy and exposure mechanisms. The UE, service, and network context gathered by the nApp layer and received from CoCo serve as inputs to the Local AI Agent, whose decision is then realised through the core: the vApp layer provisions the PCC rules governing SMF and UPF behaviour, the SDPCF controls the SDPGW over the N4 interface using PFCP, and all invocations of external APIs are authorised through the CAPIF [17], [19]. The implementation of these interfaces is described in Section 6.2, whilst their validation at pilot scale will be carried out in WP5.

The dynamic function creation requirement (REQ-FSAF-DES-F-M-04) is addressed through the dynamic, resource-aware architecture presented in this deliverable. Upon selecting a safety flavour, the Local AI Agent instructs the vApp layer to deploy the SDPCF, SDPGW, and VGF on demand, through the aerOS platform and the corresponding TOSCA service templates, in the domain of the continuum selected on the basis of path cost and resource availability [10]. This approach is particularly important for the SAFE-6G methodology, since safety enforcement is not treated as a permanently fixed network

configuration, but as a function that can be instantiated and adapted to the requirements of individual users, services, and network conditions.

The communication requirements are addressed through the secure connectivity and SDP mechanisms of the function. The VPN technology requirement (REQ-FSAF-COM-F-M-05) was refined during implementation: WireGuard was selected as the supported VPN technology of the VGF, whilst the remaining components were kept independent of the tunnelling protocol, so that additional protocols can be supported through corresponding VGF variants. The rationale for this refinement is given in Section 6.1.3 [12]. The SDP tunnel establishment requirement (REQ-FSAF-COM-F-M-06), which calls for a protected tunnel between the UE and the SDPGW, is realised jointly by the SDPGW and the VGF, which together form the accepting side of the SDP model: the SDPGW admits protected traffic exclusively towards the VGF, and the VGF terminates the encrypted tunnel established with the UE. The resulting architecture combines SDP-based resource protection and concealment with encrypted connectivity between the UE and the domain hosting the protected service.

The requirements for UE-specific authorised access (REQ-FSAF-COM-F-M-07 and REQ-FSAF-COM-F-M-08) are directly reflected in the individualised access-control model. Once the required authentication, authorisation, and trust conditions have been satisfied, the UE is granted access only to the dedicated services, applications, or resources for which access has been authorised. This is enforced at two levels: the SDPGW restricts the protected session to the path towards the VGF, and the tunnel established by the VGF is scoped to the authorised service, rather than providing unrestricted access to the underlying network.

The device and UE authentication requirement (REQ-FSAF-COM-F-M-09) was defined in D2.1 around an A4 server storing device and UE information together with the requested target application [15]. During the architectural evolution of the Safety Function, the functionality of the A4 server was retained, but it is no longer concentrated in a dedicated component. Instead, it is distributed across existing 3GPP and SAFE-6G components, each acting as the authoritative source for its part of the information. Device and UE authentication is performed by the native 3GPP security mechanisms, including 5G-AKA, through the AUSF and UDM [13], and the device and subscription context is retrieved by the SDPCF from the UDM over N10 [7]. The user information and the requested target application are provided with the service request through the SAFE-6G trust management framework and consumed by the Local AI Agent, which takes the authorisation decision on the basis of the cLoTw. Session-specific credentials are generated by the VGF, which authenticates the UE at the tunnel level through the WireGuard handshake [12], whilst the session state maintained by the SDPCF and the execution reports published on the message broker provide the corresponding record of granted access. This distribution avoids duplicating identity data already held in the 5G core, keeps a single authoritative source for each type of information, and avoids introducing a centralised store that would constitute an additional high-value target and single point of failure. Consequently, the Safety Function does not replace the native 5G authentication framework, instead, it introduces an additional user- and service-centric authorisation layer that determines which protected resources may be accessed after successful authentication.

Finally, dynamic policies (REQ-FSAF-COM-F-M-10) are a central element of the architecture. The Local AI Agent evaluates trust-related information, user and service context, network telemetry, and infrastructure conditions to determine the appropriate safety flavour, with the enforced decision derived from the cLoTw. The models underlying these decisions are kept up to date through the MLOps pipeline described in Section 5.1, which supports the continuous monitoring, retraining, and controlled redeployment of the Local AI Agent as new operational data becomes available. When relevant session or network events occur, or a revised cLoTw is received, the resulting decisions are propagated by the vApp layer to the deployed components, with session-level changes applied by the SDPCF to the SDPGW through PFCP session modification. This enables dynamic, context-aware adaptation of the required safety level, resource allocation, protected connectivity, and access permissions.

4 REFERENCE ARCHITECTURE

The Safety Function rests on a unified multi-layer architecture designed to provide modular, scalable trust enforcement across the edge-cloud continuum. By leveraging the SDP paradigm, the architecture establishes dynamic, individualised security boundaries that support the transition of the 5G/6G core towards a Zero-Trust model [4], [7]. The design is intentionally modular, keeping intelligence, orchestration, network context acquisition, and enforcement logically separated whilst operating as a coordinated closed-loop system. This separation matters because evaluating trust, selecting a safety flavour, deploying the required functions, and enforcing network-level configurations are distinct tasks with different operational requirements.

The architecture, shown in Figure 1, comprises three functional layers. The Local AI Agent forms the intelligence layer: it receives the translated user intent, in the form of trust input from CoCo, together with the network context supplied by the nApp layer, and assesses the trust context and enforcement feasibility in order to select the appropriate safety flavour. The vApp layer translates the resulting decision into orchestration actions, including the deployment, reuse, migration, or configuration of SDPCF, SDPGW, and VGF instances, and the provisioning of the corresponding policy rules in the core network. The nApp layer gathers the topology and resource information of the edge-cloud continuum on which path selection and resource allocation depend. This separation follows the common SAFE-6G Trust Function model, whilst keeping the Safety Function focused on SDP-based concealment, resource isolation, and encrypted connectivity.

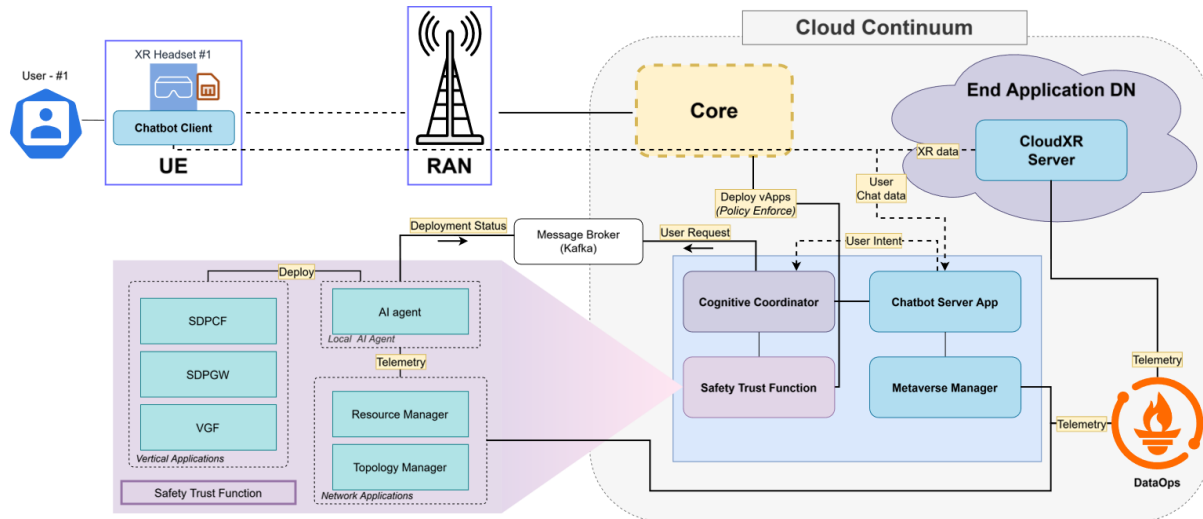


Figure 1: Safety Trust Function – High Level Architecture view

Figure 1 illustrates the proposed architecture, which extends the 5G core network with SDP security capabilities integrated into the standard session-management and user-plane domains. The UE, which in the considered use case serves the user's Extended Reality (XR) headset and hosts the chatbot client, connects to the network through the Radio Access Network (RAN) using standard 5G procedures. It communicates with the AMF over the N1 interface for Non-Access Stratum (NAS) signalling, whilst the RAN connects to the AMF over N2 for control-plane signalling and to the SDPGW

over N3 for user-plane traffic. The standard 5G core provides the required authentication, identity, policy, and network-function services through the Authentication Server Function (AUSF), Unified Data Management (UDM), Policy Control Function (PCF), Network Repository Function (NRF), and Network Exposure Function (NEF) [7].

The SDPCF, also referred to as the enhanced Session Management Function (eSMF), extends the standard SMF role with SDP-related session-control capabilities. Following UE authentication and the initiation of a PDU Session Establishment procedure, the SDPCF manages the protected session and applies the enforcement decisions produced by the Local AI Agent. A lightweight SDP Agent installed on the UE provides the client-side entry point for access to protected services: it triggers the required PDU session procedure through the standard 5G NAS and modem stack of the UE, and hosts the tunnel client through which the UE connects to the VGF. The agent does not itself implement the 3GPP PDU Session Establishment procedure; it assists the service-access request, whilst the session establishment is carried out by the standard UE and 5G core mechanisms [7].

The SDPGW, also referred to as the enhanced User Plane Function (eUPF), provides the corresponding user-plane enforcement. Once the protected session has been established by the SDPCF, the SDPGW applies the associated forwarding and security policies to the traffic received from the RAN over N3, forwarding it over N6 exclusively towards the VGF. The VGF provides a controlled gateway towards the protected application services, including Multi-access Edge Computing (MEC) and end-application services such as the CloudXR server. The architecture therefore distinguishes between the Default PDU path, which provides standard connectivity, and the Safety PDU path, which is subject to the controls of the Safety Function and carries the Safety Encrypted Tunnel towards authorised services. Figure 1 is technology-agnostic with respect to the tunnel; the current implementation employs WireGuard, as described in Section 6.1 [12].

The overall operation is thus based on the separation of standard 5G connectivity, SDP-based session control, and user-plane enforcement. The Local AI Agent determines whether, and at which safety level, access to a protected service is granted; the SDPCF establishes the corresponding session; the SDPGW acts as the enforcement point that rejects unauthorised traffic by default; and the VGF establishes secure tunnels between the authorised UE and the domain hosting the protected service, keeping communication encrypted and logically isolated [1]–[3]. Operation across these components follows a continuous feedback loop: the Local AI Agent analyses the trust input from CoCo together with the network context supplied by the nApp layer and determines the required safety flavour, after which the vApp layer deploys and configures the SDPCF, SDPGW, and VGF through the underlying orchestration framework and provisions the corresponding policy rules in the core network. Through this coordinated process, the architecture creates dynamic, context-aware safety boundaries tailored to each user and service. The detailed message flow is described in Section 4.1.

4.1 COMPONENT INTERACTION AND MESSAGE FLOW

The components of the Safety Function interact through a controlled authenticate-then-connect sequence, in which trust input, network context, and resource availability are combined into a safety decision before any protected access path is created. Figure 2 depicts this interaction between the

three layers of the Safety Function (the Local AI Agent, the vApp layer, and the nApp layer) and the external entities with which they cooperate: the message broker of the SAFE-6G trust management framework, the CAPIF, the aerOS orchestration platform, the DataOps platform, and the 5G core.

The interaction is underpinned by a continuous context-acquisition process. At configurable intervals, the nApp layer queries the DataOps platform to gather the currently available resources and the up-to-date topology of the edge-cloud continuum. This background loop runs independently of any individual service request, ensuring that the Safety Function always holds a recent view of network and resource conditions, so that safety decisions can be taken without first waiting for data collection.

The decision process is initiated by trust input from CoCo, published through the message broker of the SAFE-6G trust management framework [5]. The Local AI Agent first consumes the nLoTw value, the non-calibrated level of trustworthiness, on the `safety_trust_score` topic, which provides an initial indication of the trustworthiness required for the requested service. The AI Agent then requests the current resource and topology information from the nApp layer, which returns the most recent view gathered from DataOps. Combining the initial trust indication with this network context, the AI Agent selects the vApp flavour, that is, the safety flavour and the corresponding configuration of the SDPCF, SDPGW, and VGF, that is both appropriate to the trust requirement and feasible given the prevailing resource and path conditions. The selected flavour is reported to the message broker as the intended action of the Safety Function.

This report forms part of a calibration process spanning all Trust Functions of SAFE-6G. CoCo collects the intended actions reported by each function, assesses their combined effect on the trustworthiness of the service, and computes the calibrated level of trustworthiness, cLoTw, which it publishes on the same topic. The AI Agent consumes the cLoTw value and determines the flavour to be enforced on its basis. This two-stage exchange ensures that the enforcement applied by the Safety Function is coherent with the actions of the other Trust Functions and reflects the calibrated trust assessment of the framework as a whole, rather than an isolated local decision.

With the perimeter deployed, the protected session is established over it. The protected DNN, cloudxr-server in the current deployment, is a dedicated DNN distinct from the default internet DNN, with its own UE address pool. The UE, a programmable CPE in the current deployment, therefore keeps its default PDU session towards the internet DNN, whilst the SDP Agent triggers an additional PDU session towards the protected DNN through the standard 3GPP procedures with the AMF, corresponding to the Default and Safety PDU paths of Figure 3. For this session, the SDPCF acts as the serving SMF and installs the corresponding forwarding rules on the SDPGW over N4. The SDPGW applies default-deny gating to the protected session, forwarding its traffic only towards the VGF, as part of the user-plane policy rule enforcement defined for the UPF [7]; the WireGuard tunnel between the UE and the VGF then becomes the sole ingress through which the hidden service can be reached. This additional enforcement layer does not replace or alter the standard interaction of the UE with the 5G core; it adapts the client-gateway SDP model from the literature to the SAFE-6G architecture and its multi-layer orchestration approach [1], [3].

Once the encrypted tunnel is active, the Safety Function continues to monitor the session through the periodic resource and topology updates collected by the nApp layer and through subsequent nLoTw and cLoTw values published on the message broker. If the observed conditions remain aligned with the selected flavour, the configuration is maintained. If the level of trustworthiness changes, the user moves to a different network area, performance degrades, or an attack indicator appears, the AI Agent re-evaluates its decision. The AI Agent determines the required action, whilst its execution is distributed across the responsible components. Strengthened or relaxed enforcement is applied by the SDPCF through PFCP session modification and by the VGF through peer reconfiguration. The deployment of components in a different domain is carried out by the vApp layer through aerOS. The transfer of the UE to a new SDPGW is performed by releasing the protected PDU session and establishing a new one towards the SDPCF and SDPGW serving the new domain, triggered through the SDP Agent. Seamless relocation without session interruption, following the 3GPP session and service continuity modes, is discussed in Section 7.1.4. At service termination, the temporary rules and tunnel resources are removed, preventing stale authorisations from lingering in the network [3], [4].

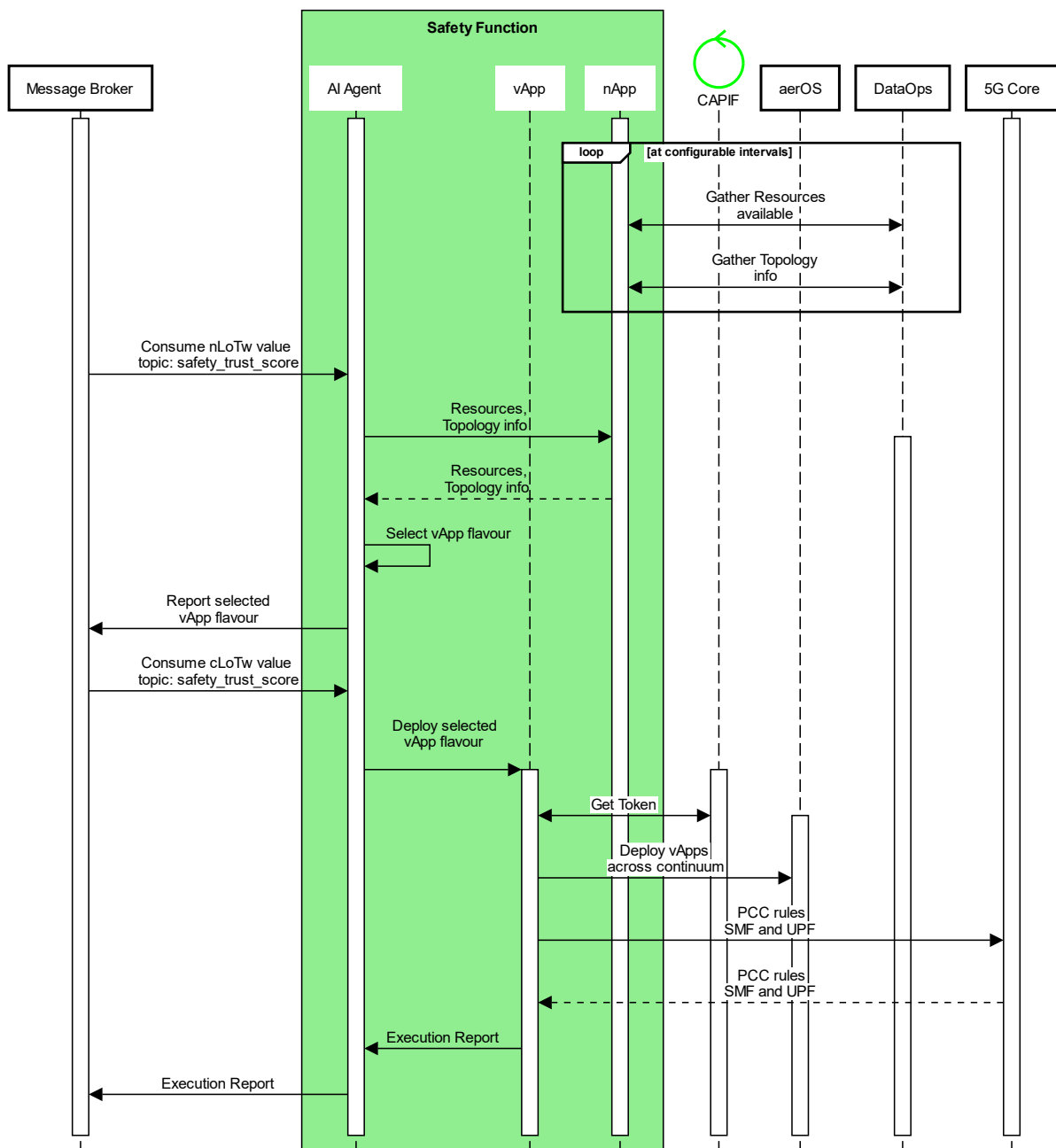


Figure 2: Safety Trust Function – Sequence Diagram

4.2 DEPLOYMENT ASSUMPTIONS AND DESIGN CONSTRAINTS

The architecture assumes a cloud-native execution environment, such as Kubernetes, in which the safety components are deployed as containerised network functions or microservices [9]. This assumption is necessary because the selected safety flavour may require the rapid instantiation of an SDPGW or VGF close to the service location, edge domain, or user-plane path. It is also assumed that the aerOS orchestration platform provides lifecycle management, placement, and resource status across the continuum, exposed to the Safety Function through the DataOps platform [10], [20]; without these capabilities, the Safety Function could not guarantee that the requested enforcement

level can be realised within the available infrastructure capacity. In the pilot deployment, for instance, the SDPGW and VGF are packaged as containerised functions. When the Local AI Agent selects a High Safety flavour, the vApp layer can request aerOS to instantiate a new VGF near the target edge service; if the user moves to another edge domain, the Safety Function can trigger the migration or redeployment of the VGF closer to the new user-plane path. Cloud-native deployment thereby supports both user mobility and adaptive safety enforcement.

The design further assumes authenticated, API-driven integration between the Safety Function, the orchestration framework, the core network functions, and the trust management components. These interfaces must support the secure exchange of policy information, telemetry, service identifiers, trust scores, and deployment commands. In SAFE-6G, this is realised through the CAPIF, as implemented by OpenCAPIF, which provides service discovery and authorised data exposure between the User Service Node (USN) and the Network Service Node (NSN) [19], [21]. Identity management is another foundational assumption: the Zero-Trust model requires a reliable authentication and authorisation capability that can validate user identity, device identity, and service authorisation before the perimeter is opened. The Safety Function therefore relies on the 3GPP authentication of the UE and on the identity services of the SAFE-6G framework, together with the ability to bind these identity decisions to network-level enforcement actions [4], [6], [13].

Figure 3 presents the deployment view of the Safety Function within the SAFE-6G test infrastructure. The user interacts through the UE, which hosts the chatbot client and connects to the network through the RAN. The chatbot server is deployed after the core, like a MEC server, where it receives the user requests from which the trust requirements of the service are derived. The components of the Safety Function, shown in purple, are deployed as vApps across the continuum, in accordance with the decisions of the Local AI Agent. The End Application DN hosts the VGF alongside the CloudXR server, so that the encrypted tunnel terminates within the same data network as the protected application. This placement confines the segment between the tunnel endpoint and the application to a single, controlled domain, whilst the CloudXR server itself remains reachable only through the VGF.

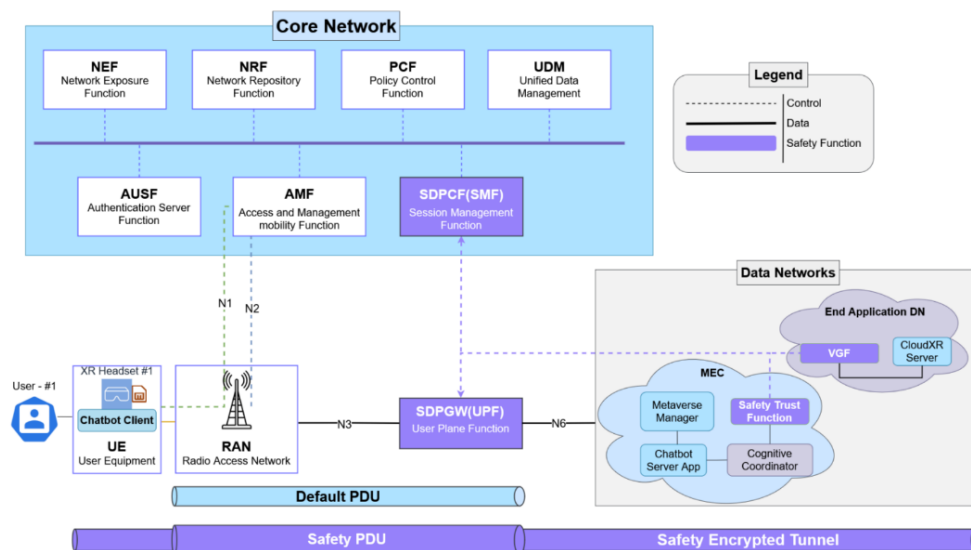


Figure 3: Safety Trust Function – Deployment view

5 TRUST FUNCTION LAYERS

This section examines the Safety Function across the three SAFE-6G Trust Function layers, explaining how the Local AI Agent, the vApps, and the nApps together transform trust decisions into encrypted PDU-session enforcement, whilst preserving the separation between intelligence, orchestration, and network context acquisition [9], [10].

5.1 LOCAL AI AGENT

The Local AI Agent serves as the intelligent core of the Safety Function, acting as the primary decision-making engine for safety enforcement. To handle the highly dynamic, multi-dimensional nature of 6G intent-based networking, the agent follows a Machine Learning (ML)-driven approach, integrated within a Machine Learning Operations (MLOps) pipeline. Unlike static, rule-based logic, which cannot account for the fluid behaviour of network traffic, user mobility, service relocation, and evolving threat vectors, ML allows the agent to identify non-linear patterns in user behaviour and network performance. This matters in 6G scenarios, in which the same user or application may require different levels of protection depending on the service context, trust score, mobility status, and current resource availability. The MLOps framework supports the continuous monitoring, retraining, validation, and controlled deployment of the models, so that the boundaries between safety levels remain adaptive and can be refined as new operational data becomes available [11], [22], [24].

5.1.1 AGENT INPUTS, DATA CONSUMPTION AND OUTPUT

The agent consumes data aggregated and refined by the DataOps platform from the core network and the edge-cloud continuum, retrieved through the nApp layer, together with the trust input received from CoCo over the message broker. Its inputs comprise:

- Normalised network logs: pre-processed telemetry from core NFs, including session characteristics from the SMF, mobility patterns from the AMF, and performance metrics such as throughput and latency.
- Continuum metrics: normalised resource utilisation values from DataOps, such as vCPU, memory, and disk usage.
- Topology information: the network view and candidate paths maintained by the Topology Manager, including their current path cost.
- Trust scores: the nLoTw vector, providing the initial indication of the requested level of trustworthiness, and the cLoTw, on which the enforced decision is based.
- Service request context: the user identity and the requested application associated with the service request.

The agent outputs an enforcement classification, that is, the safety flavour to be applied to the encrypted PDU session. This output is produced in two stages: the flavour initially selected on the basis of the nLoTw is reported to CoCo as the intended action of the Safety Function, whilst the flavour determined on the basis of the cLoTw is passed to the vApp layer, which applies the corresponding isolation and tunnelling policies. These inputs correspond to the variables of the safety flavour assignment function described in Section 7.1.

From an operational perspective, the model used by the Local AI Agent is governed through a controlled MLOps process covering dataset preparation, feature validation, model training, testing, deployment, monitoring, and retraining. The DataOps pipeline ensures that telemetry is cleaned, normalised, and transformed into meaningful features, whilst MLOps ensures that model behaviour remains valid as network conditions evolve. This matters because 6G networks are expected to support new traffic patterns and use cases for which historical data may initially be limited, so the Safety Function must improve its classification accuracy over time whilst maintaining the explainability and traceability of safety decisions, in line with evidence that explainable-AI integration can improve both the reliability and the interpretability of ML-based classification in 6G network-slicing contexts [22]–[24].

5.2 VERTICAL APPLICATIONS (VAPPS)

The vApp layer acts as the orchestration layer of the Safety Function, converting the decision produced by the Local AI Agent into deployable network services and configuration actions. In practice, this means determining which safety-related functions must be instantiated, where they should be placed, how they should be configured, and how long they should remain active. This orchestration role is essential in a cloud-native 6G environment, since encrypted PDU-session protection may need to adapt dynamically to trust updates, mobility events, traffic anomalies, or service requirements [9], [10]. The vApp layer therefore coordinates the safety components according to the required enforcement mode:

- Low Safety flavour (baseline encrypted PDU session): establishes authenticated, encrypted connectivity under a baseline tunnel protection profile, applying standard policy verification and maintaining readiness to escalate should telemetry or service sensitivity change.
- Medium Safety flavour (enhanced encrypted PDU session): applies an enhanced tunnel protection profile with tighter credential management, complemented by elevated monitoring, intermediate access controls, and stricter policy binding around the authorised UE-service path.
- High Safety flavour (strict encrypted PDU session): enforces maximum isolation through SDP-based concealment, applying a strict tunnel protection profile with session-bound credentials and a tunnel scoped exclusively to the protected application.

A Low Safety session may be sufficient for a non-sensitive content-preview service, and a Medium Safety session may suit an authenticated metaverse collaboration room, whilst a High Safety session may be required for an industrial control dashboard or a safety-critical digital twin, where unauthorised visibility or disruption could have operational consequences.

To realise these encrypted PDU-session modes, the vApp layer orchestrates the following components:

- SDPCF: acts as an enhanced SMF, managing the protected sessions.
- SDPGW: acts as an enhanced UPF and secure gateway for the targeted applications.
- VGF: operates as a VPN gateway, responsible for the secure tunnel between the UE and the domain hosting the requested application.

Zero-Trust enforcement relies on the 3GPP authentication of the UE and on the identity services of the SAFE-6G framework, which verify user and device credentials before access is granted. On this basis, the vApp layer realises a controlled, service-specific perimeter that limits the exposure of the internal network regardless of the external access environment [4], [6].

The orchestration lifecycle of the vApp layer spans preparation, instantiation, configuration, monitoring, adaptation, and termination. During preparation, it receives the selected safety flavour and translates it into a set of required components and policies. During instantiation, it requests the deployment of the SDPCF, SDPGW, and VGF where needed. During configuration, it provides the components with the appropriate service identifiers, user authorisation information, tunnel parameters, and enforcement rules. During monitoring, it collects the operational state of the deployed functions and reports the outcome to the Local AI Agent in the form of an execution report. During adaptation, it modifies the configuration if the trust level changes, and during termination, it removes the temporary configuration and releases the associated resources. This lifecycle ensures that the safety perimeter is created only when needed, maintained only whilst valid, and removed once the service context no longer requires it. When the Local AI Agent selects Medium Safety enforcement for a collaborative training service, for instance, the vApp layer first checks whether an SDPCF and SDPGW already exist for the relevant domain: if so, it reuses them and reconfigures them with updated policy parameters; if not, it requests their instantiation through the orchestration platform. During the session, the vApp layer monitors the state of the deployed functions, and when the service ends, or protected access is no longer required, it triggers the termination of the temporary access configuration and releases unused resources [9], [10].

Figure 4 illustrates the orchestration of a protected PDU session, driven by the AI service layer, within the 5G core network. At the service layer, the AI Agent uses the available UE context to determine the required service and protection configuration, and passes the corresponding service and security context to the Function Lifecycle Management component of the vApp layer. This component coordinates the deployment and configuration of the required network functions through the aerOS platform, which manages the lifecycle of the Safety vApps deployed in the Kubernetes cluster, namely the SDPCF, SDPGW, and VGF [10]. The SDPCF manages the protected session, the SDPGW provides the user-plane gateway and enforcement functions, and the VGF provides the VPN gateway functionality required for the protected connectivity path. The standard AMF, SMF, and PCF remain part of the wider 5G core, providing the underlying registration, mobility, session, and policy-control capabilities, including for sessions outside the Safety PDU path. The resulting process establishes a controlled PDU-session path in which access to the protected service is enabled only after the relevant Safety Function components have been instantiated and configured. The protected service is therefore not directly or broadly exposed to the UE; instead, its traffic is carried through the SDPGW and the VGF, under the session control of the SDPCF, within the encrypted tunnel [7], [9], [10].

The source code of the SDPCF, SDPGW, and VGF is maintained in a [private repository within the SAFE-6G GitLab group](#). Access is restricted to consortium members, reflecting the proprietary status of these components.

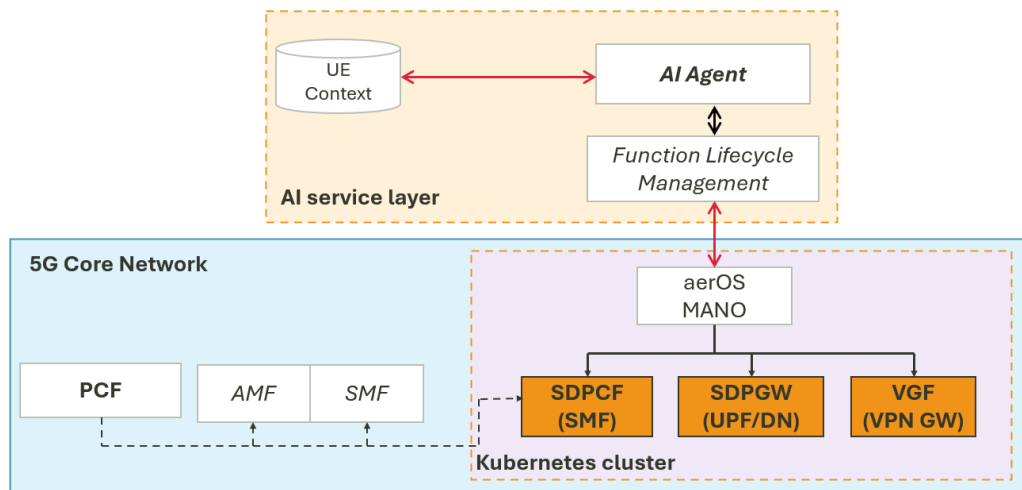


Figure 4: Safety Trust Function – Orchestration flows across vApps and SDP components

5.2.1 IMPLEMENTATION OF SDP CONTROLLER FUNCTION (SDPCF)

The SDPCF is a Python-based session management function that operates as the SMF of the SDP stack. Rather than being statically provisioned, it is deployed on demand: upon receiving a translated user request, the Local AI Agent assesses the trust context using the topology and resource information supplied by the nApp layer and determines the appropriate safety flavour, after which the vApp layer instantiates the SDPCF, together with the SDPGW and VGF, in the selected domain of the edge-cloud continuum. The SDPCF is therefore a product of the AI Agent's decision, placed where the prevailing trust context, path cost, and resource availability best serve the user's safety requirements.

Once deployed, the SDPCF assumes the standard session management responsibilities of an SMF for the traffic it serves. It registers with the NRF as a 3GPP-compliant NF, advertising the protected DNN in its NF profile, so that the AMF selects it as the SMF for PDU sessions towards that DNN. It interacts with the PCF over N7 for policy control and with the UDM over N10 for subscription data, and manages PDU-session establishment, modification, and release for UEs directed to the protected service [7], [14]. Southbound, it controls the SDPGW, which acts as the UPF of the SDP stack, over the N4 interface using PFCP, establishing the user-plane sessions through which protected traffic is forwarded [18]. In coordination with the VGF, which manages the VPN layer, the SDPCF ensures that each PDU session is carried within an encrypted tunnel between the UE and the domain hosting the target application, under the tunnel protection profile corresponding to the safety flavour assigned by the AI Agent.

In the current deployment, the SDPCF is registered under the service name `safe6g-smf` and serves a dedicated DNN, `cloudxr-server`, associated with the CloudXR use case. UEs attached to this DNN are allocated addresses from a dedicated session subnet, isolating protected traffic from general-purpose sessions at the addressing level. Slice selection is bound to the DNN through the configured selection profile, and the SDPCF operates within the tracking areas defined for the test PLMN. Internally, the SDPCF maintains the state of each active PDU session, including its assigned safety flavour and the associated SDPGW and VGF configuration. Should the AI Agent revise its safety flavour assignment, in response to updated telemetry, a change in service sensitivity, or a detected risk indicator, the vApp

layer triggers the corresponding update, and the SDPCF modifies the affected sessions through PFCP session modification towards the SDPGW, whilst the VGF applies the revised tunnel parameters.

The SDPCF exposes an HTTP-based SBI on port 7777, consistent with the Open5GS service-based architecture with which it interoperates, and a PFCP endpoint on port 8805 towards the SDPGW. Inter-NF communication is carried over HTTP within the cluster network; progressive alignment with the SBA security framework of 3GPP TS 33.501 is discussed in Section 8.2.3. The SDPCF does not replace or interfere with the standard 3GPP registration and authentication procedures handled by the AMF; it operates as an additional session management layer, activated once the UE has successfully entered the 5G core, as described in Section 4.

The component is packaged as a Docker container and deployed into the target aerOS domain through a TOSCA service template, with placement determined by the AI Agent's assessment of path cost, resource capacity, and provisioned resource usage. It is instantiated prior to the SDPGW and VGF, as encoded in the TOSCA dependency relationships, so that session management is operational before the user-plane and tunnelling functions are brought up. Access to the [SDPCF repository](#) is restricted to consortium members, reflecting its proprietary status and ongoing patent protection.

5.2.2 IMPLEMENTATION OF SDP GATEWAY (SDPGW)

The SDPGW is the user-plane function of the SDP stack, operating as the UPF for the PDU sessions managed by the SDPCF. Like the SDPCF, it is deployed on demand: once the AI Agent has determined the safety flavour for a user request, the vApp layer instantiates the SDPGW in the domain of the edge-cloud continuum selected on the basis of path cost, resource capacity, and provisioned resource usage. The SDPGW constitutes the enforcement anchor of the dynamically created perimeter, ensuring that traffic belonging to a protected PDU session can reach nothing other than the encrypted access path established for it.

Within the 3GPP user plane, the SDPGW fulfils the standard responsibilities of a UPF. It terminates the N3 interface towards the RAN on the standard GTP-U port 2152, decapsulating the user traffic received from the gNB, and is controlled by the SDPCF over the N4 interface using PFCP on port 8805, through which the SDPCF establishes, modifies, and releases the forwarding rules governing each PDU session [7], [18]. On the N6 side, the SDPGW forwards decapsulated traffic through a dedicated TUN interface, *upf-n6*, using kernel-level routing. It serves as the PDU-session anchor for the protected DNN, hosting the gateway address of the dedicated UE session subnet (10.46.0.1/24) configured in the SDPCF, alongside the general UE address pool (10.45.0.1/16).

The SDP enforcement behaviour of the SDPGW is realised through a restrictive N6 forwarding policy. When operating under the *vgf-only* restriction policy, the SDPGW permits user-plane traffic to egress exclusively towards the VGF endpoint, with all other destinations excluded from both the permitted destination set and the routing table. A UE attached to the protected DNN is therefore unable to reach the target application, or any other network resource, directly: the only reachable destination is the VGF, and the application is accessible solely through the encrypted tunnel that the VGF establishes with the SDP Agent on the UE. The UE does not move its existing connectivity onto the protected DNN.

The protected session is established as an additional PDU session, alongside the default session towards the internet DNN, as supported by the 5G system [7]. On the CPE, the SDP Agent routes the traffic towards the VGF endpoint through the protected session, so that traffic for the protected application is carried through the WireGuard tunnel over the protected DNN, whilst all other traffic continues to use the default session. Traffic addressed to any other destination is discarded at the SDPGW, so that the protected application remains concealed from the user plane and cannot be discovered or probed by a UE outside the authorised tunnel. This constitutes a network-layer realisation of the default-deny principle, enforced at the first user-plane hop after the RAN.

The N6 restriction is applied uniformly across all three safety flavours, providing a common baseline of concealment that confines every protected session to the encrypted access path, regardless of the assigned level. Higher flavours reinforce this baseline through stricter credential management and narrower tunnel scope at the VGF: differentiation between Low, Medium, and High Safety is thus realised in the tunnel protection profile applied by the VGF, as described in Section 7.1, rather than in the forwarding behaviour of the SDPGW. This separation keeps the user-plane enforcement simple, deterministic, and independent of the trust decision, whilst allowing the strength of protection to be calibrated at the tunnel layer.

The SDPGW is designed for deployment across heterogeneous domains of the continuum. Its exposure mode is configurable, supporting host networking, NodePort, LoadBalancer, ClusterIP, and UDP ingress modes, so that the same component can be instantiated on an edge node with direct access to the RAN or within a cloud domain behind a load balancer. In the current deployment, the SDPGW operates in host-network mode, with its PFCP node identifier derived from the host IP, to ensure that it is reachable by the SDPCF and the gNB at a stable address. Where NodePort exposure is used, N3 and N4 are mapped to ports 32152 and 30805 respectively, with the external traffic policy set to local in order to preserve the source addressing required for GTP-U and PFCP. Equally, the N6 forwarding mode can be configured for kernel routing, routing with NAT, UDP peer forwarding, or disabled operation, allowing the component to adapt to the network constraints of the target domain.

The SDPGW is implemented in Python and packaged as a lightweight Docker container, retrieved from the project GitLab container registry and deployed into the target aerOS domain through a TOSCA service template. Its resource footprint is deliberately modest, requesting 100 millicores of CPU and 128 MiB of memory, with limits of 500 millicores and 512 MiB, which supports rapid on-demand instantiation without placing significant load on constrained edge nodes. It is instantiated after the SDPCF, as encoded in the TOSCA dependency relationships, so that PFCP association can be established immediately upon start-up and no user-plane traffic is forwarded without a controlling session management function. Access to the [SDPGW repository](#) is restricted to consortium members, reflecting its proprietary status and ongoing patent protection.

5.2.3 IMPLEMENTATION OF VPN GATEWAY FUNCTION (VGF)

The VGF is the tunnelling component of the SDP stack, operating as a VPN manager responsible for establishing and maintaining the encrypted tunnel through which each protected session is carried. It is implemented on WireGuard, a modern kernel-level VPN protocol selected for its minimal attack

surface, high performance, and cryptographic design aligned with the concealment objectives of the Safety Function [12]. Like the SDPCF and SDPGW, the VGF is deployed on demand by the vApp layer into the domain of the edge-cloud continuum selected for the user request, and is positioned in proximity to the target application, so that the unprotected segment between the tunnel endpoint and the application is confined to the local domain.

The selection of WireGuard constitutes a refinement of requirement REQ-FSAF-COM-F-M-05, which in D2.1 calls for support of a variety of VPN protocols [15]. During implementation, WireGuard was selected as the single supported VPN technology on the basis of five properties that directly serve the objectives of the Safety Function. First, WireGuard does not respond to unauthenticated packets, so the tunnel endpoint itself remains concealed, extending the dark-infrastructure principle to the VGF without additional mechanisms. Second, its fixed cryptographic construction removes cipher negotiation, eliminating downgrade and misconfiguration risks and allowing the safety flavours to be differentiated through a consistent tunnel protection profile. Third, its compact codebase minimises the attack surface of the enforcement point. Fourth, its one-round-trip handshake and kernel-level implementation limit the setup and per-packet overhead, which matters for latency-sensitive services such as CloudXR. Fifth, its native support for endpoint roaming suits mobile users [12]. The requirement remains addressed at the architectural level: the SDPCF, the SDPGW, and the decision logic of the Local AI Agent do not depend on the tunnelling protocol, since the SDPGW restricts protected traffic to the VGF endpoint irrespective of the protocol it terminates. Support for additional protocols, such as IKEv2/IPsec or OpenVPN, would therefore require a corresponding VGF variant, without changes to the remaining components.

The encrypted tunnel extends from the WireGuard client on the UE to the VGF. Protected traffic generated by the UE is encrypted on the device, carried within the PDU session across the RAN, encapsulated in GTP-U over N3, and forwarded by the SDPGW over N6 exclusively towards the VGF endpoint, as enforced by the SDPGW restriction policy described above. The VGF decrypts the traffic and forwards it to the target application within the local domain. The combination of SDPGW-level destination restriction and VGF-level tunnel termination ensures that the application is reachable only by a UE holding valid WireGuard credentials provisioned for that specific session, and only through the encrypted path.

WireGuard employs a fixed, modern cryptographic construction based on the Noise protocol framework, using Curve25519 for key exchange, ChaCha20-Poly1305 for authenticated encryption, and BLAKE2s for hashing. Session keys are renegotiated automatically at regular intervals, providing perfect forward secrecy for all protected traffic. A property of particular relevance to the Safety Function is that WireGuard does not respond to packets that do not carry a valid handshake from a known peer: to an unauthenticated scanner, the VGF endpoint is indistinguishable from a closed port. This extends the concealment principle enforced by the SDPGW to the tunnel endpoint itself, so that neither the application nor its access gateway discloses its existence to unauthorised entities. Under load, WireGuard's cookie-based mechanism further mitigates handshake-flooding denial-of-service attempts [12].

Upon instantiation, the VGF prepares the peer configuration for the authorised UE. WireGuard does not rely on certificates: each peer is identified by a static Curve25519 public key, and mutual authentication is based on each side holding the public key of the other [12]. The SDP Agent generates its key pair locally on the UE and provides its public key to the VGF, which returns its own public key together with the permitted address range and the endpoint parameters; the private key of the UE therefore never leaves the device. The Safety Function is responsible for this exchange and for updating the UE accordingly: the SDP Agent installs the received configuration in its WireGuard client and, in the same step, configures the PDU session context for the protected DNN on the modem, as described in Section 5.3.1. Each UE is represented as a distinct WireGuard peer, with its permitted address range restricted to the addresses required to reach the target application, so that even an authenticated peer cannot use the tunnel to reach other resources in the local domain. All protected sessions share the same strong cryptographic foundation provided by WireGuard. Differentiation between safety flavours is realised through the tunnel protection profile applied by the VGF, which governs the lifecycle of peer credentials, the scope of reachable destinations within the tunnel, and the degree of binding between the tunnel and the individual session. Higher safety flavours apply progressively tighter credential management and narrower tunnel scope, reducing the window and extent of exposure should a credential be compromised. The parameters of each tunnel protection profile are summarised in Table 6.

In the current deployment, the VGF is reachable at 192.168.100.10 on port 31820, mapped from the WireGuard default port 51820 through a NodePort service, and configured as the sole permitted N6 next hop of the SDPGW. The component is packaged as a Docker container, retrieved from the project GitLab container registry, and deployed into the target aerOS domain through a TOSCA service template. It is instantiated alongside the SDPGW, following the SDPCF, as encoded in the TOSCA dependency relationships. Access to the [VGF repository](#) is restricted to consortium members, reflecting its proprietary status and ongoing patent protection.

5.2.4 INTERFACES AND INTEGRATION POINTS

The Safety Function integrates with three distinct domains: the 5G core, through standard 3GPP reference points; the user plane and tunnel layer, through which protected traffic is carried; and the SAFE-6G intelligence and orchestration layer, through which safety decisions are formed and enforcement components are deployed. This section describes each integration point, the protocols employed, and the role each plays in the end-to-end operation of the function.

5.2.4.1 5G CORE INTEGRATION

The SDPCF integrates with the 5G core as a 3GPP-compliant SMF, interoperating with the Open5GS network functions deployed in the SAFE-6G test infrastructure. All control-plane interactions follow the Service-Based Architecture, using HTTP-based SBI exchanges on port 7777. Upon instantiation, the SDPCF registers its NF profile with the NRF, advertising its service endpoint, supported DNN, and served tracking areas, thereby becoming discoverable by the other NFs of the core [14], [16]. The SDPCF operates under the direct communication model, interacting with peer NFs without intermediation by the Service Communication Proxy (SCP).

Reference **Peer NF** **Protocol** **Role in the Safety Function**
point

N11	AMF	HTTP (SBI)	Receives PDU-session establishment, modification, and release requests for UEs directed to the protected DNN; peer discovered via the NRF
N7	PCF	HTTP (SBI)	Retrieves session management policy and QoS parameters applicable to the protected session
N10	UDM	HTTP (SBI)	Retrieves subscription data and session management context for the requesting UE
Nnrf	NRF	HTTP (SBI)	NF registration and discovery, enabling the dynamically deployed SDPCF to join the core at runtime
N4	SDPGW	PFCEP (UDP 8805)	Establishes, modifies, and releases user-plane forwarding rules for each protected PDU session

Table 2: Core Network Interfaces & Protocols

Registration through the NRF is of particular significance for the Safety Function, as it allows an SDPCF instance deployed on demand to become operational within the core without static reconfiguration of the AMF or other NFs. In the current deployment, the SDPCF serves the dedicated cloudxr-server DNN within the test PLMN, covering three configured tracking areas, with slice selection bound to the DNN through the configured selection profile.

5.2.4.2 USER-PLANE AND TUNNEL INTEGRATION

Protected traffic traverses three user-plane interfaces between the UE and the target application. On N3, the SDPGW receives GTP-U-encapsulated traffic from the gNB on the standard port 2152. On N6, it forwards decapsulated traffic through a dedicated TUN interface using kernel-level routing, with the vgf-only restriction policy constraining egress to a single permitted next hop: the VGF. Between the UE and the VGF, the WireGuard tunnel carries all protected traffic in encrypted form, with the VGF exposed on UDP port 31820, mapped from the WireGuard default port 51820 through a NodePort service.

This arrangement produces a layered encapsulation along the protected path. Traffic leaving the UE is first encrypted by the WireGuard client, then carried within the PDU session and encapsulated in GTP-U across the transport network. At the SDPGW, the GTP-U layer is removed, and the still-encrypted WireGuard traffic is forwarded over N6 to the VGF, where the tunnel is terminated and the traffic is delivered to the target application within the local domain. At no point between the UE and the VGF is protected traffic exposed in plaintext, and at no point can it be routed towards any destination other than the VGF.

Interface	Endpoint	Protocol/Port	Functionality
N3	SDPGW	GTP-U / UDP 2152	User-plane traffic from the RAN
N4	SDPGW	PFCP / UDP 8805	Session rule provisioning from the SDPCF
N6	SDPGW → VGF	IP routing via TUN interface	Restricted egress, permitted towards the VGF only
Tunnel	UE ↔ VGF	WireGuard / UDP 31820	Encrypted transport of protected traffic

Table 3: User Plane Interfaces & Protocols

5.2.4.3 INTELLIGENCE AND ORCHESTRATION INTEGRATION

The formation of safety decisions and the deployment of enforcement components rely on interactions between the Local AI Agent, the nApp and vApp layers, and the external platforms of the SAFE-6G framework. The nApp layer supplies the AI Agent with the network context required for an informed safety decision: at configurable intervals, it gathers the available resources and the current topology of the edge-cloud continuum from the DataOps platform, with the Topology Manager maintaining the network view and the Resource Manager maintaining the record of available resources. This information is returned to the AI Agent on request, so that each decision is based on a recent view of network conditions without waiting for data collection.

Trust input is exchanged through the message broker of the SAFE-6G trust management framework, on the `safety_trust_score` topic [5]. The AI Agent consumes the nLoTw value, reports its selected vApp flavour as its intended action, and consumes the cLoTw value calculated by CoCo from the intended actions of all Trust Functions, on the basis of which the enforced flavour is determined. Execution reports are published on the same broker once enforcement is complete, providing the framework with an auditable record of the actions taken.

Enforcement is executed by the vApp layer. Prior to invoking any external interface, the vApp layer obtains an access token from the CAPIF, in accordance with the 3GPP Common API Framework, authorising its subsequent invocations towards aerOS and the 5G core [19]. It then instructs aerOS to deploy the SDPCF, SDPGW, and VGF in the domain selected by the AI Agent, resolving the corresponding TOSCA service templates, with component images retrieved from the project GitLab container registry, where they are published by the project CI/CD pipeline. In parallel, the vApp layer provisions the PCC rules governing SMF and UPF behaviour for the protected session in the 5G core [17]. Where a new protected session is required, the SDP Agent on the UE receives a directive to establish a PDU session towards the protected DNN, after which the standard 3GPP session establishment procedure is carried out through the AMF and the SDPCF.

Should a subsequent cLoTw value, or a change in network conditions reported by the nApp layer, lead the AI Agent to revise its decision, the vApp layer applies the corresponding update to the deployed components, with session-level changes applied by the SDPCF to the SDPGW through PFCP session modification.

5.3 NETWORK APPLICATIONS (NAPPS)

The nApp layer provides the network intelligence on which the decisions of the Local AI Agent rely. Rather than applying enforcement actions itself, it maintains an up-to-date view of the topology and resources of the edge-cloud continuum, gathered at configurable intervals from the DataOps platform, and supplies this view to the Local AI Agent on request. This allows safety decisions to account for path conditions, gateway load, and resource availability, so that the enforcement subsequently realised by the vApp layer and the SDPCF, SDPGW, and VGF translates into network behaviours that the infrastructure can sustain, such as controlled forwarding, the isolation of traffic flows, encrypted PDU-session establishment, resource-aware routing, and the restriction of unauthorised access attempts.

5.3.1 CORE UTILITY MODULES

To maintain a near-real-time, holistic view of the network, the nApp layer relies on two core modules:

- **Topology Manager:** maintains the network view of the 6G continuum, including the candidate paths between UEs and protected services and their current state and path cost, against which the Local AI Agent evaluates latency and other constraints.
- **Resource Manager:** monitors the physical and virtual assets of the continuum, such as vCPU, memory, and disk, so that the Local AI Agent can ensure that the overhead introduced by flavour-specific isolation mechanisms does not degrade the performance of the broader network ecosystem.

The nApp layer, comprising the Topology Manager and the Resource Manager, is implemented as open-source software and is publicly available in the relevant [SAFE-6G GitLab repository](#). Gateway placement illustrates how this information is used. Each request received from CoCo identifies the user, the device, and the target application, whose identifier corresponds to the protected DNN. On the UE side, the protected DNN is configured by the SDP Agent, upon receiving the directive to establish the protected session, the agent defines the corresponding PDU session context on the modem of the CPE [through its standard configuration interface, e.g. AT commands] and triggers the PDU Session Establishment procedure towards that DNN, which the network accepts because the DNN is provisioned in advance in the UE subscription held in the UDM [7]. For devices that cannot host such an agent, the same steering can be achieved through 3GPP URSP rules, as discussed in Section 7.1.1 [17]. On this basis, the Local AI Agent uses the topology information reported by the Topology Manager to identify the domain hosting the target application and the candidate paths towards it, together with their path cost, and the capacity information reported by the Resource Manager to determine where the required safety flavour can be sustained. It then selects the domain for the SDPGW and VGF accordingly, and the vApp layer deploys them in that domain through aerOS or reuses existing instances.

The nApp layer builds on the monitoring of the edge-to-cloud continuum managed by aerOS and interacts with the 6G Packet Core through Representational State Transfer (REST) APIs, so that the information on which each safety decision is based is traceable and aligned with network health and

resource allocation. This also keeps enforcement compatible with service performance expectations: strict encrypted PDU-session enforcement may require stronger isolation, additional tunnel processing, and more restrictive forwarding, which can raise latency or reduce throughput when resources are limited. The information provided by the Topology Manager and the Resource Manager therefore allows the Local AI Agent to select feasible paths, avoid overloaded gateways, and refrain from activating safety mechanisms that would destabilise the service [9]–[11].

Table 4 summarises the aerOS-facing API operations through which the Safety Function interacts with the orchestration environment. These interfaces matter because an encrypted PDU-session policy cannot be enforced in isolation: the Safety Function must first know which services exist, where they are deployed, whether they are ready, and whether they need to be reallocated, updated, deleted, or fully purged. The Next Generation Service Interfaces – Linked Data (NGSI-LD) entity interface provides discovery of, and read access to, the contextual information exposed through aerOS, and is used by the nApp layer to build its view of the continuum. The service-lifecycle endpoints under `/hlo_fe/services/{service_id}` are used by the vApp layer to deploy and manage the service components hosting the SDPCF, SDPGW, VGF, or related protected applications. The aerOS APIs thus form the orchestration bridge between a Safety Function decision and the service placement or lifecycle action required to realise the selected safety flavour [10], [20].

<i>Endpoint</i>	<i>Method</i>	<i>Description</i>
<code>/orionld/ngsi-ld/v1/entities</code>	GET	Used for discovery and reading NGSI-LD entities from aerOS
<code>/hlo_fe/services/{service_id}</code>	POST	Used to allocate/deploy a service
<code>/hlo_fe/services/{service_id}</code>	GET	Used to read service status and readiness
<code>/hlo_fe/services/{service_id}</code>	DELETE	Used to undeploy/deallocate a service
<code>/hlo_fe/services/{service_id}/purge</code>	DELETE	Used for full purge/cleanup when a complete removal is required
<code>/hlo_fe/services/{service_id}</code>	PUT	Used to reallocate a service
<code>/hlo_fe/services/{service_id}</code>	PATCH	Used to update the full-service template
<code>/hlo_fe/services/{service_id}/component/{service_component_id}</code>	PATCH	Used to update one specific service component

Table 4: aerOS APIs

Table 5 complements the aerOS lifecycle interfaces by identifying the Open5GS SBA APIs used for the registration, discovery, verification, update, and deregistration of network functions. These interfaces matter for the Safety Function because the SDPCF must be registered within the core network service registry in a controlled, traceable way. The NRF-facing APIs allow the SDPCF, acting as an eSMF, to publish its profile, refresh its availability, be discovered by other functions, and be cleanly removed once inactive. This registry-based interaction keeps safety enforcement aligned with the service-based principles of the 5G/6G core, rather than operating as an isolated overlay detached from the core control plane [7], [14], [16].

<i>Endpoint</i>	<i>Method</i>	<i>Description</i>
<code>/nnrf-nfm/v1/nf-instances/{nfInstanceId}</code>	PUT	Used by safe6g-smf to register or refresh its NF profile in the Open5GS NRF

/nnrf-nfm/v1/nf-instances/{nfInstanceId}	GET	Used to read back a specific registered NF instance if needed
/nnrf-disc/v1/nf-instances	GET	Used by other NFs for discovery, and sometimes useful for verification/debugging
/nnrf-nfm/v1/nf-instances/{nfInstanceId}	PATCH	Used to update parts of the NF profile if the implementation supports partial update
/nnrf-nfm/v1/nf-instances/{nfInstanceId}	DELETE	Used when the SMF deregisters or shuts down cleanly

Table 5: Open5GS APIs

The Open5GS APIs in Table 5 provide the control-plane visibility needed for coordinated Safety Function operation. The PUT operation registers or refreshes the profile of the relevant network function, allowing the NRF to maintain an up-to-date view of the available capabilities. GET operations support validation and troubleshooting, by reading specific NF profiles or discovering available NF instances. PATCH supports partial updates to an NF profile where the implementation allows it, for example to reflect a change in endpoint, capacity, or operational state. DELETE supports clean deregistration when a function shuts down or is redeployed. In the Safety Function context, these operations ensure that orchestration actions triggered through aerOS are reflected in the core network registry, so that service discovery, session management, and safety-aware enforcement remain consistent across the platform [7], [14], [16].

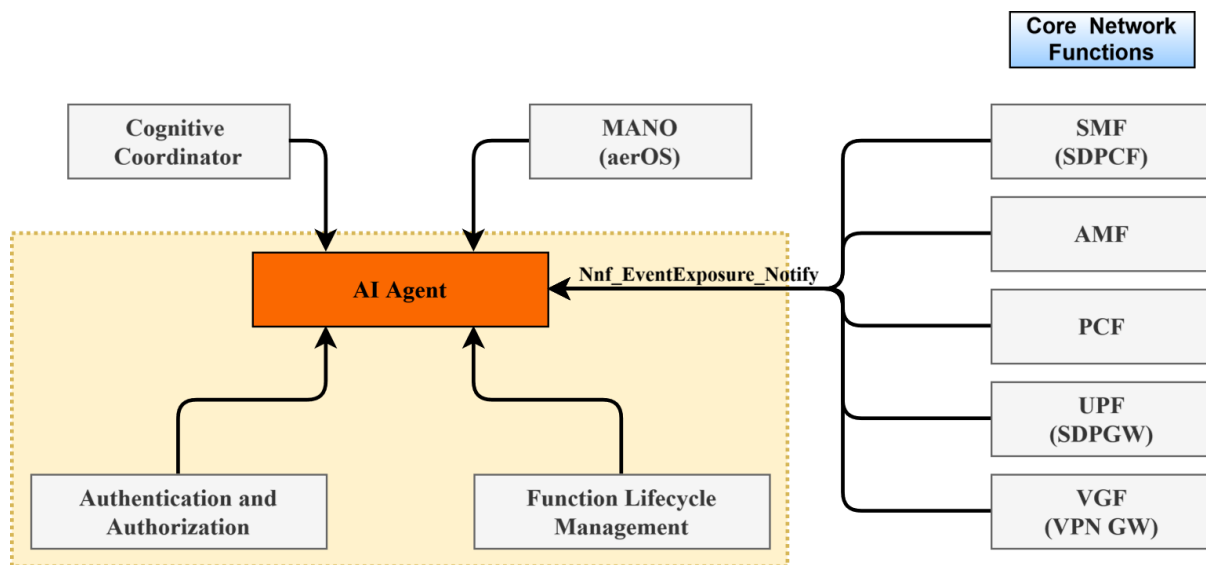


Figure 5: Safety Trust Function – Local AI Agent interactions with the system components

Consider a protected service reachable via two candidate edge paths. The Topology Manager may report that the shortest path is already congested and that the SDPGW serving it has limited remaining capacity. Drawing on resource availability information provided by the Resource Manager, the Safety Function identifies an alternative path exhibiting marginally higher latency but sufficient VGF and gateway capacity. Where High Safety is required, the Safety Function selects this second path, on the basis that a technically feasible and stable enforcement path is preferable to a lower-latency alternative liable to fail under the additional security overhead [8]–[10]. Figure 5 focuses on the Local

AI Agent and its interactions with the surrounding system components, showing how telemetry from the core network, orchestration status, resource information, and trust-related inputs combine to decide whether the current safety level should be maintained, strengthened, migrated, or terminated, representing the closed-loop intelligence required for adaptive PDU-session protection.

6 TRUST EVALUATION & ENFORCEMENT LOGIC

Trust evaluation and enforcement within the Safety Function are governed by a dynamic interplay between user intent and infrastructure reality, coordinated through CoCo and the Local AI Agent. Evaluation begins when user trust requirements are translated by CoCo into an nLoTw vector, expressing the desired safety posture in structured form. The Safety Function then reconciles this vector against its Trust Function Capacity, that is, the maximum safety level enforceable under the current virtualised resource quotas, such as Central Processing Unit (CPU), memory, storage, bandwidth, gateway availability, and tunnel capacity. The final enforcement decision is derived from the cLoTw, which reflects the intended actions of all Trust Functions and therefore prevents the Safety Function from committing resources in ways that could conflict with other functions or compromise overall network stability. To maintain this calibrated state, the function monitors, at configurable intervals, the session, mobility, and resource information described in Section 3, retrieved through the DataOps platform by the nApp layer [7], [8], [10], [14].

Trust levels are interpreted through mapping logic that determines the required encrypted PDU-session enforcement mode. A favourable trust context triggers Low Safety enforcement, establishing authenticated connectivity under a baseline tunnel protection profile with baseline policy verification. As trust requirements, service sensitivity, or risk indicators increase, the function transitions to Medium or High Safety enforcement. Medium Safety applies an enhanced tunnel protection profile together with intermediate access controls and stricter policy binding around the authorised UE-service path, whilst High Safety enforces maximum isolation through strict SDP-based concealment and a strict tunnel protection profile. Across all three modes, enforcement is maintained through the VGF, with the degree of concealment, the tunnel protection profile, and the policy binding calibrated to the prevailing trust context. This logic reflects the Zero-Trust principle that protected services should remain concealed until the relevant UE-service access path has been authorised on the basis of the calibrated trust context, whilst remaining fully compatible with standard 3GPP access and session-management procedures [3], [4], [6], [7].

6.1 ENCRYPTED PDU-SESSION ENFORCEMENT MAPPING

The Low Safety mode is intended for circumstances in which the trust context is favourable, the requested service is of limited sensitivity, and the network exhibits no indications of anomalous behaviour. Under these conditions, the Safety Function prioritises agility and quality of experience, whilst establishing authenticated connectivity under a baseline tunnel protection profile and applying baseline policy verification. The objective is not to dispense with security, but rather to avoid unnecessary overhead where the prevailing risk profile does not warrant stronger isolation. Enforcement in this mode rests principally upon identity confirmation, continuous telemetry observation, and sustained readiness to escalate should the operational context change [8].

The Medium Safety mode is selected where the system identifies elevated risk, heightened service sensitivity, or a requirement for stronger user traceability, risk mitigation, and access control. Under these conditions, the Safety Function applies an enhanced tunnel protection profile, accompanied by tighter authorisation checks and partial micro-segmentation around the requested service. Whilst the

SDPGW maintains the common default-deny baseline, the VGF is configured to sustain more tightly controlled encrypted traffic paths, and monitoring of the session is intensified. This intermediate mode balances operational efficiency against enhanced protection, and is expected to be applicable to a broad range of 6G services in which safety is a material concern, yet maximum isolation is not continuously warranted [4], [6], [8].

The High Safety mode is activated where the trust context indicates high sensitivity, elevated risk, suspected attack activity, or a pronounced requirement for isolation. Under these conditions, the Safety Function applies hard enforcement through the SDP stack: the protected service remains dark until the UE-service access context has been authorised, the SDPGW maintains strict default-deny behaviour, and the VGF establishes a controlled encrypted tunnel under a strict tunnel protection profile to isolate the communication path. This mode is the most resource-intensive of the three, as it entails additional gateway capacity, more frequent credential management operations, and more restrictive network policies; it nonetheless affords the strongest protection against reconnaissance, unauthorised access, lateral movement, and denial-of-service exposure [1]–[3].

By way of illustration, consider a UE operating initially under Low Safety enforcement whilst accessing a public training catalogue. Upon entering a private collaborative classroom, the Safety Function upgrades the session to Medium Safety enforcement and applies correspondingly stronger access control. Subsequently, when the same UE requests access to an industrial simulation linked to operational data, the increased service sensitivity and elevated telemetry volume prompt a further upgrade to High Safety enforcement, applying strict concealment and the strict tunnel protection profile at the VGF. Each transition is applied through PFCP session modification by the SDPCF and the reconfiguration of the corresponding VGF peer, without terminating the underlying PDU session. This progression demonstrates how the protection mode may evolve within a single continuous service experience, without altering the underlying 3GPP connectivity assumptions [3], [7].

In this mapping, the VGF can be read not only as a tunnel endpoint, but also as the mechanism that binds an authorised PDU session to a logical safety domain. Traditional VPN deployments commonly separate traffic into distinct domains, such as general enterprise access, confidential applications, or restricted infrastructure; within the Safety Function, however, these domains are not treated as static indicators of trust. A Low Safety flavour may correspond to a general enterprise VPN domain with SIM-based and policy-based validation, a Medium Safety flavour to a sensitive-service domain requiring additional enterprise identity evidence, and a High Safety flavour to a highly restricted VPN domain in which continuous verification, risk scoring, strict SDP concealment, and strong tunnel isolation are required. This preserves the Zero-Trust principle that access is granted to a verified UE-service context, not to a broad network simply because the UE has joined a VPN [3], [4], [6]. Table 6 summarises the mapping between these logical VPN-like service domains and the corresponding safety flavours, including their associated verification and enforcement mechanisms.

Logical VPN Domain	Safety Flavour	Typical Verification and Enforcement
General access	Low	3GPP authentication, allowed Data Network Name (DNN)/S-NSSAI context, telemetry observation, and baseline policy verification under a baseline tunnel protection profile.
Sensitive applications	Medium	3GPP authentication augmented by user identity verification, tighter policy binding, an enhanced tunnel protection profile, and elevated SDPGW/VGF monitoring.
Restricted services	High	Strict SDP concealment, risk evaluation, a dedicated or strongly isolated VGF path under a strict tunnel protection profile, and strict default-deny gateway behaviour.

Table 6: Logical VPN-like domains and corresponding safety flavours

The decision may therefore be expressed as a dynamic safety-domain assignment. Prior to the establishment of the protected PDU session, the Safety Function evaluates, in conjunction with the relevant 5G core NFs, the UE identity, the level of trustworthiness requested for the service, and the requested application and its associated policy, together with the prevailing path cost, available resource capacity, and provisioned resource usage across candidate enforcement paths. The outcome is not merely a VPN label, but the selection of the logical safety domain whose requirements correspond to the computed trust context, conceptually expressed as:

$$\text{Safety Flavour} = f(\text{User Identity}, \text{Requested Level of Trustworthiness}, \text{Requested Application}, \text{Path Cost}, \text{Resource Capacity}, \text{Provisioned Resource Usage})$$

In the above equation, the Requested Level of Trustworthiness is given by the nLoTw when the intended action is selected, and by the cLoTw when the enforced flavour is determined. The VGF subsequently realises the selected domain as the corresponding encrypted tunnel and VPN-like isolation context, whilst the SDPGW restricts the protected session to the authorised path towards the VGF. Network slicing thereby provides transport and resource isolation, the VPN-like domain provides connectivity separation, and the SDP policy determines whether the UE may enter that domain at the required safety level [3], [7].

6.2 ATTACK HANDLING AND RESILIENCE LOGIC

The Safety Function is designed to reduce exposure to several classes of attack that are particularly relevant to programmable, cloud-native networks. The dark-infrastructure principle mitigates port scanning, since unauthorised entities cannot discover the hidden service or determine which ports are open. Denial-of-service exposure is reduced because unauthorised traffic is dropped before reaching the protected service, limiting an attacker's ability to consume application resources. Man-in-the-middle and traffic-sniffing risks are reduced through authenticated, encrypted communication paths, and unauthorised lateral movement is constrained because access is granted only to the specific service and context authorised for the user, rather than to a broad network segment [3], [4]. Suppose that an unauthorised host attempts to scan a range of ports associated with a protected edge service: because the SDPGW follows a default-deny policy and the VGF does not respond to unauthenticated packets, the scan reveals no useful indication of open services. In a second case, the same host

generates repeated connection requests in order to exhaust service resources; since this traffic carries no valid tunnel credentials, it is dropped before reaching the protected application. This behaviour matches findings from the SDP literature, where protected networks did not reveal open ports and retained substantially higher legitimate throughput under denial-of-service conditions [1], [2], [12].

The resilience logic also covers cases where the desired safety enforcement cannot be fully applied. If a High Safety flavour is required but the necessary gateway or tunnel capacity is unavailable, as indicated by the resource and topology view supplied by the nApp layer, the Safety Function does not enforce an unstable configuration. Instead, the AI Agent reports the feasible flavour as its intended action, allowing CoCo to take the limitation into account when computing the cLoTw. Depending on policy and on the outcome of this recalibration, the system may delay the service, migrate it to a domain with sufficient capacity, apply a Medium Safety flavour temporarily where policy permits, or reject the request if the required safety level is mandatory [8]–[10]. This controlled fallback is essential for maintaining both trustworthiness and network stability.

Resilience is further supported by fallback and failure-handling logic at the component level. The architecture uses containerised microservices managed by aerOS, allowing failed components, such as the SDPCF, to be respawned automatically; PFCP recovery mechanisms allow the SDPGW to detect the restart and re-establish the association with the new instance [10], [18]. Mechanisms for preserving session state across such a restart, including redundant instances and state synchronisation, are discussed in Section 8.2.3. Internal feasibility guardrails also prevent the execution of high-overhead isolation policies when network capacity becomes critically constrained, deferring enforcement until resources can be reallocated. If the system fails to achieve the desired isolation, as reported through the execution reports published on the message broker, CoCo triggers a recalculation of the trust levels to shift the network into a more conservative operational state. This multi-layered approach keeps the Safety Function adaptive, protecting the edge-cloud continuum whilst maintaining fault tolerance across its virtualised components.

7 FUTURE RESEARCH PATHWAYS

The Safety Function is a core component of the SAFE-6G trust framework, providing SDP-based resource isolation and secure access across the edge-cloud continuum. Following a Zero-Trust approach, it keeps protected services concealed until the relevant user, service, and policy conditions have been satisfied, and realises this through the common SAFE-6G layered model: the Local AI Agent determines the appropriate safety flavour on the basis of the calibrated trust context; the vApp layer deploys and configures the SDPCF, SDPGW, and VGF across the continuum; and the nApp layer supplies the topology and resource information on which path and domain selection depend.

The work presented in this deliverable fulfils the objectives set for the Safety Function. The SDP paradigm has been integrated into the 5G/6G core as an overlay that preserves standard 3GPP procedures; three safety flavours have been defined and mapped to concrete enforcement mechanisms; the calibration loop with CoCo aligns user-centric trust requirements with the capacity of the infrastructure; and the SDPCF, SDPGW, and VGF have been implemented, integrated with an Open5GS-based core, and deployed on demand through aerOS, addressing each of the requirements mapped in Section 3.1.

Security in programmable, AI-driven networks is, however, a field in continuous evolution: threat models, standards, cryptographic primitives, and regulatory expectations all advance alongside the technologies they govern. The directions discussed in this chapter should therefore be understood as positive future research opportunities that build on the established SAFE-6G foundation. They concern the continued evolution of the Safety Function towards larger scale, over longer time horizons, and in closer coordination with the wider SAFE-6G framework and the broader 6G ecosystem, and several of them will be pursued through the WP5 pilot integration and validation activities.

7.1 ARCHITECTURAL INTEGRATION AND OPERATION

This section considers how the Safety Function, already integrated with the 3GPP architecture and with the other SAFE-6G Trust Functions, can be further extended as deployments grow in scale, services become more latency-sensitive, and users move across the network.

7.1.1 COORDINATION WITH 3GPP PROCEDURES

A deliberate design principle of the Safety Function is that it operates as an overlay on standard 3GPP procedures rather than as a modification of them. The UE registers, authenticates, and establishes PDU sessions through the AMF and the SDPCF exactly as in any 5G system, whilst SDP-specific actions, namely component deployment, PCC rule provisioning, restrictive user-plane forwarding, and tunnel establishment, are coordinated around these procedures. This choice keeps the function compatible with existing devices and core implementations, and the ordering dependencies between SDP-specific and standard steps have been identified and handled through the orchestration sequence described in Section 4.1: a dynamically deployed SDPCF registers with the NRF before the AMF selects it, PCC rules are provisioned before the protected session is established, and the tunnel is set up once the session and forwarding rules exist. As deployments grow in scale and heterogeneity, formalising these

dependencies as explicit state machines shared between the vApp layer and the 5G core provides a natural pathway for extending the current architectural foundation.

In the current implementation, traffic steering towards the protected DNN is handled by the SDP Agent on the programmable CPE, which receives directives to establish new PDU sessions. The 3GPP UE Route Selection Policy (URSP) mechanism offers a complementary, standardised path, allowing the network to instruct the UE to route specific application traffic into a designated DNN or slice [17]. Combining URSP with the existing agent-based approach would extend the reach of the Safety Function to off-the-shelf devices that cannot host additional software. More broadly, the question of whether future 3GPP releases should consider SDP-style concealment as a native capability of the service-based architecture, rather than as an overlay exposed through CAPIF and PCC rules, represents a promising research direction to which the SAFE-6G pilot experience can contribute evidence and architectural insight.

7.1.2 MULTI-FUNCTION COORDINATION AND DECISION PRECEDENCE

Within the SAFE-6G framework, the calibration loop with CoCo already ensures that the intended actions of the Safety Function are assessed alongside those of the other Trust Functions before enforcement proceeds, so that no function acts in isolation. As the number of Trust Functions and concurrent services grows, the reconciliation of competing objectives provides an increasingly rich research domain: isolation requirements, 3GPP session-management behaviour, and resource-optimisation goals may pull in different directions, as when a stricter perimeter consumes resources that another function would allocate differently, or when relocation for safety reasons interacts with continuity objectives. Formalising a precedence model for such cases, ranging from explicit priority rules to multi-objective optimisation in which the calibrated trust value acts as a constraint, offers a valuable extension of the existing calibration mechanism.

Stability over time is a related research opportunity. The architecture already supports transitions between flavours without terminating the underlying PDU session, as described in Section 7.1. Where the calibrated level of trustworthiness fluctuates around the boundary between two flavours, mechanisms from control theory, such as hysteresis thresholds and minimum dwell times for each flavour, offer a principled way to avoid unnecessary reconfiguration, and the pilot activities will provide the realistic trust dynamics against which such mechanisms can be tuned.

7.1.3 SCALABILITY AND ORCHESTRATION

The design of the Safety Function already provides the key ingredients for scalable operation: lightweight, containerised components, on-demand deployment, and the reuse of existing instances where the trust context allows, as described in Section 5.2. Scaling towards dense 6G deployments, such as event venues or industrial campuses in which thousands of UEs may request access to different protected services within the same period, opens further opportunities for optimisation and research. A central design axis is the granularity of the perimeter, as outlined in Table 7.

<i>Perimeter granularity</i>	<i>Isolation strength</i>	<i>Instantiation cost</i>	<i>Natural fit</i>
Per-session (dedicated)	Highest	Highest	High Safety, short-lived critical sessions
Per-user	High	Moderate	Users accessing several services under a stable trust context
Per-service	Moderate	Low per user	Services with many users of similar trust level
Pooled (shared)	Logical only	Lowest	Low Safety, high-density general access

Table 7: Safety Perimeter Granularity levels

Aligning perimeter granularity with the assigned safety flavour ensures that the cost of isolation is incurred where the trust context justifies it, and the lightweight resource profile of the enforcement components makes dense instantiation on constrained edge nodes feasible. Gateway placement is closely related: it is a variant of the facility and controller placement problems studied in the SDN literature, which are NP-hard in their general form [25]. The AI Agent already accounts for path cost, resource capacity, and provisioned resource usage when selecting a deployment domain, and extending this towards predictive placement based on historical demand, scheduled events, or mobility patterns provides a natural research pathway. At the level of the trust framework, batching, hierarchical calibration, and the reuse of calibrated values for equivalent contexts offer ways to keep calibration exchanges with CoCo efficient as session rates grow, and provide opportunities for joint investigation with the other Trust Functions.

The safety-performance trade-off inherent to SDP-based approaches has been explicitly accounted for in the design. Earlier SDP evaluations show strong resilience gains balanced against a longer initial connection setup [1], [2]. In the Safety Function, setup time is the sum of several contributions, each with a distinct optimisation path, as outlined in Table 8.

<i>Contribution</i>	<i>Nature</i>	<i>Optimisation direction</i>
Trust calibration (nLoTw → intended action → cLoTw)	Framework-level coordination	Reuse of calibrated values, asynchronous calibration
CAPIF authorisation	Token acquisition, amortised over a session	Token caching within validity period
Component deployment through aerOS	Container scheduling and start-up	Pre-staged images, warm pools, instance reuse
PCC rule provisioning	Control-plane signalling	Parallelisation with component deployment
PDU-session establishment	Standard 3GPP procedure	Inherited from the core
Tunnel handshake	One round-trip for WireGuard	Negligible relative to other contributions

Table 8: Safety Function Setup Time Optimization view

This decomposition shows that the protection mechanism itself contributes only marginally to setup time, and that further optimisation can primarily focus on anticipatory deployment and component reuse, both of which the current architecture already supports. Once the session is established, per-

packet overhead is modest: the double encapsulation of WireGuard within GTP-U reduces the effective maximum transmission unit, with WireGuard adding 60 bytes over IPv4 and 80 bytes over IPv6 [12]. The WP5 pilot measurements will further characterise these effects for high-throughput services such as CloudXR.

7.1.4 MOBILITY AND PERIMETER RELOCATION

The cloud-native design of the Safety Function already allows the VGF to be redeployed closer to a new user-plane path when a user moves to another edge domain, as described in Section 4.2, and WireGuard is well suited to such scenarios, as a tunnel remains valid when the outer address of the UE changes, with the VGF updating the peer endpoint upon receiving an authenticated packet from the new address [12]. Building on these capabilities, relocation strategies can be refined further. A perimeter may be retained at the cost of a longer path, relocated by releasing and re-establishing the session, or relocated in a make-before-break manner, mirroring the session and service continuity modes defined by 3GPP for PDU-session anchor relocation [7]. Extending these modes to SDP perimeters, which carry additional state such as tunnel credentials and restrictive forwarding rules, represents a promising area for future research, including mechanisms for pre-provisioning perimeter state in the target domain. Since relocation into a different domain may also alter the trust context and trigger recalibration with CoCo, mobility management and trust management are naturally coupled in SAFE-6G, and high-mobility scenarios, such as vehicular or drone-based services, represent an attractive test case for further investigating this coupling.

7.2 TRUSTWORTHINESS AND SECURITY RESILIENCE

This section considers the continued evolution of the AI-driven decision process, and the long-term strengthening of the Safety Function's own control and enforcement points.

7.2.1 DATA AVAILABILITY AND AI RELIABILITY

The MLOps pipeline described in Section 5.1 already provides the mechanisms for continuous monitoring, retraining, and controlled redeployment of the Local AI Agent, ensuring that the model can adapt as operational data accumulates. The reliability of any AI-driven trust function remains closely tied to the quality, freshness, and representativeness of its input data, and as 6G deployments mature, the diversity of operational conditions across pilots, domains, and vertical applications will continue to grow. An agent trained mainly on laboratory traffic may, for instance, benefit from adaptation once exposed to unusual but legitimate industrial traffic, whilst an overly conservative model may be slower to flag early indicators of a flooding attack [22]–[24]. Several complementary directions can further enrich the training base: synthetic data and digital-twin environments can supplement real-world data, including through controlled attack injection during pilot activities; federated learning allows models to be trained across domains where data sovereignty constraints apply [26]; and alignment with standardised analytics sources, such as the 3GPP NWDAF, offers a path towards more homogeneous telemetry across network functions [27]. Confidence-aware decision-making is a further refinement: where the confidence of the agent is low, for instance when inputs fall outside the distribution seen during training, defaulting to a more conservative flavour and flagging

the decision for review provides an additional safeguard and a useful direction for further investigation.

7.2.2 EXPLAINABILITY AND REGULATORY ALIGNMENT

Explainability has been considered from the early design stages of the SAFE-6G framework, with the AI-driven Trust Functions designed to support explainable outputs, systematic logging, and human-in-the-loop override mechanisms. The Safety Function contributes to this foundation directly: the selected flavour, the calibrated trust value, and each execution report are published on the message broker, forming an auditable record of the decision–enforcement cycle. Attaching a structured rationale to each decision, identifying whether it was driven by service sensitivity, a risk indicator, or the calibrated trust value, represents a natural extension that could further support operators and stakeholders where decisions affect service availability. Continued alignment with evolving regulatory guidance and standardisation activities therefore provides a constructive pathway for extending the framework’s applicability to future deployment contexts.

This design direction is reinforced by the EU Artificial Intelligence Act, which classifies AI systems used as safety components in the management and operation of critical digital infrastructure as high-risk, subject to requirements on transparency, human oversight, record-keeping, and robustness [28]. Whether a given trust function falls within this category depends on its deployment context; nevertheless, the design choices outlined above position the SAFE-6G framework favourably with respect to these obligations as the regulatory framework is progressively operationalised.

7.2.3 PROTECTION OF CONTROL AND ENFORCEMENT POINTS

The Safety Function already relies on orchestration-based recovery of its components and on PFCP restart detection between the SDPCF and the SDPGW, as described in Section 7.2. Building on these mechanisms, redundant SDPCF instances, secure synchronisation of session state, and controlled failover represent a natural pathway towards increasingly resilient operation, and the service-based architecture provides the building blocks for them: NF profiles registered with the NRF carry priority information supporting selection among redundant instances, and PFCP includes recovery mechanisms that allow peers to detect restarts and re-establish association state [16], [18]. For the SDPGW and VGF, credential protection, gateway hardening, and continuous monitoring of enforcement-point integrity will further strengthen the broader SAFE-6G security posture [4], [6]. Progressive alignment with the SBA security framework of 3GPP TS 33.501, together with the secure distribution of tunnel credentials to the SDP Agent and supply-chain integrity from the CI/CD pipeline to the deployed components, can support the continued evolution towards production-grade deployment [13]. These directions build upon, rather than alter, the design presented in this deliverable, and provide a coherent basis for the continued hardening and validation in WP5 and in subsequent research activities.

7.2.4 CRYPTOGRAPHIC AGILITY AND POST-QUANTUM READINESS

The selection of WireGuard provides the Safety Function with strong, modern protection today. Its fixed cryptographic construction eliminates cipher negotiation, and with it an entire class of

downgrade and misconfiguration risks, whilst implying that the adoption of new algorithms follows a protocol version transition. This consideration gains relevance in view of "harvest now, decrypt later" threats, in which encrypted traffic is recorded today for decryption once quantum capabilities mature, which is a concern for industrial services whose operational data has long confidentiality lifetimes. WireGuard's optional preshared key already provides a mitigation layer that remains robust against quantum attacks on the key exchange, and the standardisation of post-quantum key encapsulation by NIST, notably ML-KEM, opens the way for hybrid schemes combining classical and post-quantum algorithms [12], [29]. Within the Safety Function, the tunnel protection profile associated with each flavour provides a natural point at which such capabilities can be introduced progressively, beginning where long-term confidentiality matters most, so that the function can evolve alongside cryptographic standards without changes to its architecture. Further investigation of cryptographic agility and post-quantum mechanisms therefore represents a forward-looking pathway for maintaining the long-term relevance of the Safety Function.

Taken together, these pathways demonstrate that the Safety Function provides a solid and extensible foundation for user-centric safety in 6G networks. Its layered architecture, standards-aligned interfaces, and configurable enforcement mechanisms provide a strong basis for continued research and for absorbing advances in standardisation, AI, and cryptography as they emerge, ensuring that the results of SAFE-6G can remain relevant as the 6G ecosystem continues to mature.

8 CONCLUSIONS

The User-centric Safety Function constitutes a cornerstone of the SAFE-6G framework, providing the mechanisms needed to move from traditional, static network security towards a dynamic, intent-driven trust model. By integrating the SDP paradigm into the control and user planes of the 5G/6G core, the function establishes a dark-infrastructure approach that minimises the attack surface across the edge-cloud continuum. It links user trust requirements to individualised resource isolation, keeping communications secure, private, and resilient against the threat vectors that accompany increasingly distributed and programmable networks.

Architecturally, the function follows the common SAFE-6G Trust Function model, with three modular layers that together provide end-to-end trust enforcement. The nApp layer maintains the topology and resource view of the continuum on which safety decisions depend. The Local AI Agent combines this view with the trust input received from CoCo to select the appropriate safety flavour, with the enforced decision derived from the calibrated level of trustworthiness computed across all Trust Functions. The vApp layer then realises this decision, deploying and configuring the SDPCF, SDPGW, and VGF across the continuum and provisioning the corresponding policy rules in the core network. This layered structure allows the Safety Function to operate as a closed-loop system that adapts to changing network conditions, user behaviour, trust requirements, and infrastructure constraints, whilst ensuring that trust decisions remain feasible within the capacity of the underlying infrastructure.

Beyond its design, the deliverable has presented the realisation of the function. The SDPCF operates as an enhanced SMF interoperating with an Open5GS-based core, the SDPGW operates as an enhanced UPF that confines protected traffic to the path towards the VGF, and the VGF provides WireGuard-based encrypted tunnels towards the domain hosting the protected service. All components are containerised, built through a CI/CD pipeline, and deployed on demand through aerOS, and together they address the requirements established for the Safety Function, as mapped in Section 3.1.

By adopting a Zero-Trust, authenticate-then-connect approach, the Safety Function embeds trustworthiness throughout the service lifecycle. Through encrypted PDU-session enforcement, micro-segmentation, and SDP-based concealment, it provides a scalable and flexible mechanism for protecting diverse use cases, from industrial automation to immersive XR and metaverse environments, whilst contributing to the project's objectives for secure, adaptive, and human-centric 6G services. In doing so, it supports the broader shift from network security as a static boundary to network safety as an adaptive, continuous, user-centric capability. The same reasoning allows VPN-like domains to be understood as practical manifestations of safety flavours: rather than assigning a UE to a static VPN on the basis of its network location, the Safety Function evaluates the user identity, the requested level of trustworthiness, the requested application, and the available path and resource capacity, and binds the encrypted PDU session to the logical domain whose level of protection is justified. This provides a bridge between traditional VPN separation, 5G network slicing, and SDP-based Zero-Trust enforcement.

Building on these results, the Safety Function will be carried forward into the WP5 pilot integration and validation activities, where its operation at scale, under mobility, and in coordination with the other Trust Functions will be further examined, in line with the research directions outlined in Section 7.

9 REFERENCES

- [1] A. Moubayed, A. Refaey, and A. Shami, "Software-Defined Perimeter (SDP): State of the Art Secure Solution for Modern Networks," *IEEE Network*, vol. 33, no. 5, pp. 226–233, Sep./Oct. 2019.
- [2] A. Sallam, A. Refaey, and A. Shami, "On the Security of SDN: A Completed Secure and Scalable Framework Using the Software-Defined Perimeter," *IEEE Access*, vol. 7, pp. 146577–146587, 2019.
- [3] Cloud Security Alliance, "Software-Defined Perimeter (SDP) Specification v2.0," CSA SDP and Zero Trust Working Group, 2022.
- [4] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST Special Publication 800-207, National Institute of Standards and Technology, Aug. 2020.
- [5] SAFE-6G Consortium, "Cognitive coordinator, AI agents and user-centric functions" SAFE-6G Deliverable D[4.1], [06/2025].
- [6] R. Chandramouli and Z. Butcher, "A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments," NIST Special Publication 800-207A, National Institute of Standards and Technology, Sep. 2023.
- [7] 3GPP, "System Architecture for the 5G System (5GS); Stage 2," 3GPP TS 23.501, Release 18.
- [8] Smart Networks and Services Joint Undertaking, Test, Measurement and Validation Working Group, "White Paper: 6G KPIs – Definitions and Target Values," Version 1.0, Jan. 2025.
- [9] ETSI, "Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV Architecture towards 'Cloud-native' and 'PaaS'," ETSI GR NFV-IFA 029, V3.3.1, Nov. 2019.
- [10] ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration; Architectural Framework Specification," ETSI GS NFV 006, V4.4.1, Dec. 2022.
- [11] T. Bilen and M. Ozdem, "A Modular DTaaS Architecture for Predictive Slice Management in 6G Systems," arXiv:2511.01989, 2025.
- [12] J. A. Donenfeld, "WireGuard: Next Generation Kernel Network Tunnel," in *Proc. Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 2017.
- [13] 3GPP, "Security Architecture and Procedures for 5G System," 3GPP TS 33.501, Release 18.
- [14] Open5GS Project, "Open5GS Documentation." [Online]. Available: <https://open5gs.org/open5gs/docs/>.
- [15] SAFE-6G Consortium, "[Definition of Technical Requirements for user-centric 6G Trustworthiness]," SAFE-6G Deliverable D2.1, [December 2025].
- [16] 3GPP, "5G System; Network Function Repository Services; Stage 3," 3GPP TS 29.510, Release 18.
- [17] 3GPP, "Policy and Charging Control Framework for the 5G System (5GS); Stage 2," 3GPP TS 23.503, Release 18.
- [18] 3GPP, "Interface between the Control Plane and the User Plane Nodes," 3GPP TS 29.244, Release 18.
- [19] 3GPP, "Common API Framework for 3GPP Northbound APIs," 3GPP TS 23.222, Release 18.

- [20] aerOS Project, “aerOS: Autonomous, scalable, trustworthy, intelligent European meta Operating System for the IoT edge-cloud continuum.” [Online]. Available: <https://aeros-project.eu/>.
- [21] ETSI Software Development Group OpenCAPIF, “OpenCAPIF.” [Online]. Available: <https://ocf.etsi.org/>.
- [22] H. Sun et al., “Advancing 6G: Survey for Explainable AI on Communications and Network Slicing,” IEEE Open Journal of the Communications Society, vol. 6, pp. 1372–1412, 2025, doi: 10.1109/OJCOMS.2025.3534626.
- [23] S. N. Karahan, M. Güllü, D. Karhan, S. Çimen, M. S. Osmanca, and N. Barışçı, “Realistic Performance Assessment of Machine Learning Algorithms for 6G Network Slicing: A Dual-Methodology Approach with Explainable AI Integration,” Electronics, vol. 14, no. 19, Art. no. 3841, 2025, doi: 10.3390/electronics14193841.
- [24] S. Guo, Y. Zhong, Z. Feng, S. Kang, and J. Chen, “Towards Transparent 6G AI-RAN: A Survey on Explainable Deep Reinforcement Learning for Intelligent Network Slicing,” Journal of Information and Intelligence, vol. 4, no. 1, 2026.
- [25] B. Heller, R. Sherwood, and N. McKeown, “The Controller Placement Problem,” in Proc. 1st Workshop on Hot Topics in Software Defined Networks (HotSDN ’12), Helsinki, Finland, 2012, pp. 7–12.
- [26] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR, vol. 54, 2017, pp. 1273–1282.
- [27] 3GPP, “Architecture Enhancements for 5G System (5GS) to Support Network Data Analytics Services,” 3GPP TS 23.288, Release 18.
- [28] Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), Official Journal of the European Union, L series, Jul. 2024.
- [29] National Institute of Standards and Technology, “Module-Lattice-Based Key-Encapsulation Mechanism Standard,” FIPS 203, Aug. 2024.