IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 October 2024, accepted 15 November 2024, date of publication 25 November 2024, date of current version 9 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3503437

RESEARCH ARTICLE

Assurance in Advanced 5G Edge Continuum

FILIPPO BERTO[1], (Member, IEEE), CLAUDIO A. ARDAGNA[1], (Senior Member, IEEE), MASSIMO BANZI[2], AND MARCO ANISETTI[1], (Senior Member, IEEE)

[1]Department of Computer Science, University of Milan, 20122 Milan, Italy
[2]TIM S.p.A., 20123 Milan, Italy

Corresponding author: Filippo Berto (filippo.berto@unimi.it)

This work was supported in part by the Project Multilayered Urban Sustainability Action (MUSA), funded by the European Union–NextGenerationEU, under the National Recovery and Resilience Plan (NRRP) Mission Four Component Two Investment Line 1.5: Strengthening of Research Structures and Creation of Research and Development "Innovation Ecosystems," set up of "Territorial Leaders in Research and Development" under Grant CUP G43C22001370007 and Grant ECS00000037; and in part by the "One Health Action Hub: University Task Force for the Resilience of Territorial Ecosystems," under Grant PSR 2021–GSA–Linea 6.

ABSTRACT The implementation of distributed applications is more frequently achieved through the configuration of service-oriented workflows, which are then deployed within the Edge-Cloud Continuum. This approach facilitates the support of distributed processing pipelines. In this context, there is an increasing demand for solutions that can continuously guarantee the non-functional properties (e.g. security and performance) of such applications. This demand has been boosted by the advent of 5G, where the Continuum is empowered by the ability to involve pervasive 5G Edge Nodes (i.e., Multi-access Edge Computing) and powerful and reliable connectivity links (i.e., network slices). Although capable to support Non-Functional Properties (NFPs) like performance, the current 5G Core Network (CN) is not yet ready to host and execute service workflows nor capable of providing trustworthy guarantees on more advanced NFPs, such as, integrity, security, and robustness. This delay in achieving the full promise of the 5G CN architecture is having a great impact on the capillary diffusion of key technologies such as AI-empowered workflows, which require a fully trusted execution environment to comply with AI regulations such as the AI act. In this paper, we propose an extension of the 5G CN functionalities that supports service workflow deployment and a methodology for the continuous assessment of Non-Functional Properties, beyond simple performance, implemented in a lightweight assurance framework. Our assurance framework is integrated within a 5G simulator to provide a trustworthy 5G CN test-bed and experimentally evaluated in realistic scenarios.

INDEX TERMS 5G edge continuum, cloud computing security, edge cloud computing, edge cloud infrastructures, non-functional assurance, quality assurance.

I. INTRODUCTION

Modern data-intensive applications are increasingly composed of micro and nano services[1,2] whose operation strongly depends on the underlying infrastructure, platform, or containers. For instance, computationally demanding applications require big data infrastructures for their execution [1], while pervasive applications require connectivity and distributed computation infrastructures, such as the ones provided by Fog and 5G-Edge computing. Even lambda function-based microservices require an execution or orchestration environment whose peculiarities may have an impact on their performances or security [2]. Today, the edge-cloud continuum is becoming the preferred landing infrastructure for modern data-intensive applications composing and deploying microservices both in the cloud or in edge nodes, depending on their non-functional requirements, e.g., enabling data gathering and pre-processing close to where the data are generated. 5G Multi-access Edge Computing (MEC) is playing a relevant role in the definition of a 5G-enabled edge-cloud continuum infrastructures (5G continuum in the following), which is the focus of this paper. 5G continuum offers a diffuse access point to the network and addresses resource requests; it also offers novel advanced

The associate editor coordinating the review of this manuscript and approving it for publication was Hadi Tabatabaee Malazi.

[1]https://www.technavio.com/report/cloud-microservices-market-industry-analysis

[2]https://www.uber.com/en-IE/blog/microservice-architecture

infrastructure-level services to handle massive amounts of data and processing/networking capabilities. Along with the diffusion of modern data-intensive applications in the 5G continuum, especially in sensitive environments such as health, there is also an increasing demand to continuously guarantee their NFPs, such as security and privacy. However, the application of traditional assurance solutions [3], [4] in the 5G continuum is very challenging due to the peculiarities of the 5G infrastructure.[3,4,5] Assurance techniques have been traditionally used to continuously verify NFPs properties of distributed systems and applications [5], although the majority of studies have neglected the role of computing infrastructures linked to telecommunications, including the 5G continuum. The prevailing approaches to computing infrastructure assurance rely primarily on infrastructure-level monitoring solutions, such as Prometheus[6] and OpenTelemetry[7] and frequently focus on functional and performance aspects only [6], [7], [8], [9], [10], [11], often relying on human interpretation of the monitored events. Very few solutions address NFPs properties of infrastructures [12], [13], [14], [15]. Moreover, they heavily rely on strong assumptions regarding the level of control, for instance, the ability to deploying active agents (e.g., for testing purposes) in the system in production through the use of specific hooks. These approaches are not feasible when applied to the continuum infrastructure potentially spanning multiple organizations, where

i) multi-tenants actively execute their processes using the infrastructure capabilities, ii) the infrastructure components are executed with high privileges compared to user services, and iii) the behavior of the infrastructure is regulated by various standards and frameworks, some of which may be unknown or proprietary.

The assurance of telecommunications infrastructure is usually further constrained to Quality of Service (QoS) aspects pertaining to the performance of the network, including bandwidth availability, packet routing time guarantees, and radio coverage. This approach excludes crucial NFPs that necessitate comprehensive integration between the computing and networking components of the 5G continuum.

In this paper, we propose the first assurance methodology for the 5G continuum. We propose a novel lightweight assurance framework that complements existing application-level assurance in the verification of 5G-specific NFPs [16]. The proposed framework is based on the premise that the components of the 5G infrastructure are capable of passively exposing data regarding their operational status to the designated assurance agents. These measurements, being passively gathered, have a negligible impact on resource consumption of the operational environment. The measurements are aggregated through assurance-specific metrics combined to form contracts assessing whether a particular NFP holds (e.g., available computing capabilities, integrity status, confidentiality). For instance, a contract can specify that an "*up-time per day*" metric (computed based on measurements of all the components' heartbeat signals) must be greater than 99% to support the NFP *availability*.

The contribution of this paper is fourfold. First, we propose our scenario of a 5G-enabled edge-cloud continuum suitable for hosting service workflows requested to support advanced modern data-intensive applications (Section II). Second, we present a novel infrastructure-level assurance methodology decoupling infrastructure modeling from assurance evaluation (Section III). Third, we present the first attempt to apply such methodology on the 5G continuum having MEC nodes, with special emphasis on the current gaps to be addressed by the 5G standards and our proposal for addressing them (Section IV). Finally, we design a fully functional 5G simulator to experimentally show the utility, feasibility, and performance of our approach (Section V).

II. THE 5G CONTINUUM

5G MEC is increasingly playing a fundamental role in the edge-cloud continuum, providing a standardized high-reliability infrastructure for service delivery and execution. 5G MEC enabled computing nodes can offer both computing and storage facilities, as part of the edge-cloud nodes, and the advanced capabilities of a 5G CN, such as low latency, network slicing, and support for security and privacy. However, there are currently limitations to their functionality:

i) a full integration within the service deployment lifecycle [16] (e.g., deployment reachability and service/resource management on MEC) and ii) support for commodity services (e.g., permanent storage or volumes) fundamental for hosting third-party services.

Figure 1 illustrates our scenario for the 5G continuum. It considers two types of actors as follows.

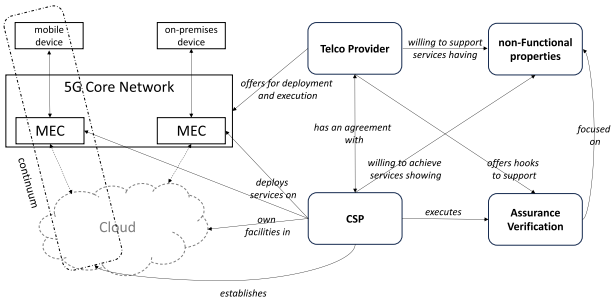

FIGURE 1. The 5G continuum scenario.

Telco provider: it supports service/workflow deployment and execution in 5G MEC nodes. It also offers i) infrastructure-level functionalities (e.g., authentication and

[3]https://www.technavio.com/report/operational-technology-security-market-analysis
[4]https://dzone.com/trendreports/microservices-and-containerization
[5]https://www.experian.com/blogs/data-breach/2022/12/08/the-2023-experian-data-breach-industry-forecast
[6]https://prometheus.io
[7]https://opentelemetry.io

authorization, routing of traffic to MEC, temporary storage) to support the business logic of the services and ii) hooks to carry out assurance verification at the 5G infrastructure level.

Cloud Service Provider (CSP): it deploys services/workflows guaranteeing specific NFPs through the peculiarities of 5G MEC. It also establishes a trusted continuum with its cloud facilities opening to new business opportunities based on continuum-native services. To this aim, a CSP needs an assurance verification approach that verifies NFPs offered by the 5G MEC, without requesting an invasive and resource-consuming solution such as the ones in [5] and [17]. Our *assurance verification* process addresses this need by inspecting the hooks offered by the Telco provider on its 5G infrastructures. The hooks permit lightweight inspections supporting specific NFPs (e.g., inspection of the access control configurations to guarantee property authentication), which are at the basis of our assurance verification.

The *5G CN* integrates with service deployment solutions to form the 5G continuum infrastructure. It is configured to have MEC nodes offering continuum-specific Telco services and hosting CSPs services.

We note that in the current business-ready solutions for the 5G continuum, such as AWS Wavelength, Google Distributed Cloud, and Microsoft Azure Private MEC, the 5G MEC nodes are considered part of the CSP perimeter. A Telco provider and a CSP have an agreement that permits the CSP to utilize machines on the Telco premises as MEC nodes. Unfortunately, their integration within the 5G CN is limited and strictly relies on the deployment and monitoring interfaces implemented by the Telco provider and their integration with the underlying infrastructure. Consequently, the CSP may possess incomplete insight and access to the system's current state.

In our scenario for the 5G continuum, the Telco operator hosts CSP services directly on its premises, allowing them to be combined with infrastructure-level services and fully integrated with 5G network functionalities. This includes dynamic on-request access to network resources such as network slices and integration of application-level authorizations with those of the Subscriber Identity Module (SIM) owner.

A. 5G CORE NETWORK

In this paper we refer to Release 17 of the 5G system [18], as this is the latest available revision of the 5G CN standard at the time of writing. Our methodology redefines part of the components related to QoS that have been introduced in this revision, providing a more general and elastic platform for non-functional assurance.

A comprehensive description of the 5G system, including its components and the integration with the MEC for automated deployment and management of Virtualised Network Functions (VNFs), is available in our previous paper [19]. Table 1 lists the components that are part of the 5G CN and summarizes their objectives. Figure 2 illustrates the configuration of the CN and its interconnections.

TABLE 1. 5G network services and components and their purpose.

Service	Description
Host	The computing platform that provides the resources required for the CNs to operate and manages their processes life-cycle
NSSF	Network slice selection service focusing on resource optimization
NEF	Access control to CN's internal data; and management of collaboration with other networks
NRF	NF's service discovery; and OAuth2 access token management
PCF	User policies management and enforcement, including charging, roaming and QoS
UDM	Service that manages user-related data from other processes, including user authentication and secure subscriber data management
AF	Integration with external applications; application-specific policy and session management; application traffic routing; and usage data for charging and billing
NSSAAF	Network slice selection based on service requirements and QoS enforcement
AUSF	User authentication; security key management for network traffic encryption and network resources access control
AMF	User registration and authentication; mobility management, including handovers; connection and security management; QoS policy enforcement
SMF	Session management service, UE IP address allocation and management, DHCP, NAS signaling, traffic steering
SCP	Signaling aggregation and routing; load balancing of network traffic; manages secure communication channels and handles overload events; compatibility layer for integration with legacy systems
MEC	Cloud-native Network Functions and Virtualised Network Functions deployment and management platform
UPF	Packet routing and forwarding; traffic inspection; QoS handling; session management; usage information collection for billing and charging purposes
UE	A device that connects to the core network through radio communication
RAN	Radio network infrastructure (antennas and controllers)
LADN	Private network or bridge to the public Internet

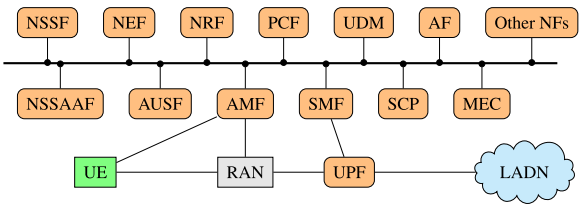

FIGURE 2. Scheme of the 5G CN architecture. The CN services, in orange, are often implemented as microservice components, interacting with the UEs and Radio Access Network (RAN). The Local Area Data Network (LADN) indicates the upstream network.

The 5G CN can intelligently be aware of the presence of relevant service applications in the Telco MEC, as it is defined by the European Telecommunications Standards Institute (ETSI) in [20], enabling the 5G User Plane Function (UPF) to route the requests to the MEC hosting the application. To do this, UPF communicates with the Session Management Function (SMF), which provides feedback on where to steer the packets. The MEC deployments advertise their services to the Policy Control Function (PCF), which updates its policies. These affect the user session and provides the routing rules to the SMF, which in turn configures the UPF to route the user data to the relevant network endpoint or to the LADN.

Handling NFPs in the CN and in the MEC's services is still an open challenge [21], [22]. The current literature provides only a limited overview and many aspects, such as authentication between MEC and CN components, remain quite unexplored [23]. Current solutions mainly target performance-related QoS aspects of the network, which are a limited subset of all NFPs. The 5G protocol can also address some more advanced NFPs such as security, by defining and applying security policies across the entire network using the Network Function PCF (5G System by 3GPP standards - working group SA3).

Our 5G System (5GS) architecture supports advanced NFPs assurance via an extended Network Data Analytics Function (NWDAF) carrying out assurance computation, security, and network data analysis (e.g., workload, QoS, anomalies). More specifically, the NWDAF network analysis notifications are used to support security-aware policy updates to monitor the network behavior and find weaknesses and unexpected behaviors resulting from security leakages. The NWDAF can be used to trigger changes at the policy level via the PCF. For instance, it can be possible to move users from a potentially compromised section or slice to a quarantine one to carry out further investigations.

In our scenario, we assume a Telco provider capable of configuring access policies and NWDAF, to address the NFPs requested by the CSP's services to be executed in the MEC.

B. FUNCTIONALITIES AND NON-FUNCTIONAL PROPERTIES

Current 5G MEC does not fully support all the functionalities and NFPs that are relevant for a CSP. For instance, functionalities to store data maintaining confidentiality and integrity at rest are usually not available at Telco MEC level. In this paper, we propose to enrich the 5GS and its MEC with de facto standard functionalities to support the most relevant NFPs, making the 5G nodes fully usable in the context of service provisioning on the continuum. Table 2 describes peculiar and available 5G functionalities of interest for an application to be deployed on MEC. It also shows the 5G CN services collaborating to achieve each functionality. Table 3 shows 5G functionalities that are not available yet, but can be implemented with limited effort to complement the actual functionalities. Such functionalities are likely to be shortly available to unleash the full potential of the 5G continuum and support more NFPs.

For instance, let us consider property *confidentiality of data at rest*. It requires services at MEC level that store data locally, thus preserving confidentiality. These services should be accessible by third-party services to store local data.

In this paper, we consider the following families of NFPs:

Access control (t_{acl}). Only authorized users can access resources, data, and services, in accordance with administrator-defined policies.

Automation (t_{auto}). The capacity of the system to avoid the need for human intervention. This encompasses activities such as configuring network setups, managing service deployments, and administering users, among others.

Availability (t_{avail}). The system's ability to reply to user requests continuously.

Confidentiality (t_{conf}). The protection of information from disclosure.

Integrity (t_{inte}). The system is capable of retaining the consistency of the data and of its internal state.

Locality (t_{loc}). The locality of data, computation, or message dissemination.

Performance (t_{perf}). The system's capability to meet users' demands within a given time frame.

Each specific Non-Functional Property p is complemented by specific attributes guiding our assurance methodology in their verification. More specifically, we define a full list of 25 fine-grained NFPs considering the set of 5G functionalities shown in Table 2 and Table 3. Table 4 shows an extract of NFPs.[8]

For each property, we present the property family, the relevant functionalities, and a brief description.

Example II.1. *Let us consider a scenario where a Telco provider offers its 5G continuum to a CSP for hosting its services.*

Let us assume a specific CSP service that requires storing services on the MEC, ensuring property p_{10} = Storage Confidentiality for the MEC. Property p_{10} belongs to property family t_{conf} and requires confidentiality of the data stored locally at the MEC.

Let us also assume that Telco MEC is empowered with a service ts_1 capable of storing binary data objects through a RESTful API (f_{33} in Table 3), accessible via token-based access control (f_{31} in Table 3). The 5G continuum can i) ensure CSP service requests via the MEC service ts_1 and ii) establish an assurance verification including the MEC service ts_1, with the scope of continuously assessing property p_{10}.

[8]The full list of NFPs is available at https://anonymous.4open.science/r/5G-Infra-assurance-tables/properties.md

TABLE 2. 5G core network functionalities currently available.

Id	Name	Description	Core network components
f_1	LADN-level connectivity	The CN can connect to the LADN	RAN, UPF, SMF
f_2	RAN-level connectivity	The CN can connect to the UEs through its RAN	RAN, UPF, SMF
f_3	UE-LADN routing	Network packet routing from UEs to LADN	RAN, UPF, SMF
f_4	UE-NF routing	Network packet routing from UEs to NFs	RAN, UPF, SMF
f_5	NF-LADN routing	Network packet routing from NFs to LADN	RAN, UPF, SMF
f_6	Massive IoT Networks	Connection with large number of IoT devices	Host, RAN
f_7	Low Latency Networks	Low latency networking functionalities on mobile devices, providing fast access to the edge nodes and to the cloud	Host, RAN
f_8	Network hop-based deployment selection	The deployment host can be selected on the number of network hops required to reach a given application	Host, MEC, NRF
f_9	RTT-based deployment selection	The deployment host can be selected on the latency reaching a given application	Host, MEC
f_{10}	Secure network channels	The CN provides secure channels for the deployed applications	Host, MEC, RAN
f_{11}	Private network slice management	5G offers network slices as private networks with SLO negotiation	RAN, UPF, SMF, NSSF
f_{12}	Bandwidth allocation	Network bandwidth allocation	Host, MEC
f_{13}	CPU allocation	CPU timeshares allocation	Host, MEC
f_{14}	RAM allocation	Memory allocation	Host, MEC
f_{15}	Hardware acceleration	Hardware accelerator allocation	Host, MEC
f_{16}	CNF-based application deployment	The CN can be the target of a CNF-based deployment	Host, NEF, SMF, MEC
f_{17}	VNF-based application deployment	The CN can be the target of a VNF-based deployment	Host, NEF, SMF, MEC
f_{18}	Policy management	Core network policy management (users, applications, network routing)	Host, NEF, MEC
f_{19}	Tenant management	The CN provides tenant management policies (i.e., limit one user per node)	Host, MEC
f_{20}	Certifications exposure	Platform certificates are exposed and available to the user	Host, MEC, RAN
f_{21}	Protocol compliance	The NFs are compliant with a given protocol (storage, network, deployment)	Host, MEC, RAN, NRF
f_{22}	UE-based Identification	UEs and their users can be uniquely identified in the network by the device IMEI and (e)SIM identifier	Host, RAN, UDM
f_{23}	Ephemeral storage	The CN provides an ephemeral storage service	Host
f_{24}	Permanent storage service	The CN provides an permanent storage service	Host, NEF
f_{25}	Storage level integrity	The CN storage provides integrity guarantees (hardware redundancy, data replication)	Host, MEC, RAN
f_{26}	Storage level encryption	The CN storage is encrypted	Host, MEC, RAN

III. 5G INFRASTRUCTURE MODELING

Figure 3 shows our infrastructure model composed of *functionalities* grouped to form *service interfaces*.

Service interfaces, possibly describing a particular protocol or specification (e.g., Open-Radio Access Network (O-RAN)), are implemented by one or more *components*, including hardware devices or software services, which collaborate to fulfill the interface requirements. Components are entities that have an internal dynamic *state* and a static *configuration*, which define their behavior. State and

TABLE 3. 5G core network functionalities likely available in the near future.

Id	Name	Description	Core network components
f_{27}	Private Cross-Network Slices	CNs' functionality of collaborating with each other to form inter-network and inter-ISP private network slices. This allows the creation of secure channels across a variety of networks	RAN, NEF
f_{28}	Contextual Deployment Policies	The controller that manages application deployments in the 5G network is aware of the context and can be extended to support more advanced requirements than resource constraints, including privacy and availability NFPs	Host, NEF
f_{29}	Geo-based deployment selection	The deployment host can be selected on its geographical position	Host, NEF
f_{30}	Geo-fenced applications	f_{28} can be extended with f_{29} to support user-defined geographical barriers, so that data and applications cannot accidentally be shared outside the selected region	Host, NEF, MEC
f_{31}	UE-based authentication	UE-based authentication service extending f_{22}	RAN, SMF, AMF, NEF
f_{32}	UE-Edge Resource Sharing	f_7 and f_{11} allow to link UE with resources on the edge network using private and low-latency connections	Host, NEF, RAN
f_{33}	Distributed permanent storage	The edge network can extend f_{24} acting as a fast-access intermediate layer in a distributed storage solution	Host, NEF
f_{34}	UE-unique encryption	Extends f_{23} and f_{24} with a UE-unique transparent encryption layer	Host, MEC, RAN, PCF
f_{35}	Storage Certified Ephemerality	f_{28} can help track the accessed resources providing f_{23} with certified ephemerality	Host, MEC, RAN, PCF
f_{36}	Secure deployment channels	The f_{11} and f_{27} functionalities of the 5G network can be used to provide both the users and the applications with secure channels applying a transparent encryption layer on any channel within and outgoing the network	Host, MEC, RAN, NEF
f_{37}	Deployment isolation	We can extend f_{28} to support isolated deployment strategies, where one user can allocate the full edge node	Host, MEC
f_{38}	Multi-ISP deployment (service roaming)	We extend f_{11} and f_{28} to allow continuous application deployment and migration across ISPs, moving the applications to another hoster once the SLOs cannot be respected	Host, MEC, NEF

TABLE 4. List of 5G core network Non-Functional Properties (extract).

Id	Name	family	Functionalities	Description
p_1	Network connection availability	t_{avail}	f_1, f_2, f_3, f_4, f_5	The infrastructure provides network connection capabilities to and from its applications and devices
p_4	Network latency performance	t_{perf}	f_7, f_9	Network infrastructure with minimal latency to a certain application
p_8	Network management automation	t_{auto}, t_{acl}	f_{18}, f_{19}	The network manages its configuration automatically
p_{10}	Storage confidentiality	t_{conf}, t_{acl}, t_{loc}	f_{26}, f_{31}, f_{34}	Can guarantee storage confidentiality
p_{13}	Storage integrity	t_{inte}	f_{25}	Can guarantee storage integrity
p_{22}	Geo-fenced deployments support	t_{conf}, t_{loc}	f_{29}, f_{30}	The infrastructure supports geo-fenced deployments, limiting access to applications and data from within a certain geographical region
p_{23}	UE-based authentication support	t_{auto}, t_{conf}	f_{31}	The infrastructure supports UE-based authentication for its applications deployments

configuration are exposed by the component to our assurance process via monitoring *endpoints* (see Section III-C). We note that endpoints can be natively available in the components (e.g., access to configuration files) or may

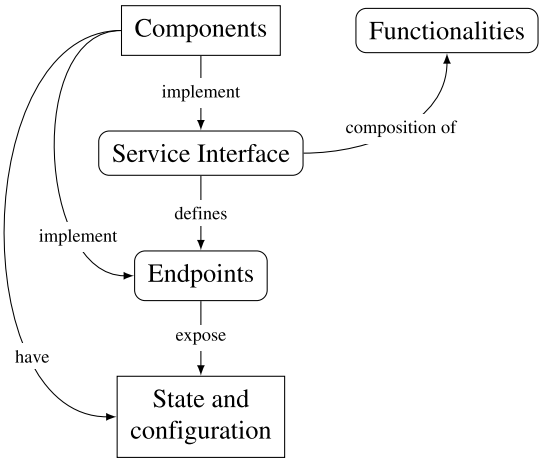

FIGURE 3. Our abstract model for the 5G continuum infrastructure.

need to be implemented by extending them (e.g., latency probing).

A. SERVICE FUNCTIONALITIES AND INTERFACE

The functionalities of a service are minimal executable actions with defined input and output constraints. Constraint can be both functional (e.g., protocol, formats) and non-functional (e.g., latency). A *service interface* $s \in \mathbf{S}$ is defined as a collection of functionalities $f \in \mathbf{F}$ such that $s = \{f_1, \ldots, f_n\} \subseteq \mathbf{F}$.

We note that different service interface s_i may have overlapping functionalities ($s_1 \cap s_2 \neq \varnothing$) or be a subset of another service interface ($s_1 \subseteq s_2$).

Example III.1. *Let us consider a service interface s_{S3} for the MEC service ts_1 in Example II.1 based on the Simple Storage Service (S3) object storage protocol.*[9] *The service interface s_{S3} defines the service protocol following the S3 specification and identifies a set of functionalities in this example $\{f_{S3GetObject}, f_{S3PutObject}, f_{S3GetBucketPolicy}\}$. The service protocol establishes how the functionalities can be accessed. For instance, S3 checks the ownership policy applied to the storage location (bucket) before providing access to its contents. At the level of each functionality, constraints on arguments and outputs can be specified. For instance, $f_{S3GetBucketPolicy}$ requires a valid bucket ID as input and returns a configuration entry containing a list of resources that the user can access and available actions as output.*

B. COMPONENTS

Components $\mathbf{C}$ are software or hardware items that, alone or collaborating with each other, implement one or more given service interfaces.

The implementation relationship $\mathbb{I}$ is a mapping from sets of components to sets of service interfaces as in $\mathbb{I}_{\wp}(\mathbf{C}) \mapsto \wp(\mathbf{S})$. We say that the set of components $\mathbf{c} \subseteq \mathbf{C}$ implements a certain service $s \in \mathbf{S}$ iff $s \in \mathbb{I}(\mathbf{c})$, which also means that $\mathbf{c}$ implements all the functionalities in $s = \{f_1, \ldots, f_n\}$. We may have multiple sets of components, possibly disjointed, implementing the same set of functionalities. In this case $\mathbf{c}_1, \mathbf{c}_2 \subseteq \mathbf{C}$ and $s \in \mathbf{S}$ we have that $s \subseteq \mathbb{I}(\mathbf{c}_1) \cap \mathbb{I}(\mathbf{c}_2)$ and $\mathbf{c}_1$ and $\mathbf{c}_2$ are said to be interchangeable for s. Similarly, a certain $\mathbf{c} \subseteq \mathbf{C}$ may implement multiple service interfaces at the same time, in this case for $s_1, s_2 \in \mathbf{S}$ we may have that $s_1 \cup s_2 \subseteq \mathbb{I}(\mathbf{c})$.

Example III.2. *Let us consider Example III.1, the service interface s_{S3} requires combination of three components $s_{S3} \subseteq \mathbb{I}(\{\circ_a, \circ_n, \circ_s\})$ as follows. A component $\circ_a$ implementing the RESTful API for functionalities requested by s_{S3}, a component $\circ_s$ providing allocable storage space, and a component $\circ_n$ providing the connectivity between them.*

In a practical scenario $\circ_a$ is a S3 compatible server, $\circ_s$ is a block storage (e.g., SSD), and $\circ_n$ is the network infrastructure of the system, such as the 5G network.

In order to correctly implement the service s_{S3}, all three components are required and sufficient, therefore $\mathbb{I}(\{\circ_a, \circ_n, \circ_s\}) = s_{S3}$.

C. STATES, CONFIGURATIONS, AND ENDPOINTS

Each component has an associated configuration and an internal state. The configuration is static and describes the component's context and its execution environment (e.g., bootstrap initialization, environment variables). The state of a component changes dynamically over time and includes information such as variable status and resource usage. Both configurations $\mathcal{C}$ and states $\mathcal{S}$ are mappings $K \mapsto V$ where K is a set of unique keywords and $V \subseteq (\mathbb{B} \cup \mathbb{I} \cup \mathbb{R} \cup \mathbb{S})$ is a set of boolean, integer, real or string values.

We assume that the 5G infrastructure components expose standardized monitoring interfaces (*endpoints* in the following) (e.g., Syslog, OpenTelemetry,[10] Prometheus[11]) having the scope to access internal state and configurations for audit purposes.

In this paper, we consider *endpoints* as the hooks of our assurance process enabling measurements gathering from the 5G infrastructure.

Example III.3. *Let us consider Example III.2 and component $\circ_a$ only. It has an internal state and a configuration that includes rules on network port bindings, storage paths, and data replication to name but a few. The state and configuration are exposed via specific endpoints. In particular, let us consider two endpoints for component $\circ_a$: i) $e_{health\ check}(time)$: health check to verify the component reachability; ii) $e_{acl}(time)$: the map of keys that have access to any buckets. These two endpoints allow us to measure aspects related to availability and authorization, which are relevant for the assurance verification of property $p_{data\ conf}$ as defined in Example II.1.*

[9] https://docs.aws.amazon.com/AmazonS3/latest/API

[10] https://opentelemetry.io

[11] https://prometheus.io

Although some endpoints are offered by the current 5G implementations available in the market, they are insufficient to cover all the properties of interest and, in some cases, offer heterogeneous and non-standard monitoring interfaces. This paper contributes to the 5G standard architecture by specifying endpoints that need to be adapted and new ones that need to be adopted to fully support the properties listed in Table 4.

IV. THE ASSURANCE VERIFICATION

The assurance infrastructure in this paper extends our previous solution in [24] being grounded on the abstract model of the 5G infrastructure in Section III and on the methodology depicted in Figure 4. Compared to the traditional assurance approaches it is capable of addressing 5G continuum-specific Non-Functional Property such as latency, privacy by data proximity, and embedded network-level multi-factor authentication, to name but a few.

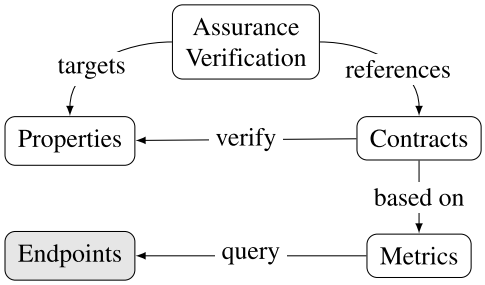

FIGURE 4. The methodology of our assurance verification.

It considers *contracts* defined in terms of *metrics* (e.g., a measure of service uptime aggregated per day) relevant for a specific NFP (e.g., uptime metrics in the case of NFP *availability*). Metrics are computed on behavioral measurements retrieved by the infrastructure *endpoints* (e.g., heartbeat for computing uptime metric).

The assurance verification succeeds in case a positive output is retrieved while evaluating the contract (e.g., uptime metric is greater than 98%).

A. METRICS

A *metric* is a function m that measures a specific aspect or behavior on a given endpoint. Metrics are defined over the states and configurations exposed by the monitoring endpoints.

Example IV.1. *Let us consider the monitoring endpoint* e_{acl} *from Example III.3 and a metric* $m_{\text{obj\_perm}}(time, object)$ *that measures the access permissions of a given object in a S3 service. We define the metric* $m_{\text{obj\_perm}}$ *using data from the endpoint* e_{acl} *at instant time as follows.*

$$\begin{aligned} &m_{obj\_perm}(time, object) := let \\ &\quad bucket = getBucket(object); \\ &\quad acl = e_{acl}(time); \\ &in\ getPermissions(acl, bucket) \end{aligned} \tag{1}$$

First, the metric calculates the name of the bucket that contains the object, then the Access Control List (ACL) is retrieved, and finally the object's permissions are extracted.

In the context of operational services, the monitoring endpoints are capable of exposing a diverse set of information. The application of metrics is essential for the identification of pertinent features and the extraction of supplementary information.

B. CONTRACTS

A contract $contract(time, \mathbf{c})$ is a boolean, time-dependent function that describes how to validate a given Non-Functional Property according to metric values obtained from relevant endpoints. It specifies a set $\mathbf{c} \subseteq \mathbf{C}$ of target components and the time instant *time* of the evaluation.

Example IV.2. *Let us consider the property data confidentiality* $p_{\text{data\_conf}}$ *and the metric* $m_{\text{obj\_perm}}$ *in Example IV.1. We define a simple contract that verifies whether the components* $\mathbf{c}$ *implementing the service* s_{S3} *have property* $p_{\text{data\_conf}}$ *for data object object at instant time, as shown in Equation 2.*

$$\begin{aligned} &p_{data\_conf}(time, \mathbf{c}, object, expected) = let \\ &\quad allowed = m_{obj\_perm}(time, \mathbf{c}, object); \\ &in\ allowed \leqslant expected \end{aligned} \tag{2}$$

The contract definition is parametric and checks whether $p_{\text{data\_conf}}$ *is true with different expected arguments. In addition, we do not set any limit on the evaluation time, expressing even cases where measurements are predicted in the future.*

C. THE ASSURANCE PROCESS

Our methodology in Figure 4 is implemented into a three-step assurance process providing continuous and collaborative verification of NFPs of infrastructure components as follows.

1) EVALUATION MAPPING

The first step constructs the dependency graph of the evaluation. This includes identifying the metrics and contracts that need to be evaluated, beginning with the list of targeted properties. To achieve this, contracts for each target property are located by reversing the V mapping (shown in Section III-C). Then, any potential duplicated evaluations are eliminated by enumerating each mentioned metric and its corresponding arguments.

2) MEASUREMENTS COLLECTION

The second step involves the collection of the measurements that are required for the evaluation. This includes requesting measurements data to the relevant monitoring endpoints and applying the necessary metrics functions.

3) CONTRACTS EVALUATION

The third step performs the actual property verification by computing the outcome of each contract using the previously

collected measurements. Each contract is evaluated producing a Boolean output.

We note that measurements and contract evaluations constitute evidence on whether a given Non-Functional Property has been verified, therefore they can support NFPs certification schemes [5], [24].

We also note that, given the complexity of the scenario and the lack of control over all the infrastructural components (e.g., third-party components), steps measurements collection and contract evaluation can potentially lead to unforeseen behaviors during their execution, such as metric implementation problems, environment failures, and logical mistakes in contracts. Our assurance process addresses such issues by identifying each issue and suggesting specific countermeasures (e.g., re-execution of measurements' collection).

V. EXPERIMENTAL EVALUATION

The scope of our experimental evaluation is twofold, i) to show the usability and utility of our approach within a real 5G assurance walkthrough (Section V-B), and ii) to verify the overhead requested by our assurance verification approach evaluating performance (Section V-C).

A. EXPERIMENTAL SETUP

We implemented the scenario depicted in Section II for experimentally evaluating our assurance framework, realizing a fully functional 5G Core Network simulator extended with MEC functionalities and monitoring capabilities. Our 5G simulator is based on the Open Networking Foundation's Aether,[12] which is deployed in a Docker environment.

The simulator consists of the 5GS core network microservices; SD-Core, which is based on the Free5GC[13] project; a virtualized O-RAN compliant gNodeB (gNB), SD-RAN, which simulates the radio part of the mobile network; and Aether ROC, which coordinates the network components and automates their configuration and resource management. The components of our 5G Core Network simulator are extended with state and configuration endpoints based on Prometheus and the corresponding measurements are exposed for our assurance verification process to be executed. We also extend our 5G CN simulator with the functionalities requested to support the properties presented in Table 4. We consider default configurations for all the 5G components of our simulator, to create a baseline for our assurance verification. We set the QoS level for UEs connection to the LADN to 200 Mb/s. All services are running on the same physical host in a controlled environment, preventing the introduction of unintended network latency and errors.

B. 5G ASSURANCE WALKTHROUGH

For conciseness, in the following, we focus on properties p_1 = "Network connection availability", p_8 = "Network Management Automation", and p_{22} = "Geo-fenced deployments support" in Table 4. For these properties, we detail how our assurance verification is executed and evaluated.

1) NETWORK CONNECTION AVAILABILITY

The property p_1 defines the capacity of the infrastructure to provide network features to the devices and services that are connected to it. This property is crucial for the proper functioning of the 5G infrastructure, and thus its continuous verification is of paramount importance.

The positive verification ensures that it delivers networking capabilities to its components accurately, connecting them smoothly and giving UEs access to the LADN.

The network feature is flexible, allowing us to set the QoS that the user and CSPs have agreed on for the connection. More in detail, the property is defined as $p_1(t, band_{ul}, band_{dl}, latency, avail)$, where $band_{ul}$ and $band_{dl}$ represent the upload and download bandwidth, respectively, $latc$ the time latency, and $avail$ the overall availability expressed as percentage over a given period of time.

a: ENDPOINTS AND METRICS

As the property p_1 encompasses several functionalities $(f_1, f_2, f_3, f_4, f_5)$ that correspond to various connectivity levels, it is necessary to evaluate multiple metrics. We consider metric $net\_avail(t, from, to)$, which is specific to each feature and permits to examine if a communication is viable between the communicating parties *from* and *to*. Additionally, it provides measurements of the latency (res.latency), upload (res.upload) and download bandwidth (res.downlink), and availability during the specific time instant t (res.avail). The 5GS interface already implements some required endpoints, such as the uplink and downlink speed. Our simulator implements additional endpoints: e_1 to monitor the host for collecting performance measures on the network interfaces, e_2 to repeatedly check for availability and latency between two components, *from* and *to*.

b: CONTRACT

p_1 is verified using a parametric contract that takes into consideration the values of the metrics and compares them to expected values (i.e., $latency$, $band_{ul}$, $band_{dl}$, $avail$). Therefore, the contract takes the form of Equation 3.

$$
\begin{aligned}
& p_1(t, band_{ul}, band_{dl}, latency, avail) := \\
& \quad \bigwedge n1, n2 \in \mathbf{UE} \cup \mathbf{NF} \cup \{DN\} \\
& \textit{let} \\
& \quad res = net\_avail(t, n_1, n_2) \\
& \textit{in} \\
& \quad \text{res.uplink} \geqslant band_{ul} \wedge \text{res.downlink} \geqslant band_{dl} \wedge \\
& \quad \text{res.latency} < latency \wedge \text{res.avail} \geqslant avail
\end{aligned} \tag{3}
$$

c: ASSURANCE RESULTS

The measurements obtained by metric net_avail in the given period of time t, expected values used, and the results

[12] https://opennetworking.org/aether/
[13] free5gc.org/

obtained from the p_1 contract evaluation are reported in Table 5.

TABLE 5. Network availability contract results.

Measurements	Expected Values	Result
Uplink	180 Mb/s	199 Mb/s
Downlink	100 Mb/s	199 Mb/s
Latency	10 ms	4 ms
Availability	99.9%	100%

The connection between the NF and the LADN is uncapped, resulting in a measured bandwidth of upwards of 40 Gb/s and a latency lower than 1 ms. Contrary, the connection between the UE and the UPF is limited by the capabilities of the gNB simulator, managing bandwidth levels up to 400 Mb/s and latency under 2 ms. Finally, measuring the connection between the UE and the LADN, with the traffic being routed by the UPF, the available bandwidth ranges between 180 Mb/s and 230 Mb/s, with an average value of 199 Mb/s, very close to the target QoS level of 200 Mb/s, while the average latency is 4 ms. The availability of each connection has been consistently 100% for all the testing intervals. All bounds required in the contract have been satisfied, therefore the property p_1 holds.

2) NETWORK MANAGEMENT AUTOMATION

The network management automation property p_8 means that the system can automatically administer its configurations, conforming to the agreed QoS by the CSP, ensuring their correctness. This property is associated with the functionalities f_{18} and f_{19}, which refer to the ability to apply user and resource management policies to the CN and to manage tenant requests at a granular level, respectively. To verify this property, we need to ensure that the system utilizes services' configurations that conform to the QoS expected values. For this experimental evaluation, we use two virtual UEs with different policy configurations.

a: ENDPOINTS AND METRICS

To verify this property, we use the metric $net\_conf(t, ue)$ that extracts the expected network configuration for each UE, and a metric $net\_state(t, ue)$ that measures the current network configuration of the system. Endpoints supporting this metric are not available in the current 5G implementations: we implemented them in our simulator by extending the PCF and NF 5G Core Network components, exposing their internal status stored in the CN's database, and exposing the UPF's firewall and routing configurations. More specifically, we consider the expected values shown in Table 6.

b: CONTRACT

p_8 is verified using a parametric contract that takes as input t, the instant in time; and ue, the targeted UE. The contract uses metrics $net\_conf(t, ue)$ and $net\_state(t, ue)$ to obtain the expected and the actual state of the UPF and compares them, returning a positive output *iff* all requirements are met. It checks whether hard expected values were correctly applied by the PCF to the UPF configuration, as shown in Equation 4.

$$
\begin{aligned}
&p_8(\text{t, ue}) := \mathit{let} \\
&\quad \text{conf} = \text{net\_conf(t, ue)} \\
&\quad \text{state} = \text{net\_state(t, ue)} \\
&\quad c_1 = \text{conf.locality} == \text{state.locality} \\
&c_2 = \\
&\quad \text{conf.qos.downlink} == \text{state.qos.downlink} \wedge \\
&\quad \text{conf.qos.uplink} == \text{state.qos.uplink} \\
&\quad c_3 = \text{conf.mec} == \text{state.mec} \\
&\mathit{in} \bigwedge \{c \mid c \in c_1, c_2, c_3\}
\end{aligned} \tag{4}
$$

TABLE 6. Network management automation expected values and results.

Aspect	Expected Values	Result
ue_1 locality	can reach LADN	can reach LADN
ue_2 locality	cannot reach LADN	**can reach LADN**
ue_1 QoS levels	downlink = 100 Mb/s, uplink = 100 Mb/s	downlink = 100 Mb/s, uplink = 100 Mb/s
ue_2 QoS levels	downlink = 100 Mb/s, uplink = 100 Mb/s	downlink = 100 Mb/s, uplink = 100 Mb/s
Integration with MEC	ue_1 uses locally hosted services	ue_1 uses MEC-hosted services
Integration with MEC	ue_2 uses locally hosted services	ue_2 **uses cloud-hosted services**

c: ASSURANCE RESULTS

In this verification, we identified two misconfigurations that invalidated the property p_8: first, the ue_2 is not allowed to reach the LADN, but a wrongly configured firewall allowed the UE to reach the public Internet; second, the same UE is supposed to use a MEC-hosted version of a service, but the misconfigured UPF forwards the traffic to the LADN, thus reaching a cloud-hosted deployment, as highlighted in Table 6. Contrary, the QoS settings are applied correctly. The property p_8 does not hold.

3) GEO-FENCED DEPLOYMENTS SUPPORT

The support for geo-fenced deployments property p_{22} is guaranteed if the system is configured to forward the user traffic to a local instance of a target application. This prevents it from sending data outside the 5G private network or to a specific computing node. This property is of interest from a privacy and performance standpoint and is based on the functionalities f_{29} and f_{30}. The initial property refers to the capability of deploying applications in a particular geographic location, while the latter refers to the ability to establish geographic boundaries for both data and application deployments, restricting their access to and from outside their designated environment. More in detail, the property

is defined as $p_{22}(t, app, position)$, where *position* is the geographical boundaries that the application should respect, and *app* is an identifier for the target application. For this experimental evaluation, we deployed two applications in the continuum, within MEC and cloud environments, and verified their behavior using a virtualized UE.

a: ENDPOINTS AND METRICS

The verification of p_{22} requires a metric net_routing(t, app, $target$) that provides the current network configuration of the edge network, in particular, how the user traffic is routed between the application *app* and the respective *target*. We also require a metric app_deployment(t, app) that describes where the application *app* is physically deployed. The metrics implementations are based on endpoints we added to our 5G simulator to export network policies of the edge network, and include UPF's capabilities extension.

b: CONTRACT

To validate the computation locality of the edge network, we defined the contract for p_{22} by combining the previously declared metrics, as shown in Equation 5.

$$\begin{aligned} & p_{22}(\text{t, app, position}) := \textit{let} \\ & \quad \text{deployment} = \text{app\_deployment(t, app)} \\ & \quad \text{edge\_dn} = \text{net\_routing(t, app, dn)} \\ & \textit{in} \\ & \quad \text{deployment} \in \text{position} \wedge \text{edge\_dn} == \text{false} \end{aligned} \tag{5}$$

c: ASSURANCE RESULTS

We verified the validity of the property on two sample applications. The first was correctly configured, routing the traffic from its users to the local instance, deployed on the MEC, and isolated from the LADN, thus confirming the property p_{22}. The second was correctly deployed on the MEC, but the network configuration routed the users' traffic to the LADN, forwarding it to the cloud instance, and invalidating the geolocality property. The property p_{22} does not hold.

We note that in this walkthrough two out of three properties do not hold. We also note that, to support the walkthrough, we extended the standard endpoints available in the current 5G Core Network.

C. PERFORMANCE EVALUATION

In order to evaluate our assurance solution we identified three experimental scenarios: i) we measure the overhead introduced by our monitoring solution comparing the original version of the CN services with one where we introduced transparent monitoring capabilities; ii) we compare the resource usage of our solution under several stress loads, with several number of connected User Equipment and network usage; iii) at last, we assess the efficacy of our assurance agent by quantifying the time required for the assessment of one of the contracts delineated in the preceding section.

1) MONITORING OVERHEAD

We first experimentally evaluated the computation overhead introduced by the need to collect measurements for executing our assurance approach in terms of CPU and RAM usage. A series of stress tests were conducted to evaluate the resource utilization of the 5G simulator with and without the execution of measurement collection operations and with and without network traffic. The traffic is generated through an instance of *iPerf3*[14] in each UE sending data to an external server hosted on the same physical machine with a constant throughput of 200 Mb/s. This gives us a rough estimate of the computation overhead introduced by each UE using the network. Where enabled, the metrics data is collected at a fixed rate, once every five seconds. The CPU and memory usage is averaged in a one-minute simulation, while the NetIO values are measured at the end of the simulation. The test has been executed in a virtual machine with 4 virtual cores and 16 GB of memory using the standard Free5GC Docker compose deployment configuration.

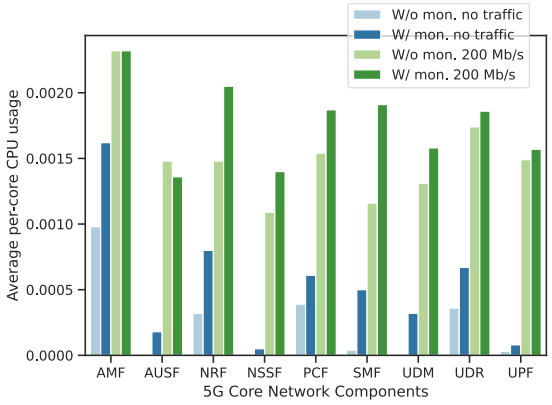

FIGURE 5. CPU usage comparison.

Figure 5 shows the computation overhead in terms of CPU usage with and without measurements collection (w/ mon., w/o mon) in the two scenarios, with and without traffic (no traffic, 200 Mb/s). The total per-core percentage of CPU usage grows from 0.00213 to 0.00483 in the scenario without traffic and from 0.01361 to 0.01592 in the scenario with 200 Mb/s of traffic. The most affected component is the SMF, whose CPU usage grows from 0.00004 to 0.00050 in the scenario without traffic and from 0.00116 to 0.00191 when we introduce 200 Mb/s of traffic, while the trend is similar with the other components. We note that the additional overhead has a constant part, linked to the computation necessary to expose the metrics, and a part proportional to the computation load.

In consideration of the memory overhead, the monitoring solution introduces a slight overhead, between 4 MiB and 11 MiB, that is discernible in the scenario devoid of traffic, yet too minute to be accurately quantified in the scenario with

[14]iPerf: https://iperf.fr/

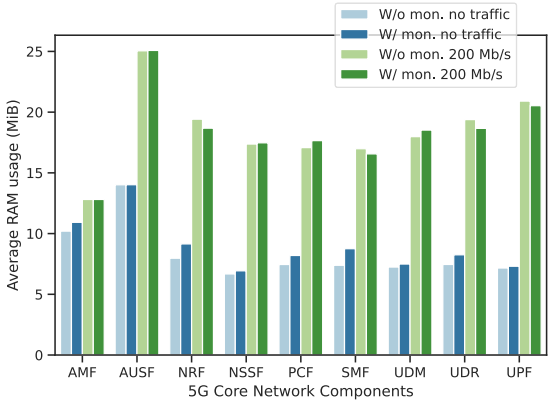

FIGURE 6. Memory usage comparison.

traffic, as it is obscured by the surrounding noise, as shown in Figure 6.

The growth is in line with the additional libraries being included and loaded in the components, and the overhead of updating the metric counters in each application, with low variance during their operation. We also measured the memory usage after one hour of simulation, not showing substantial changes.

The network input/output levels are, as expected, higher due to the monitoring traffic. In the simulation with traffic we measured and additional 20 KB to 150 KB of data being transmitted.

2) MONITORING OVERHEAD SCALING

A second fundamental aspect of the monitoring solution is its scalability, which enables it to adapt to changes in the number of connected UEs and network traffic. Accordingly, five supplementary experimental scenarios were devised, in which the resource utilization of the CN components was gauged. These comprised a baseline scenario without connected UEs, two scenarios with three UEs each producing 5 or 10 Mb/s, and two scenarios with five UEs each producing 5 or 10 Mb/s. The tests have been executed in the same environment as the ones in Section V-C1. The resource usage is measured once every minute through Prometheus using the Docker API and averaged in a 15 minute time frame.

The CPU usage in all scenarios is significantly lower than 0.20% in each NFs, with negligible changes among the different experimental scenarios, given the very limited CPU usage, except for the AMF. The AMF's measurements show an increase of usage as the number of UEs increases, from 0.05% in the baseline, to 0.15% in both 3 UEs scenarios, and 0.20% and 0.25% respectively in the 5 UEs scenarios, with 5 and 10 Mb/s. We note that the CPU usage is very low in all cases compared to the resource usage linked to the RAN simulator.

We also note that the UPF usage is not particularly affected by the traffic because in the Free5Gc core network the network traffic is managed by a specialized kernel module, thus moving most of the load outside of the user space.

Similarly to CPU usage, the memory usage is generally consistent across the five scenarios, with little variation during the simulation and measurements between 10 to 20 MiB.

3) ASSURANCE ENGINE PERFORMANCE

The processing time required by our assurance agent to verify a contract was benchmarked in the experimental scenario described earlier. We considered a contract that is representative of the common type of checks we would implement in a real network, focusing on the availability of a network slice's bandwidth.

To do so, we implement a contract that executes the following three steps: i) collect the last 1000 measurements of a NF metric using a Prometheus source; ii) calculate the average value; iii) compare it with a fixed threshold, as shown in equation 6;. The data collection is repeated for each iteration of the benchmark a total of 100 times without caching any of the previous results.

$$
\begin{aligned}
& c_{test}(\mathrm{t}) = let \\
& \text{time\_range} = \{0, \ldots, -999\} \\
& \text{evidence} = \{ \\
& \quad \text{net\_state}(\mathrm{t}_i, \text{ue}).\text{uplink} \mid i \in \text{time\_range} \\
& \} \\
& \text{average} = \frac{\sum_{e \in \text{evidence}} e}{1000} \\
& in \quad \text{average} > 200\text{mb/s}
\end{aligned} \tag{6}
$$

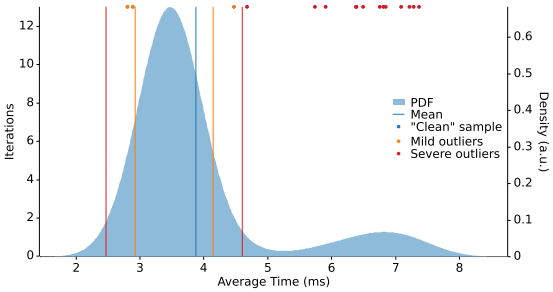

FIGURE 7. Probability density function of the assurance process execution time.

Figure 7 shows the density function of the measured time of execution for the assurance process. We can see that the average metric collection time is 3.872 ms with a standard deviation of 1.089 μs and a median of 3.452 ms. The results consistently show a response time of under 5 ms. Profiling the execution, we measured that the computational part of the contract evaluation averages 37 μs, demonstrating that network latency is still the most significant component of the total time measured.

VI. CONCLUSION

In this paper, we presented a novel lightweight assurance methodology for 5G Core network. It enables to use 5G in the framework of advanced 5G-empowered Cloud

Continuum where workflow of services requiring non-functional properties can be executed.

The technique we proposed extends the core network with standard and widespread monitoring techniques, allowing ease of integration with other monitoring and verification solutions.

In order to demonstrate our assurance framework we realized a fully functional 5G simulator enriched with functionalities that are nowadays missing in the commercial 5G architectures allowing to support service workflow execution and assurance computation.

We demonstrated the utility and performance of the proposed solution in real scenarios under several configurations and application scenarios.

REFERENCES

[1] M. Anisetti, C. A. Ardagna, and F. Berto, "An assurance process for big data trustworthiness," *Future Gener. Comput. Syst.*, vol. 146, pp. 34–46, Sep. 2023.

[2] A. Mathew, V. Andrikopoulos, and F. J. Blaauw, "Exploring the cost and performance benefits of AWS step functions using a data processing pipeline," in *Proc. 14th IEEE/ACM Int. Conf. Utility Cloud Comput.*, Dec. 2021, pp. 1–10.

[3] M. Anisetti, C. A. Ardagna, N. Bena, and R. Bondaruc, "Towards an assurance framework for edge and IoT systems," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Sep. 2021, pp. 41–43.

[4] M. A. Javed, F. U. Muram, H. Hansson, S. Punnekkat, and H. Thane, "Towards dynamic safety assurance for Industry 4.0," *J. Syst. Archit.*, vol. 114, Mar. 2021, Art. no. 101914.

[5] M. Anisetti, C. A. Ardagna, and N. Bena, "Continuous certification of non-functional properties across system changes," in *Proc. Int. Conf. Service-Oriented Comput.*, 2023, pp. 3–18.

[6] L. Heng, G. Yin, and X. Zhao, "Energy aware cloud-edge service placement approaches in the Internet of Things communications," *Int. J. Commun. Syst.*, vol. 35, no. 1, Jan. 2022, Art. no. e4899.

[7] N. A. Angel, D. Ravindran, P. M. D. R. Vincent, K. Srinivasan, and Y.-C. Hu, "Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies," *Sensors*, vol. 22, no. 1, p. 196, Dec. 2021.

[8] X. He, Z. Tu, X. Xu, and Z. Wang, "Programming framework and infrastructure for self-adaptation and optimized evolution method for microservice systems in cloud–edge environments," *Future Gener. Comput. Syst.*, vol. 118, pp. 263–281, May 2021.

[9] H. Wang, T. Liu, B. Kim, C.-W. Lin, S. Shiraishi, J. Xie, and Z. Han, "Architectural design alternatives based on cloud/edge/fog computing for connected vehicles," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2349–2377, 4th Quart., 2020.

[10] R. Makhlouf, "Cloudy transaction costs: A dive into cloud computing economics," *J. Cloud Comput.*, vol. 9, no. 1, p. 1, Jan. 2020.

[11] S. Parikh, D. Dave, R. Patel, and N. Doshi, "Security and privacy issues in cloud, fog and edge computing," *Proc. Comput. Sci.*, vol. 160, pp. 734–739, May 2019.

[12] P. Ranaweera, A. Jurcut, and M. Liyanage, "MEC-enabled 5G use cases: A survey on security vulnerabilities and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–37, Oct. 2021.

[13] Q.-V. Pham, F. Fang, V. N. Ha, Md. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[14] N. Hassan, K. A. Yau, and C. Wu, "Edge computing in 5G: A review," *IEEE Access*, vol. 7, pp. 127276–127289, 2019.

[15] M. Anisetti, C. A. Ardagna, and E. Damiani, "Container-level security certification of services," in *Business System Management and Engineering: From Open Issues To Applications*. Cham, Switzerland: Springer, 2012, pp. 93–108.

[16] M. Anisetti, F. Berto, and R. Bondaruc, "QoS-aware deployment of service compositions in 5G-empowered edge-cloud continuum," in *Proc. IEEE 16th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2023, pp. 471–478.

[17] M. Anisetti, C. A. Ardagna, E. Damiani, and F. Gaudenzi, "A semi-automatic and trustworthy scheme for continuous cloud service certification," *IEEE Trans. Services Comput.*, vol. 13, no. 1, pp. 30–43, Jan. 2020.

[18] *System Architecture for the 5G System (5GS) (3GPP TS 23.501 Version 17.5.0 Release 17)*, Standard ETSI TS 123 501 V17.5.0, 2022.

[19] M. Anisetti, F. Berto, and M. Banzi, "Orchestration of data-intensive pipeline in 5G-enabled edge continuum," in *Proc. IEEE World Congr. Services (SERVICES)*, Jul. 2022, pp. 2–10.

[20] *Multi-access Edge Computing (MEC); Framework and Reference Architecture*, Standard ETSI GS MEC 003 V3.1.1, 2022.

[21] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 196–248, 1st Quart., 2020.

[22] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location privacy in mobile edge clouds: A chaff-based approach," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2625–2636, Nov. 2017.

[23] G. Sahu and S. S. Pawar, "Security Challenges in 5G Network," in *Software Defined Networking for Ad Hoc Networks*. Cham, Switzerland: Springer, 2022, pp. 75–94.

[24] M. Anisetti, C. A. Ardagna, F. Berto, and E. Damiani, "A security certification scheme for information-centric networks," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 2397–2408, Sep. 2022.

FILIPPO BERTO (Member, IEEE) is currently a Postdoctoral Researcher with the University of Milan, Italy. In particular, he works in the field of security assurance, 5G networks, and edge-cloud computing, focusing on infrastructures, networks, and services certification techniques. His research interests include cybersecurity, edge computing, and distributed systems.

CLAUDIO A. ARDAGNA (Senior Member, IEEE) has been a Visiting Researcher with BUTP, Khalifa University, GMU. He is currently a Full Professor with the University of Milan, the Director of the CINI National Laboratory on Big Data, and the Co-Founder of Moon Cloud s.r.l. He has published more than 140 contributions in international journals, conference/workshop proceedings, and chapters in international books. His research interests include cloud-edge security and assurance, and data science.

MASSIMO BANZI has been a Senior Standardization Manager with TIM S.p.A., since 2011, dealing with topics mainly related to cloud computing, big data analytics, and the impact on support systems of the convergence of network and virtualization, within several SDOs among which TMForum, ETSI, NGMN, and ITU-T. Previously, he covered the role of Responsible for the Open Source Strategies Project and Configuration Management Competence Center in the Telecom Italia IT Department. He has also been a Contract Professor with the Free University of Bolzano.

MARCO ANISETTI (Senior Member, IEEE) is currently a Full Professor with the University of Milan. His research interests include computational intelligence and its application to the design and evaluation of complex systems. He has been investigating innovative solutions in the area of cloud security assurance evaluation. In this area, he defined a new scheme for continuous and incremental cloud security certification, based on distributed assurance evaluation architecture.

Open Access funding provided by 'Università degli Studi di Milano' within the CRUI CARE Agreement