Towards a Unified Compute and Communication Infrastructure for Application and Network Management

Luis M. Contreras
Telefonica
luismiguel.contrerasmurillo@telefonica.com

Roland Schott
Deutsche Telekom
roland.schott@telekom.de

Sabine Randriamasy
Nokia Bell Labs
sabine.randriamasy@nokia-bell-labs.com

Richard Yang
Yale University
yry@cs.yale.edu

Jordi Ros-Giralt
Qualcomm Europe, Inc.
jros@qti.qualcomm.com

Abstract

Since the 2002 IAB workshop, network topologies have become significantly more complex. Cloud-based applications, edge-cloud applications, and the virtualization of network service functions require the selection of transmission paths that reflect both compute and communication capabilities to accommodate a new breadth of applications with stringent Quality of Experience requirements (e.g., artificial intelligence, autonomous driving, and augmented reality). During the operational lifetime of a service, network management entities intervene at different levels of infrastructure that cover one or more domains. The infrastructure information they consume, therefore, needs to be simple, sufficient, and unambiguous. This paper explores potential standardization tracks for the definition and exposure of richer and more consistent infrastructure data spanning both compute and communication domains. It also addresses the promises and challenges of combining and abstracting this information.

Keywords Data consistency for richer observability · Automation, orchestration, and autonomy · Communication · Compute

1 Introduction

While network management standards are constantly maturing, they do not cover the tight integration of data centers with the network. In a few words, the network boundary is expanding by incorporating resources from these data centers. Transmission paths today involve both compute and communication facilities, and their management and optimization should reflect this. This is an unavoidable trend due to the virtualization of network functions (e.g., RFC7665 [13]). Service placement and service selection require knowledge of both compute and communication resources to ensure (1) the efficient usage of the infrastructure, including cost optimization such as low energy costs, and (2) optimized user experiences.

Standardized network management and *de facto* standard cloud management systems are progressing separately, and there is no common way to define and expose related decision spaces. This issue becomes essential when multiple domains are involved, as the exposed information must be comparable to allow consistent calculations and selection. Additionally, efficiency and network confidentiality call for simple and focused decision space definitions.

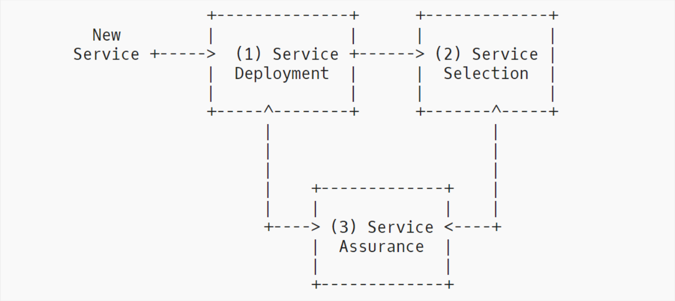
This position paper explores potential standardization tracks for the definition and exposure of compute and communication infrastructure (CCI) to optimize virtualized network services and user applications. It discusses the promises and challenges of creating an unambiguous and sufficient abstraction of CCI and their exposure through standard interfaces.

2 Problem Statement

With the emergence of a new generation of applications with stringent performance requirements —e.g., distributed AI training and inference, driverless vehicles, and virtual/augmented reality— the need for advanced solutions that can model and manage compute and communication resources has become increasingly important. Today's networks connect compute resources deployed across a continuum, ranging from data centers (cloud computing) to the edge (edge computing). While the same architecture principles apply across this continuum, in this paper we focus on the deployment of services at the edge, involving the cooperation of different actors —i.e., network operators, service providers and applications— in a heterogeneous environment.

At the edge, operators have a considerable advantage over public cloud providers because they own and manage extensive WAN and cellular networks (e.g., 5G) worldwide. Leveraging their communication infrastructure, operators are deploying compute nodes at the edge of their networks to provide computing services that are close to users. Thanks to this user proximity, operators and service providers can enable the following key benefits in comparison to the distant cloud infrastructure [2]: (1) Reduced application latency; (2) Increased communication bandwidth; (3) Increased application reliability; (4) Improved privacy and security; (5) Improved personalization; and (6) Reduced energy consumption.

To unlock the true potential of this edge computing infrastructure, however, important capabilities are lacking in today's protocol interfaces. In the next section, we discuss these limitations by studying the different stages involved in the lifecycle of a service.



```
                   +--------------+      +-------------+
       New         |              |      |             |
       Service +-----> (1) Service +------> (2) Service |
                   |  Deployment  |      |  Selection  |
                   |              |      |             |
                   +-----^--------+      +-------^-----+
                         |                       |
                         |                       |
                         |                       |
                         |    +-------------+    |
                         |    |             |    |
                         +----> (3) Service <----+
                              |  Assurance  |
                              |             |
                              +-------------+
```

Figure 1: Service life cycle

2.1 Service Life Cycle

As shown in Figure 1, a service is deployed on a distributed compute and communication infrastructure (CCI) through a two-stage process, starting with the service deployment stage followed by the service selection stage:

- **(1) Service deployment.** This stage is carried out by the service provider and involves the deployment of a new service (e.g., a distributed AI training/inference, an XR/AR service) on the CCI. The service provider needs to properly size the amount of compute and communication resources assigned to this new service to meet the expected user demand. The decision on where the service is deployed and how many resources are requested from the infrastructure depends on the levels of Quality of Experience (QoE) that the provider wants to guarantee to the users of the service. To make a proper deployment decision, the provider must have visibility on the resources available within the infrastructure, including compute (e.g., CPU, GPU, memory and storage capacity) and communication (e.g., link bandwidth and latency) resources. For instance, to run a Large Language Model (LLM) with 175 billion parameters, a total aggregated memory of 350GB and 5 GPUs may be needed [1]. The service provider needs an interface to query the infrastructure, extract the available compute and communication resources, and decide which subset of resources are needed to run the service.

- **(2) Service selection**. This stage is initiated by the user, through a client application that connects to the deployed service. There are two main actions that must be performed in the service selection stage: (2.a) *compute node selection* and (2.b) *path selection*. In the compute node selection step, as the service is generally replicated in N locations (e.g., by leveraging a microservice architecture), the application must decide which of the service replicas it connects to. This decision depends on the compute properties (e.g., CPU/GPU availability) of the compute nodes running the service replicas. On the other hand, in the path selection decision, the application must decide which path it chooses to connect to the service. This decision depends on the communication properties (e.g., bandwidth and latency) of the available paths. Similar to the service deployment case, the application needs an interface to query the infrastructure and extract the available compute and communication resources, with the goal to make informed node and path selection decisions. Note that in some scenarios, the network or service provider can make node and path selection decisions in lieu of the application. It is also important to note that, ideally, the node and path selection decisions should be jointly optimized, since in general the best end-to-end performance is achieved by jointly taking into account both factors. In some cases, however, such decisions may be owned by different players. For instance, in some network environments, the path selection may be decided by the network operator, wheres the compute node selection may be decided by the application or the service provider. Even in these cases, it is crucial to have a proper interface (for both the operators and the application) to query the available compute and communication resources from the system.

- **(3) Service assurance.** Due to the stringent Quality of Experience (QoE) requirements of edge applications, service assurance (SA) is also essential. SA continuously monitors service performance to ensure that the distributed computing and communication system meets the applicable Service Level Objectives (SLOs). If the SLOs are not met, corrective actions can be taken by the service provider, the application, or the network provider. The evaluation of SLO compliance needs to consider both computing metrics (e.g., compute latency, memory requirements) and communication metrics (e.g., bandwidth, latency). Corrective actions can include both new service placement and new service selection tasks. For instance, upon detecting that a certain compute node is overloaded, increasing the compute delay above the corresponding SLO threshold, the application can reinvoke service node selection (2.a) to switch its workload to another less utilized compute node. Similarly, upon detecting that a certain communication link is congested, increasing the communication delay above the corresponding SLO threshold, the application can reinvoke service path selection (2.b) to switch the data flow to another less congested link. If SA detects that there are not enough compute or communication resources to guarantee the SLOs, it can also invoke service placement (1) to allocate additional compute and communication resources.

Table 1 summarizes the problem space, the information that needs to be exposed, and the stakeholders that need this information.

Table 1: Problem space, needs, and stakeholders.

Action to take	Information needed	Who needs it
(1) Service placement	Compute and communication	Service provider
(2.a) Service selection: compute node selection	Compute and communication	Network operator, service provider or application
(2.b) Service selection: path selection	Communication	Network operator or application
(3) Service assurance	Compute and communication	Network provider, service provider or application

2.2 Existing Standard Gaps

There exist two main gaps in the current standard specifications that prevent network operators, service providers, and applications to optimally deploy and select services in the edge cloud: (1) modeling of compute resources and their performance metrics; and (2) exposure of metrics to the service providers and applications.

We explain these two gaps in the following two sections respectively.

2.2.1 Gap in Modeling of Compute Resources and their Performance Metrics

The modeling of compute resources and their performance metrics for edge computing presents unique challenges and opportunities compared to traditional cloud environments. This distinction arises primarily due to the inherent differences in the nature and management of resources between the two paradigms.

While the cloud is a highly homogeneous environment where all the compute and communication resources are typically uniform, the edge is composed of heterogeneous resources. These resources are provided by a variety of hardware and software vendors and are managed by different network operators. This heterogeneity introduces complexity in modeling, as it requires accommodating a wide range of devices, capabilities, and performance characteristics.

Traditional clouds are developed by single organizations, such as Amazon, Google, and Microsoft, which have full control over all the resources. These organizations can establish their own standards, which often become de facto standards, such as Kubernetes. In contrast, the edge is built by a variety of network operators and service providers utilizing heterogeneous resources. This diversity necessitates the development of open standards to enable interoperability in such a heterogeneous environment. Without these standards, it would be challenging to ensure seamless integration and efficient operation across different edge devices and platforms.

While the IETF has developed a wide range of standard specifications to model communication resources (e.g., [5], [18]), there is a notable gap in the development of specifications for modeling computing resources, which now can be perceived as an extension of the traditional network. Additionally, there is a need for detailed specifications that outline the key metrics required to properly characterize these models.

2.2.2 Gap in Exposure of Metrics to the Service Providers and Applications

In addition to the preceding gap in modeling and metrics, there is also the gap in designing an open standard for exposing these metrics to service providers and applications. Such a standard would ensure that all stakeholders have access to consistent and reliable information about the performance and capabilities of compute resources at the edge.

There are two primary mechanisms for exposing compute metrics: *on-path* and *off-path*. On-path exposure communicates compute information using the same path taken by the data plane. This method ensures that the information is directly tied to the data flow, providing real-time insights into the compute resources being used. On the other hand, off-path exposure generically relies on a proxy server that gathers compute information and exposes it to the service provider or application. This approach can offer a more centralized and potentially more scalable solution, as the proxy server can aggregate data from multiple sources and provide a comprehensive view of the compute resources. It also enables *separation of concerns* and avoids external applications to interact with internal network protocols for the purpose of security, privacy, scalability, etc.

Currently, standards exist for exposing communication metrics —e.g., bandwidth, latency. For example, the IETF ALTO protocol provides a framework for exposing network information to applications, such as metrics and properties, enabling them to make informed decisions based on the available communication resources. However, there is a notable gap when it comes to similar standards for compute metrics. To address this gap, there is a need to develop specifications that detail how compute metrics should be exposed. These metrics could include CPU and GPU usage, memory availability, and processing power, among others. By providing a standardized way to expose these metrics, service providers and applications can better optimize their operations and improve overall performance.

3 Design Considerations and Operator Requirements

Following the preceding gap analysis, we now present an initial draft of design considerations and operator requirements.

3.1 Considerations about selection and distribution of metrics

Compute metrics and their acquisition and management have been addressed by standardization bodies outside the IETF with the goal to guarantee reliable assessment and comparison of cloud services. The National Institute of Standards and Technology (NIST) proposes a framework in [10] that identifies and characterizes the information and relationships needed to describe and measure properties of cloud services that are representative, accurate, and reproducible. To this end, they consider three areas: (i) metrics for selecting cloud services, (ii) metrics for service agreements, and (iii) metrics for service measurement and verification. The Distributed Management Task Force (DMTF) defines a Base Metrics Profile (see [3]) that defines the minimum object model to specify a metric scope and acquisition context.

Once defined, the compute metrics are to be selected and exposed to management entities acting at different levels, such as a centralized controller or a router, taking different actions such as service placement, service selection, and assurance, with decision scopes ranging from local compute facilities to end-to-end services. Parameters reflecting these aspects are being investigated in [11, 14, 16], which consider the following design space:

Collection and Distribution of Metrics vs. Dynamicity. Whether an on-path or off-path mechanism is used to perform the service placement, selection, and assurance, or a combination of both, depending on the specific use case.

Abstraction Level and Information Access. Entities that consume metrics for placement, selection, and assurance decisions often lack access to computing information, may not have sufficient details, or encounter vendor-dependent data.

3.2 Operator Requirements

We identify the following as an initial list of operator requirements when addressing the gaps.

Standardization and Compatibility. Exposure of compute and communication information should be (1) common for any vendor, (2) comparable, and (3) aligned over different standardization bodies.

Collection and Exposure. Compute and communication information should be (1) collectable or exposed via a standardized API, (2) available for cloud-native containerized and virtual network functions in a comparable manner, and (3) collectable for all network and application resources (compute, storage, interfaces, routers, etc.) included in the communication path.

Real-Time and Reporting. Compute and communication information should be (1) available in real-time, (2) fine-grained enough to be useful to the operator and service provider, and (3) should cover topics including performance (e.g., latency), available resources, or other functional KPIs.

Consistency and Usability. Compute and communication information (1) should be usable by monitoring and assurance systems, (2) should provide end-to-end observability by the composition of metrics, and (3) should deliver the appropriate information for domain orchestrator or multiple-domain orchestrator of the operator to control allocation of resources, and the application management for service instantiation.

Security and Reliability. Compute and communication information handling (1) should be robust against cyber-attacks, (2) should be reliable and trustworthy, (3) should be available for health check, and (4) should have the option to ensure privacy (e.g., for virtual private networks).

Scalability and Adaptability. Compute and communication information (1) should cover scale-in and scale-out mechanisms, (2) should be dynamic in the sense that network elements can be added and removed, and (3) exposure of metrics has to scale.

4 Highlights on Related Standardization Efforts

This section highlights current efforts related to the management of compute infrastructure in standardization bodies that also deal with communication standards.

4.1 IETF

Given the numerous metrics already defined, the main challenge for the IETF would be, first, to specify which of these metrics would be appropriate for their use cases and, second, to define and specify new metrics when they are lacking for important use cases.

The work on compute metrics at the IETF is in its early stages and mainly related to low-level infrastructure metrics, such as those in RFC7666 [6], which defines a portion of the Management Information Base for virtual machines (VMs) controlled by a hypervisor. This portion specifies low-level infrastructure metrics reflecting resources consumed by a VM, such as CPU, memory, and storage. Since 2020, two use cases have motivated work on higher-level compute metrics.

A use case for user service selection has been defined in the CATS WG, which specifies a framework in [15] to enable an ingress edge router to select the best possible compute facility to run a service, based on both compute and network metrics. The WG also looks at defining compute metrics such as total delay and server capability in [11] and different abstraction levels in [19]. While the CATS WG focuses on an on-path approach, this use case has been addressed by an off-path approach in [9], which proposes a light extension to the ALTO protocol to jointly expose network and compute metrics.

A use case for the placement of service functions has been defined in [8], where network metrics need to be combined with compute metrics when service functions are placed across different data centers. [8] proposes leveraging [9] for the joint exposure of both types of metrics.

For both use cases, a common understanding and exposure scheme for compute metrics is being investigated in [16]. This draft explores the parameters to consider when selecting appropriate metrics for a given use case, which may relate to the entities consuming the metrics, the actions they take, the performance they assess, or the level of accuracy they need. It also proposes two abstracted metrics that reflect the performance and cost related to selecting a given compute facility and that can be exposed via an off-path protocol.

4.2 3GPP

The 3GPP has established comprehensive guidelines for the management and orchestration of edge computing services in the specification TS 23.558, titled "Architecture for Enabling Edge Applications"[4]. This specification outlines several key functional entities that are integral to edge computing services. Key components of joint compute and communication infrastructure management defined in this specification include: (1) the Edge Enabler Server (EES), which facilitates the discovery and communication between application clients and Edge Application Servers (EAS); (2) the Edge Enabler Client (EEC), supporting EAS discovery and interaction on user equipment (UE); and (3) the Edge Configuration Server (ECS), managing configurations for dynamic service provisioning and continuity. Each of these components plays a crucial role in lifecycle management, provisioning, performance assurance, and fault supervision of edge services.

4.3 O-RAN

The O-RAN Alliance's Working Group 6 (WG6) on Cloudification and Orchestration focuses on the management of infrastructure that supports the disaggregation of radio access network (RAN) functions due to the functional split. This working group addresses several critical aspects, including architectural design, deployment scenarios, and information models.

WG6 specifications facilitate a flexible and scalable RAN architecture that can support multiple software implementations from different vendors on a common platform, including hardware accelerators such as FPGA, DSP, ASIC, and GPU. Key aspects of joint compute and communication infrastructure management addressed by WG6 include: (1) the Orchestration Architecture, which defines the interaction between the Service Management and Orchestration (SMO) framework and the O-Cloud; (2) Lifecycle Management, covering dynamic orchestration use cases such as scaling, slice management, and fault tolerance; (3) the Information Model, specifying the data structures for seamless integration and management of cloudified RAN infrastructure; and (4) Hardware Abstraction, which includes the Accelerator Abstraction Layer (AAL) to provide a unified interface for various hardware components [17].

4.4 ETSI MEC

The European Telecommunications Standards Institute (ETSI) has developed a comprehensive framework for Multi-access Edge Computing (MEC), which facilitates the control and operation of edge computing facilities across both single and multi-domain environments [12]. The ETSI MEC Industry Specification Group (ISG) aims to create a standardized, open environment that enables efficient and seamless integration of applications from various vendors, service providers, and third parties.

The ETSI MEC specifications relevant to the management of joint compute and communication infrastructure include several key areas: (1) Architectural Design, which outlines the framework and components for MEC systems; (2) Deployment Scenarios, providing guidelines for various network environments; (3) Metrics and Performance, establishing benchmarks for evaluating system efficiency; (4) Traffic Management, ensuring optimal data routing and load balancing; (5) Quality of Service (QoS) Measurement, defining methods to maintain high service standards; and (6) Security and Privacy, detailing mechanisms to protect data and applications within the MEC environment.

4.5 De facto Standards: Kubernetes

The cloud-native trend, which supports the deployment of service functions as combined microservices, is revolutionizing service operations. The major de facto standard for this cloud-native management approach is Kubernetes. Kubernetes orchestrates containerized applications, providing a robust platform for automating deployment, scaling, and operations of application containers across clusters of hosts [7].

Kubernetes provides several essential features for managing joint compute and communication infrastructure [7], including (1) Resource Management, offering timely information on resources; (2) Automatic Scaling, adjusting resources based on real-time metrics; (3) Self-Healing, ensuring high availability by automatically handling failures; (4) Service Discovery and Load Balancing, facilitating efficient traffic distribution; and (5) Security and Compliance, enhancing operations with built-in security measures like RBAC and network policies.

5 Conclusions

Traditional clouds are developed by single organizations that have full control over all resources and can establish their own de facto standards. In contrast, the edge is built by a variety of network operators and service providers utilizing heterogeneous resources. Optimizing communication paths and service selection while saving costs and resources requires a joint understanding of both compute and network capabilities, necessitating a standardized representation.

We have explored the promises and challenges of providing standardized models of unified compute and communication infrastructure, along with mechanisms to expose their properties. This position paper aims to build synergies within the IETF towards a common understanding of the modeling and exposure mechanisms for metrics reflecting compute capabilities. A proposed focus is to align this work with the requirements of network management entities that would consume compute-related information. A concurrent goal is to liaise with standardization bodies outside the IETF.

References

[1] Serving opt-175b, bloom-176b and codegen-16b using alpa. Alpa Project.

[2] The future of ai is hybrid. 2023. Whitepaper, Qualcomm Technologies, Inc.

[3] D. N. D. V. 1.0.1. Base metrics profile. Technical report, 2009.

[4] 3GPP. Enabling edge computing applications in 3gpp. `https://www.3gpp.org/news-events/3gpp-news/edge-sa6`, 2021. Accessed: 2024-10-15.

[5] G. Almes, S. Kalidindi, M. J. Zekauskas, and A. Morton. A One-Way Delay Metric for IP Performance Metrics (IPPM). RFC 7679, Jan. 2016.

[6] H. Asai, M. MacFaden, J. Schönwälder, K. Shima, and T. T. T. ZOU). Management Information Base for Virtual Machines Controlled by a Hypervisor. RFC 7666, Oct. 2015.

[7] J. Bartlett. *Cloud Native Applications with Docker and Kubernetes: Design and Build Cloud Architecture and Applications with Microservices, EMQ, and Multi-Site Configurations*. Apress, Berkeley, CA, 2023.

[8] L. M. Contreras, S. Randriamasy, and X. Liu. ALTO extensions for handling Service Functions. Internet-Draft draft-lcsr-alto-service-functions-02, Internet Engineering Task Force, Mar. 2023. Work in Progress.

[9] L. M. Contreras, S. Randriamasy, J. Ros-Giralt, D. A. L. Perez, and C. E. Rothenberg. Use of ALTO for Determining Service Edge. Internet-Draft draft-contreras-alto-service-edge-10, Internet Engineering Task Force, Oct. 2023. Work in Progress.

[10] F. de Vaulx, E. Simmon, and R. Bohn. Cloud computing service metrics description. Technical report, NIST Cloud Computing Program, Advanced Networking Technologies Division, 2018.

[11] Z. Du, K. Yao, C. Li, D. Huang, and Z. Fu. Computing Information Description in Computing-Aware Traffic Steering. Internet-Draft draft-du-cats-computing-modeling-description-03, Internet Engineering Task Force, July 2024. Work in Progress.

[12] ETSI. Multi-access edge computing. `https://www.etsi.org/images/files/ETSITechnologyLeaflets/MultiAccessEdgeComputing.pdf`, 2024. Accessed: 2024-10-15.

[13] J. M. Halpern and C. Pignataro. Service Function Chaining (SFC) Architecture. RFC 7665, Oct. 2015.

[14] C. Li, Z. Du, M. Boucadair, L. M. Contreras, and J. Drake. A Framework for Computing-Aware Traffic Steering (CATS). Internet-Draft draft-ietf-cats-framework-03, Internet Engineering Task Force, Sept. 2024. Work in Progress.

[15] C. Li, Z. Du, M. Boucadair, L. M. Contreras, and J. Drake. A Framework for Computing-Aware Traffic Steering (CATS). Internet-Draft draft-ldbc-cats-framework-06, Internet Engineering Task Force, Feb. 2024. Work in Progress.

[16] S. Randriamasy, L. M. Contreras, J. Ros-Giralt, and R. Schott. Joint Exposure of Network and Compute Information for Infrastructure-Aware Service Deployment. Internet-Draft draft-rcr-opsawg-operational-compute-metrics-06, Internet Engineering Task Force, July 2024. Work in Progress.

[17] U. Schwager. Introduction to o-ran working group 6. https://docbox.etsi.org/Workshop/2023/03_NFVCONFERENCE/SESSION2_ETSI%20ISG%20NFV%20PARTNERSHIPS_COOPERATIONS/05_SESSION%202_Udi_Schwager_Introduction%20to%20O-RAN%20WG6.pdf, 2023. Accessed: 2024-10-15.

[18] Q. Wu, Y. R. Yang, Y. Lee, D. Dhody, S. Randriamasy, and L. M. Contreras. Application-Layer Traffic Optimization (ALTO) Performance Cost Metrics. RFC 9439, Aug. 2023.

[19] K. Yao, H. Shi, and C. Li. CATS metric Definition. Internet-Draft draft-ysl-cats-metric-definition-00, Internet Engineering Task Force, July 2024. Work in Progress.