A Unified Application Profile Model for Microservices-Based Applications in the Cloud-Edge Computing Continuum

Abdelhak Kadouma*
University of Vaasa, Fingletek
Vaasa, Finland
x3307846@student.uwasa.fi

Ibrahim Afolabi
Fingletek
Espoo, Finland
ibrahim.afolabi@fingletek.com

Amit Shukla
University of Vaasa
Vaasa, Finland
amit.shukla@uwasa.fi

Miloud Bagaa
Université du Québec à Trois-Rivières
Québec, Canada
Miloud.Bagaa@uqtr.ca

Adlen Ksentini
Eurecom
Biot, France
ksentini@eurecom.fr

Mohammed Elmusrati
University of Vaasa
Vaasa, Finland
moel@uwasa.fi

***Abstract*—The dynamic and distributed nature of the Cloud Edge Computing Continuum (CECC) necessitates innovative approaches to application profiling for microservices-based applications. Existing application profiling approaches typically address only isolated aspects, such as user behavior, system performance, or data interactions. Moreover, many rely on static representations that cannot adapt to real-time changes. These problems make it harder to manage resources properly, meet SLA requirements, and handle the full lifecycle of modern distributed applications. This paper presents a Unified Application Profile Model (APM) Template, which integrates microservices, application consumers, and data sources to create a comprehensive and adaptable model tailored for CECC environments. The proposed APM provides a formal structure, defined using OWL (Web Ontology Language), ensuring precise representation of application components and their interrelationships. This formalization aids in understanding and managing the unique challenges of CECC, including dynamic resource allocation, latency sensitivity, and scalability. Key findings demonstrate how this APM surpasses traditional profiling approaches by unifying critical application dimensions and addressing CECC-specific demands. The template sets the foundation for future research into practical implementations, including integration strategies and dynamic adaptability, positioning the APM as a significant step forward in enabling efficient and sustainable application lifecycle management for CECC environments.**

***Index Terms*—cloud edge computing continuum, microservices, application profile, ontology**

I. Introduction

The rapid evolution of distributed computing has led to the emergence of the Cloud Edge Computing Continuum (CECC), which integrates cloud and edge infrastructures to facilitate scalable, low-latency, and resource-efficient application deployment [1]. CECC environments encompass centralized cloud data centers and geographically distributed edge nodes, designed to address the growing demand for latency-sensitive applications by positioning computational tasks closer to data sources and end-users [10], [13]. Microservices-based architectures have gained prominence within CECC due to their modularity, adaptability, and scalability [14]. Unlike monolithic applications, microservices decompose software into small, loosely coupled services that communicate over a network, facilitating independent deployment and fault isolation [8]. This modular approach enables critical services to be deployed at the edge for real-time processing, while computationally intensive tasks are offloaded to cloud resources [9]. However, managing microservices in distributed environments introduces challenges in coordinating interactions, optimizing dependencies, and ensuring performance across varying network conditions and resource constraints [11]. Profiling microservices-based applications in CECC requires a comprehensive approach that captures the interplay between microservices, application consumers, and data sources [2]. Traditional profiling methods, such as static system-level monitoring and descriptor-based profiling, primarily focus on individual components without considering the dynamic interplay of distributed applications [12]. These methods often rely on predefined resource utilization thresholds, which fail to adapt to real-time fluctuations in user demand, network latency variations, or edge resource availability. This paper introduces a Unified Application Profile Model (APM) that provides a structured, semantic representation of microservices-based applications in CECC environments. By leveraging formal modeling techniques such as the Web Ontology Language (OWL), the proposed APM ensures semantic clarity, modularity, and adaptability across diverse CECC scenarios [7]. The key contributions of this research include:

- **A Unified Profiling Model:** A formalized APM that integrates microservices, user interactions, and data dependencies to enhance profiling accuracy in CECC envi-

ronments.

- **Ontology-Based Representation:** A semantic approach using OWL to define application components and their interdependencies, ensuring modularity and interoperability.
- **Dynamic Resource Adaptation:** A profiling method that supports real-time deployment adjustments, enabling efficient resource allocation and SLA compliance.

In this work, we position the APM as the system of record for application semantics across the cloud–edge continuum. The APM does not implement control logic; rather, it exposes static descriptors (e.g., topology, data flows, SLA terms) and binds dynamic fields (e.g., observed load, latency, location hints) that are updated by monitoring. A lifecycle management (LCM) component in the orchestration stack consumes the APM to evaluate policies and enact scaling, placement, or migration. This separation of concerns—profile (APM) vs. control (LCM)—is central to our design and to the portability of the model across different CECC platforms.

This paper is structured as follows: Section II reviews related work in microservices profiling, cloud-edge computing, and semantic modeling. Section III presents the Unified Application Profile Model, detailing its structure and ontology representation. Section IV demonstrates the model's application in a real-world case study. Section V discusses the evaluation of the profiling model and outlines future research directions. Finally, Section VI concludes the paper.

II. Related works

The exploration of application profiling within the Cloud-Edge Computing Continuum (CECC) and microservices architecture has garnered significant attention in recent research. This area addresses the challenges of managing applications in dynamic, distributed, and heterogeneous environments, where the interplay between user behavior, system performance, and data management is critical. Profiling methodologies have evolved from narrowly focused approaches to more comprehensive models that integrate multiple perspectives.

Early profiling methods primarily focused on understanding user behavior and preferences. For instance, Shaman et al. [4] proposed a model for user profiling based on network metadata at the application level, leveraging a gradient-boosting machine learning algorithm to improve user identification accuracy. This approach provided deep insights into user behavior and preferences by analyzing network-level data, achieving moderate accuracy (74%) in user identification. However, it was limited by its reliance on small datasets and specific environments, reducing its generalizability. Similarly, Bailey et al. [3] introduced a model aggregating user data from heterogeneous sources using Semantic Web technologies, enabling location and application-agnostic profiles but raising scalability and privacy concerns.

As application complexity grew, profiling models began incorporating system-centric and data-centric elements. Ferrari et al. [2] developed the Linked Data JSON SPARQL Application Profile (JSAP), a resilient model designed for applications exploiting the SPARQL Event Processing Architecture (SEPA). This model enhanced the development of distributed and context-aware applications through a structured, JSON-LD-compliant file format, enabling high interoperability and modularity. López-Acosta et al. [5] proposed a Metadata Application Profile (MAP) for structuring large-scale data in Social Network Analysis (SNA). This model utilized Elasticsearch (ES) and Python to address challenges related to data quality, volume, and structure, enabling the creation of co-authorship networks and identification of core authors through centrality metrics. MAP demonstrated the utility of data profiling for scientific research by improving metadata elements for effective indexing and search.

The advent of CECC introduced new challenges requiring models that could dynamically adapt to distributed and latency-sensitive environments. Jha et al. [13] developed IoTSim-Edge, a simulation framework for modeling IoT and edge computing environments, evaluating resource management and latency but lacking predictive adaptation capabilities. Korontanis et al. [10] evaluated modeling languages like TOSCA and CAMEL for deployment and runtime profiling in cloud-edge environments, revealing shortcomings in hybrid runtime flexibility and geolocation-aware deployment.

Despite these advancements, existing profiling models remain fragmented, as summarized in Table I and Table II. Current methodologies either focus on isolated profiling dimensions (user, system, or data) or rely on static profiling techniques that fail to capture real-time changes in CECC environments. Furthermore, many models lack formal semantic representations, such as ontologies, which ensure interoperability across heterogeneous infrastructures.

TABLE I
Comparison of Some Profiling Approaches (Part 1)

Ref.	**Focus Methodology**	**Profiling Dimensions**	**Formal Representation**
Shaman et al. [4]	User profiling using network metadata	Primarily user-centric	Not formally defined
Ferrari et al. [2] (JSAP)	Semantic interoperability with JSON SPARQL application profile	System and data-centric	JSON-LD, semantic approach
López-Acosta et al. [5]	Metadata profiling for scientific data in SNA	Data-centric	Informal metadata model
Proposed Unified Approach	**Unified profiling for microservices-based applications in CECC**	**Integrates user, system & data**	**OWL-based formal ontology**

To address these limitations, this research introduces the Application Profile Model (APM)—a unified profiling approach that integrates user, system, and data-centric profiling through an OWL-based semantic representation. Unlike prior methodologies, APM supports real-time adaptability, enables predictive analytics, and facilitates intelligent microservice placement within CECC. By incorporating semantic represen-

TABLE II
COMPARISON OF SOME PROFILING APPROACHES (PART 2)

Ref.	Adaptability	CECC-specific Features	Strengths and Limitations
Shaman et al. [4]	Static profiling	Not explicitly for CECC	+ Provides insights into user behavior - Lacks adaptability and a multi-dimensional perspective
Ferrari et al. [2] (JSAP)	Moderate adaptability	Designed for IoT/SEPA environments, not fully CECC	+ Ensures good semantic consistency - Lacks real-time adaptability and scalability
López-Acosta et al. [5]	Limited adaptability	Not designed for CECC	+ Improves metadata quality - Scalability issues; focuses only on structured data
Proposed Unified Approach	**Real-time and dynamic**	**Explicitly designed for Cloud-Edge Computing**	+ Integrates user, system, and data profiling - Potential scalability challenges

tations, dynamic resource allocation, and SLA-aware profiling, APM ensures a holistic, scalable, and interoperable approach for managing microservices-based CECC applications.

III. UNIFIED APPLICATION PROFILE MODEL

The APM is a novel, unified model designed to address the complexities of managing microservices-based applications in the Cloud-Edge Computing Continuum (CECC). As CECC environments are inherently dynamic, distributed, and heterogeneous, the APM provides a holistic and adaptable representation of application components and their associated attributes. It integrates four interconnected layers—Application Metadata, Application Consumers, Microservices-Data, and Service Level Agreements (SLAs)—to facilitate lifecycle management, including resource allocation, deployment optimization, and SLA compliance. The primary objective of this APM is to overcome the limitations of existing profiling approaches, which often rely on static representations or isolated perspectives. By offering a dynamic and comprehensive model, the APM enables real-time adaptability and informed decision-making, ensuring optimal application performance and resource efficiency in CECC environments.

The APM is formally represented using OWL for semantic clarity and interoperability. Figure 1 provides a visual representation of the APM, organizing its key components and their associated attributes. The APM is structured into four core layers working in synergy to provide comprehensive profiling, enabling real-time adaptability, resource optimization, and compliance with performance guarantees.

A. Metadata Layer: Application Identification and Lifecycle Management

The Metadata Layer forms the foundation by encapsulating essential identification and lifecycle attributes. It ensures applications are uniquely identifiable across distributed infrastructures, facilitating version control, traceability, and change tracking. This layer includes a globally unique identifier, human-readable name, versioning system, and timestamps for creation and modification times. These elements ensure seamless application monitoring and management, preventing inconsistencies in distributed environments.

B. Application Consumers Layer: User Interaction and Demand Profiling

The Application Consumers Layer captures interaction dynamics between users and the application, providing insights critical to performance optimization. It models consumer roles (administrators, developers, end-users), geographical distribution data for latency-sensitive deployments, and interaction patterns including access frequency, preferred methods (mobile app, API), and session durations. This layer enables dynamic resource adjustment based on user demand, facilitating intelligent workload distribution and improved Quality of Service (QoS). Anticipating high-traffic periods allows proactive resource scaling, minimizing latency and optimizing user experience.

C. Microservices-Data Layer: Application Architecture and Data Sources

The Microservices-Data Layer defines the structural and functional aspects of microservices-based applications, capturing interdependencies and associated data sources. Each microservice is profiled based on resource parameters (CPU, memory, bandwidth, storage) for optimized allocation across cloud and edge nodes. The APM tracks traffic characteristics, data flow types, and communication rates to identify potential bottlenecks in inter-service interactions. Geographical deployment considerations enable optimal placement within cloud, edge, or far-edge environments based on cost, availability, and latency constraints. Performance monitoring captures latency, throughput, and error rate metrics for real-time decision-making. Data sources are analyzed for access patterns, classification (hot vs. cold data), and regulatory constraints such as GDPR compliance. These attributes enable intelligent data storage and retrieval mechanisms, ensuring data locality optimization while maintaining regulatory adherence.

D. SLAs Profiling Layer: Service Performance and Compliance Management

The SLAs Profiling Layer formalizes performance guarantees and compliance parameters, ensuring applications operate within predefined Service Level Agreements. This layer establishes core attributes such as availability requirements, latency thresholds, and acceptable error rates, defining operational constraints. These parameters automate SLA monitoring, allowing dynamic resource reallocation in response to workload

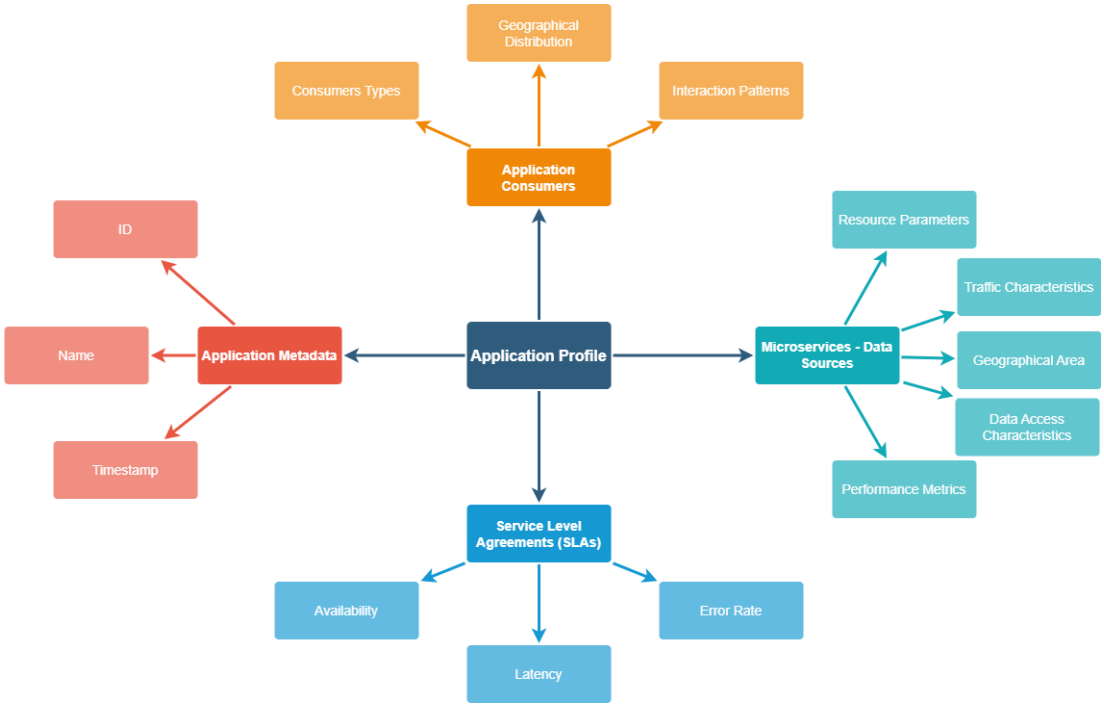

Fig. 1. Holistic Structure of the Proposed Application Profile Model (APM).

fluctuations. By incorporating predictive analytics, this layer enables proactive resource provisioning, ensuring latency-sensitive applications maintain uninterrupted service delivery. SLA compliance tracking helps mitigate risks associated with service degradation, improving overall system resilience and fault tolerance.

To provide a formalized and machine-readable representation, the APM is structured using the OWL which ensures modularity, extensibility, and interoperability across diverse CECC environments. Figure 2 is a high-level OWL representation of the APM, outlining its core entities and relationships. By structuring the profiling model using OWL, the APM provides a semantically rich and machine-readable framework that enables seamless integration across heterogeneous platforms.

A fundamental aspect of effective application profiling within CECC is the ability to systematically collect, structure, and refine information over time. To achieve this, the APM classifies profiling data into two primary categories: Static Data and Dynamic Data.

- **Static Data**, provided by the end user, refers to deployment-time attributes that remain constant throughout the application lifecycle. These attributes are predefined at deployment and serve as the foundation for system configuration, monitoring, and management. Static data includes core application metadata (Application ID, Name, Version) that uniquely identify the application within the system. It also encompasses Service Level Agreements (SLAs) defining essential operational constraints, including availability guarantees, latency thresholds, and error rate limits. Additionally, static user roles are specified, detailing access privileges and expected interactions. In the context of microservices, this category includes microservice identifiers, dependencies, and initial resource allocation parameters, ensuring each microservice is properly configured with CPU, memory, and networking requirements from the outset.
- **Dynamic Data** is collected at runtime and continuously evolves as the application operates. Unlike static data, which remains unchanged unless explicitly modified, dynamic profiling enables the system to adjust to real-time conditions and predict future workloads. This category is further divided into predictable and unpredictable dynamic data:
  - **Predictable Dynamic Data** consists of metrics that follow recognizable trends, such as expected traffic loads, user access frequencies, and seasonal workload fluctuations. These attributes can be forecasted using historical trends and predictive analytics, allowing the system to proactively adjust resources in anticipation of demand surges.
  - **Unpredictable Dynamic Data** includes high-variance metrics subject to sudden fluctuations, such as unexpected traffic spikes, anomalies in system behavior, and variations in user engagement due to external factors. These unpredictable changes necessitate real-time monitoring and adaptive decision-making to ensure system stability and prevent performance degradation.

The APM's OWL-based structure incorporates both static and dynamic profiling information, enabling a holistic view

```
@prefix app: <http://example.org/app#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# Metadata Layer
<METADATA_ID> a app:ApplicationMetadata ;
    app:hasID "METADATA_ID"^^xsd:string ;
    app:hasName "APPLICATION_NAME"^^xsd:string ;
    app:hasTimestamp "APPLICATION_TIMESTAMP"^^xsd:dateTime .

# Application Consumers Layer
<CONSUMER_ID> a app:ApplicationConsumer ;
    app:hasType "CONSUMER_TYPE"^^xsd:string ;
    app:hasGeographicalDistribution "GEO_LOCATION"^^xsd:string ;
    app:hasInteractionPattern "INTERACTION_PATTERN"^^xsd:string .

# Microservices-Data Layer
<MICROSERVICE_ID> a app:Microservice ;
    app:hasResourceRequirements <RESOURCE_REQUIREMENTS_ID> ;
    app:hasTrafficCharacteristics <TRAFFIC_CHARACTERISTICS_ID> ;
    app:hasGeographicalArea <GEOGRAPHICAL_AREA_ID> ;
    app:hasDataAccessCharacteristics <DATA_ACCESS_CHARACTERISTICS_ID>
;   app:hasPerformanceMetrics <PERFORMANCE_METRICS_ID> .

# SLAs Layer
<SLA_ID> a app:SLA ;
    app:hasAvailability "AVAILABILITY_REQUIREMENT"^^xsd:float ;
    app:hasLatency "LATENCY_REQUIREMENT"^^xsd:float ;
    app:hasErrorRate "ERROR_RATE_REQUIREMENT"^^xsd:float
```

Fig. 2. High-level structure of the application profile

of application performance. By structuring static attributes alongside dynamically collected metrics, the model facilitates adaptive and predictive resource management. This approach ensures microservices dynamically scale based on changing workloads, user access patterns are continuously refined, and SLA compliance is actively monitored.

The separation of static and dynamic data maintains profiling accuracy during deployment while providing real-time adaptability, allowing CECC applications to evolve based on operational requirements and unexpected developments.

IV. Case Study: Real-Time Surveillance System

To demonstrate the Application Profile Model (APM), we examine its application in a real-time surveillance system that integrates IoT devices, video analytics, and multi-service architectures. The system supports live streaming and video-on-demand capabilities, with IoT cameras deployed on ground-based infrastructures and UAVs collecting environmental and video data. Key requirements include seamless network connectivity, high computational efficiency, scalability, sufficient storage, high availability, low-latency responses, secure communications, and data redundancy.

The APM structures this surveillance system into four distinct profiling layers: Metadata, Application Consumers, Microservices-Data, and SLA Profiling, each capturing specific application attributes to enable dynamic adaptability and efficient resource allocation.

The Metadata Layer provides fundamental identification attributes with a globally unique application ID, human-readable name, timestamp for version control, and descriptive summary. This structure supports lifecycle management by enabling traceability and consistency in deployments. Figure 3 shows the OWL representation of these attributes.

```
# ---------------------- Metadata Layer ----------------------
<METADATA_ID> a app:ApplicationMetadata ;
    app:hasID "SurveillanceSystem123"^^xsd:string ;
    app:hasName "Surveillance System"^^xsd:string ;
    app:hasTimestamp "2024-01-01T00:00:00Z"^^xsd:dateTime ;
    app:hasDescription "Real-time surveillance system for monitoring video feeds and analytics."^^xsd:string
```

Fig. 3. Surveillance System Application Metadata Layer

```
# ---------------------- Application Consumers Layer ----------------------

<CONSUMER_CUSTOMER_ID> a app:ApplicationConsumer ;
    app:hasType "Customer"^^xsd:string ;
    app:hasGeographicalDistribution <GEO_DISTRIBUTION_CUSTOMER> ;
    app:hasInteractionPattern <INTERACTION_PATTERN_CUSTOMER> .

<CONSUMER_ADMIN_ID> a app:ApplicationConsumer ;
    app:hasType "Admin"^^xsd:string ;
    app:hasGeographicalDistribution <GEO_DISTRIBUTION_ADMIN> ;
    app:hasInteractionPattern <INTERACTION_PATTERN_ADMIN> .


# ---------------------- Geographical Distribution ----------------------

<GEO_DISTRIBUTION_CUSTOMER> a app:GeographicalDistribution ;
    app:hasLocation "Urban Regions"^^xsd:string ;
    app:hasAccessPoints "Web, Mobile"^^xsd:string .

<GEO_DISTRIBUTION_ADMIN> a app:GeographicalDistribution ;
    app:hasLocation "Global, multi-region access"^^xsd:string ;
    app:hasAccessPoints "Web, API, IoT Dashboard"^^xsd:string .


# ---------------------- Interaction Patterns ----------------------

<INTERACTION_PATTERN_CUSTOMER> a app:InteractionPattern ;
    app:hasPeakInteractionTime "Evening Hours"^^xsd:string ;
    app:hasInteractionFrequency "Moderate"^^xsd:string .

<INTERACTION_PATTERN_ADMIN> a app:InteractionPattern ;
    app:hasPeakInteractionTime "Throughout the day"^^xsd:string ;
    app:hasInteractionFrequency "High"^^xsd:string .
```

Fig. 4. Surveillance System Application Consumer Layer

The Application Consumers Layer in the surveillance use case models user interactions by capturing consumer roles, geographical distribution, and behavioral patterns. Figure 4 illustrates differentiation between customers (accessing video feeds in urban regions) and administrators (overseeing system configurations globally). The layer details interaction patterns, including peak usage periods and interaction frequency, and models access points ,identifying whether users interact via web applications, mobile platforms, to refine network, optimize computational resource allocation and latency.

The Microservices-Data Layer captures the structural and operational attributes of the surveillance system's microservices to enable efficient resource allocation across cloud and edge nodes. As shown in Figure 5, the profiling model describes the Video Analytics microservice, a GPU-accelerated video processor requiring 8 vCPUs, 32GiB RAM, and 2 NVIDIA GPUs to handle 1000 requests per second in continuous streaming. It is deployed at edge nodes in surveillance regions to optimize latency-sensitive processing and reduce unnecessary cloud transfers. The data access profile involves frequent writes for real-time ingestion and low-latency reads for immediate retrieval, supporting high-throughput analytics. Performance metrics ensure operation within defined limits, maintaining latency below 200ms, very high throughput, and an error rate under 1

Fig. 5. Surveillance System Application Microservice-Data Layer - Video Analytics Service

Fig. 6. Surveillance System Application SLAs Layer

The SLA Profiling Layer ensures performance commitments through availability, latency, and error rate constraints. Figure 6 illustrates SLA attributes embedded within the application profile for automated compliance tracking and real-time policy enforcement, maintaining service uptime and sub-second response times.

A. Evaluation of the Profiling Model

The APM will excel at comprehensive microservice characterization by systematically modeling resource requirements, performance limitations, and service interdependencies. As depicted in Figure 5, The Video Analytics microservice in the surveillance system will demonstrate this capability through formal specification of computational needs, traffic patterns, data usage, and deployment configuration.

The Application Consumers Profiling, illustrated in Figure 4, will formally model user roles, geographical distribution, and interaction patterns to support latency-aware deployments and workload optimization.

SLA Profiling, depicted in Figure 6, will ensure the surveillance system maintains strict performance commitments. This will support automated SLA monitoring, enabling real-time policy enforcement and proactive resource adjustments to ensure continuous compliance.

This structured representation of microservices, user interactions, data access patterns, and SLA guarantees will ensure that the surveillance system or any CECC application can dynamically adapt to evolving operational conditions, ensuring efficient resource utilization, and compliance with operational constraints.

B. Lifecycle Management Enhancements

The proposed APM will enhance lifecycle management in CECC environments by facilitating resource allocation, deployment optimization, and SLA monitoring. Below, we will discuss how the APM will support these key processes:

- **Resource Allocation and Scaling**: The APM supports adaptive resource allocation by profiling microservices and their specific resource requirements. During high-demand scenarios, such as live-streaming or intensive video analytics, the Video Analytics microservice requires substantial computational resources for real-time object detection and anomaly recognition. In contrast, non-essential services, such as Frontend interfaces, can be scaled down during low-activity periods to improve overall resource efficiency.
- **Deployment Optimization**: Geographical distribution profiling within the Application Consumers Layer will inform intelligent deployment strategies based on performance and cost trade-offs. Latency-sensitive microservices (IoTDeviceCamera, Video Analytics) will be deployed at edge nodes within surveillance regions to minimize network delays, while compute-intensive workloads will be offloaded to the cloud for scalable infrastructure. This hybrid deployment approach will optimize data locality, ensuring that processing occurs as close to the source as possible. Figure 4 visualizes this strategic deployment, showing how user behavior and interaction patterns will guide microservice placement for maximum efficiency.
- **SLA Monitoring and Compliance**: The SLAs profiling will enable continuous monitoring of key performance metrics, ensuring that the surveillance system meets strict reliability and responsiveness requirements. For example, if network congestion threatens to increase latency, the APM will initiate load balancing by redirecting video streams to less congested edge nodes or deploying additional computing resources.

In the surveillance scenario, the APM will encode topology, stream rates, accelerator needs, and SLA targets. A monitoring collector updates the APM's dynamic fields (e.g., ingress rate, GPU utilization, end-to-end latency). When the LCM detects that predicted or observed latency approaches the SLA bound, it uses the APM to identify compliant candidate sites (e.g., edge nodes with GPU availability and locality constraints) and enacts a placement or scaling action. Thus, APM enables dynamic management; the LCM performs it.

V. Discussion and Future Work

This study introduces a Unified Application Profile Model (APM) that addresses the unique challenges of profiling

microservices-based applications in the Cloud Edge Computing Continuum (CECC). By integrating application metadata, microservices and data sources, application consumers, and service-level agreements into a comprehensive framework, the APM enables dynamic, adaptable, and semantically structured application management in distributed environments.

The prototype case study of a real-time surveillance system demonstrated the APM's effectiveness in adapting to dynamic workloads, optimizing hybrid cloud-edge deployments, and maintaining performance guarantees under varying conditions. The model facilitated real-time resource allocation, latency-sensitive microservice placement, and proactive SLA monitoring, validating its practical applicability in CECC environments.

As this work introduces a conceptual model, future efforts will focus on its empirical validation. We are aware of the need to assess the APM's real-world performance, and benchmarking plans are already being defined. These evaluations will consider metrics such as profiling overhead, adaptation latency, SLA violation rates, and semantic reasoning efficiency. Simulation environments like CECC testbeds will be used to emulate scalable microservices scenarios and validate the APM's dynamic adaptability under stress conditions.

Several directions for enhancement have emerged. Most notably, the integration of AI/ML techniques is expected to significantly boost the model's intelligence and automation capabilities. Predictive learning models can analyze historical and real-time data collected from integrated monitoring tools (e.g., Prometheus, Kepler) to anticipate changes in workload, detect anomalies, and enable proactive resource adjustments. In particular, Generative AI models could be used to assist end-users in automatically generating initial application profiles based on simple high-level descriptions, which are then refined and validated semantically within the APM framework.

Additionally, energy-efficient profiling methods should be explored to support sustainability goals in large-scale deployments. Tackling scalability challenges—such as ontology processing time and profile update propagation—will ensure the model's applicability across diverse and complex CECC infrastructures. Further enhancements may include security and privacy-aware extensions to protect sensitive profiling data, as well as automated tools to streamline profile creation, validation, and synchronization with deployment platforms.

Altogether, this research lays a foundation for a flexible and intelligent profiling ecosystem. By clearly defining the profiling structure, integration strategies, and evaluation roadmap, this work sets the stage for the APM to evolve from a conceptual model into a practical, standards-aligned solution for managing next-generation distributed applications in the CECC.

VI. Conclusion

The Unified APM represents a significant step forward in efficient, scalable, and sustainable application profiling for CECC environments. Its holistic design, adaptability, and semantic structuring make it a foundational model for modern distributed systems, with the potential to evolve into a standard framework for application profiling in the CECC paradigm. By addressing both current challenges and emerging trends, this research contributes to the advancement of profiling methodologies, laying the groundwork for optimizing next-generation distributed applications in increasingly complex cloud-edge infrastructures.

Acknowledgment

This work was partially supported by the European Union's Horizon Research and Innovation program under AC3 project and grant agreement No 101093129.

References

[1] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, Jan. 2018.

[2] A. Ferrari, E. Riforgiato, and L. Roffia, "JSON SPARQL Application Profile for Linked Data," in *2022 31st Conference of Open Innovations Association (FRUCT)*, Helsinki, Finland, pp. 1-10, Apr. 2022.

[3] C. Bailey, U. Kruschwitz, and M. Gardner, "The unified profile: Applying user profile data within intelligent environments," in *2008 IET 4th International Conference on Intelligent Environments*, Seattle, WA, USA, pp. 1-7, Jul. 2008.

[4] F. Shaman, B. Ghita, N. Clarke, and A. Alruban, "User Profiling Based on Application-Level Using Network Metadata," in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, Barcelos, Portugal, pp. 1-6, Jun. 2019.

[5] A. López-Acosta, A. García-Hernández, S. Vázquez-Reyes, and A. Mauricio-González, "A Metadata Application Profile to Structure a Scientific Database for Social Network Analysis (SNA)," in *2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pp. 208-214, 2020.

[6] R. Chandra, S. Tiwari, S. Agarwal, and N. Singh, "Semantic web-based diagnosis and treatment of vector-borne diseases using SWRL rules," *Knowledge-Based Systems*, vol. 274, pp. 110645, 2023.

[7] H. Guermah, T. Fissaa, H. Hafiddi, M. Nassar, and A. Kriouile, "An Ontology Oriented Architecture for Context Aware Services Adaptation," *arXiv*, vol. 1404, no. 3280, pp. 1–10, Apr. 2014. Available: http://arxiv.org/abs/1404.3280.

[8] A. Diepenbrock, F. Rademacher, and S. Sachweh, "An Ontology-based Approach for Domain-driven Design of Microservice Architectures," in *Proceedings of Gesellschaft für Informatik*, Bonn, 2017, pp. 1777–1791.

[9] G. Morais and M. Adda, "OMSAC - Ontology of Microservices Architecture Concepts," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Nov. 2020, pp. 293–301. doi: 10.1109/IEMCON51383.2020.9284892.

[10] I. Korontanis, A. Makris, and K. Tserpes, "A Survey on Modeling Languages for Applications Hosted on Cloud-Edge Computing Environments," *Applied Sciences*, vol. 14, no. 6, Art. no. 6, Jan. 2024, doi: 10.3390/app14062311.

[11] F. Al-Doghman, N. Moustafa, I. Khalil, N. Sohrabi, Z. Tari, and A. Y. Zomaya, "AI-Enabled Secure Microservices in Edge Computing: Opportunities and Challenges," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1485–1504, Mar. 2023.

[12] A. Shenoy, J. Hiner, S. Lysecky, R. Lysecky, and A. Gordon-Ross, "Evaluation of Dynamic Profiling Methodologies for Optimization of Sensor Networks," *IEEE Embedded Systems Letters*, vol. 2, no. 1, pp. 10–13, Mar. 2010.

[13] D. N. Jha et al., "IoTSim-Edge: A Simulation Framework for Modeling the Behaviour of IoT and Edge Computing Environments," *arXiv*, Oct. 07, 2019. doi: 10.48550/arXiv.1910.03026.

[14] N. Dragoni et al., "Microservices: Yesterday, Today, and Tomorrow," in *Present and Ulterior Software Engineering*, M. Mazzara and B. Meyer, Eds., Cham: Springer International Publishing, 2017, pp. 195–216.