Exploiting the Efficient Data Modeling in Network Digital Twin to Empower Edge-Cloud Continuum

Syed Mohsan Raza[1], Roberto Minerva[1], Noel Crespi[1], Maira Alvi[1], Manoj Herath[1], Hrishikesh Dutta[1]
[1]Samovar, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France.

Email: {syed-mohsan_raza, roberto.minerva, noel.crespi, maira.alvi, manoj.herath, hrishikesh.dutta}@telecom-sudparis.eu

***Abstract*—Specifications for Network Digital Twin (NDT) from Standardization Development Organizations (SDOs), such as the Internet Engineering Task Force (IETF), and academic contributions focus primarily on benefiting network operators. However, they often overlook the needs of stakeholders in the Edge-Cloud Continuum (ECC), such as Service Providers, customers, and Platform or Infrastructure Providers. In ECC, resource heterogeneity, agile software component integration, and quality of service requirements are challenges. To address these challenges, continuous and granular monitoring of software and physical resources is required. In this paper, we present the design and ongoing implementation of a data model. It captures and characterizes the physical and software properties, i.e., Key Performance Indicators (KPIs), of Kubernetes-managed components in the ECC. Collected data is structured in NGSI-LD-compliant format and managed through interoperable context brokers for authenticating the requests of various stakeholder applications. We also demonstrate sample data curation from a lab-configured platform, its integration into the context broker, and how it responds to the queries concerning a particular component information, thereby representing a partial implementation of proposed data model to render the views for particular stakeholders.**

***Index Terms*—Network Digital Twin, Edge-Cloud Continuum, Data Model, Kubernetes, Microservices**

I. Introduction

The Network Digital Twin (NDT) represents network components and resources, both physical and virtual, along with their interactions. It serves as a logical counterpart to the physical network, providing a holistic view of the entire communication system and its subsystems. NDT models the infrastructure by characterizing network components, called basic or descriptive models. It replicates the communication events, resource distribution, and the complexities of modern communication systems. This representation allows the gain of more functionally separated models that benefit various stakeholders. In definitions and specifications from Network Management Research Group (NMRG) in Internet Research Task Force (IRTF) [1] and International Telecommunication Union (ITU) [2], these functional models are particularly useful for Network Operators (NOs). According to these specifications, NDT enhances the efficiency of traditional network processes such as troubleshooting, fault management, configuration, and simulation.

NDT extensively collects data from network components such as their states, resource usage and configurations that help to simulate and identify near-optimal configurations for specific Quality of Service (QoS) demands. This supports evaluating function placement strategies at both global and local levels, allowing NOs to observe, assess, and adjust potential service allocations. As a result, NDT is a valuable tool and a safe sandbox for planning solutions to support both current and future service offerings without resource jeopardizing. Due to these primary benefits from the NOs perspective, several telecommunication companies such as Nokia, Huawei, Ericsson, and Orange are investing to empower the NDT with a cost-effective platform that is also highly available for fast prototyping [3].

A. Motivation

Here are a few motivations driving our efforts to contextualize the ECC using enhanced capabilities through relevant and efficient data modeling in NDT.

ECC is a complex continuum integrating Edge, Core, and Cloud segments, where computational capacity increases from Edge to Cloud. Micro data centers at the Edges have limited computation compared to macro data centers in the Core and the Cloud. Collaboration between these segments involves heterogeneous resources, orchestration challenges, and application placement, requiring advanced data modeling like NDT to characterize these components fully.

Besides the benefits of NDT for Network Operators (NOs), other ECC stakeholders—Service Providers (SPs), Service Customers (SCs), and Infrastructure Providers (IPs)—are not clearly addressed in current NDT specifications, limiting their ability to leverage NDT within ECC segments.

As ECC segments adopt softwarization, openness, and components with dynamic behavior. Stakeholders require granular resource monitoring down to application functions, driven by Microservices-Based Architectures (MBA). MBAs enable fast, independent development, and container-based virtualization accelerates microservice deployment, improving time to market.

Ontology-based knowledge representation provides reusable vocabularies for system interactions. Existing ontologies cover network components and applications, however, limited in representing ECC segments, microservices, and containerization in NDT, as they do not meet evolving ECC stakeholder demands.

Lastly, the ITU [2] and NMRG [1] specifications regarding the NDT concept recommend leveraging the new functional data model by utilizing the data/attribute dimensions of existing models, in case the required functional models are not available.

B. Contributions

Considering the above motivation and research gaps of NDT capabilities to represent the ECC, we have the following contributions.

- A literature review is briefly conducted to validate the aforementioned gaps.
- A data model to contextualize the ECC using both high-level (abstract) and detailed ontological representations is described.
- Extending our previous work [4], which involved the early design of the data model by investigating the various layers in ECC with separate concerns and functions, we present a succeeding detailed ontology and implementation phase of the proposed model.
- A set of used tools for data collection are briefly explained.
- A data sample collected using the proposed data model from a real Kubernetes-managed cluster is presented. Moreover, the presented data's NGSI-LD conversion adheres to FIWARE platform guidelines.

II. Related Work

To enhance network management, several research proposals from academia and Telcos present NDT as a promising model for approaches like Intent-based networking and Zero-touch network management [5]. These efforts, categorized under prescriptive modeling, leverage NDT for the business benefits of network operators (NOs). To achieve optimal task offloading, efficient resource utilization from Edge to Cloud, user mobility management, and application function placement, researchers have proposed the integration of various Machine Learning (ML) models in NDT [6]. For instance, Federated Learning for data collection from Edge nodes in Cloud NDT and handling network dynamics using agent-based approaches. However, the success of these ML/AI models depends on the data model. If the data model is coarse-grained and limited in capturing data dimensions and necessary attributes, the precision of the ML/AI model will deviate from expectations.

In ECC segments, Service Providers (SPs) deploy microservices, each serving different functions. Microservice-based Architecture (MBA) benefits users, Cloud, and network applications equally in the context of maintainability, easy interfacing, and functional traceability. While developers do not manage resource deployment during development, SPs must chain microservices to ensure optimal application response times [7]. To deploy and orchestrate microservices, Kubernetes is the widely adopted solution. However, fast integration through containerization introduces challenges like placement, scheduling, and autoscaling. NDT can improve orchestration decisions by collecting and processing data from Kubernetes-managed clusters [8].

Ontology-based domain or ECC segment knowledge representation is marginally exploited [9]. The collaborative work between the Ontology Engineering Group and Huawei Research Ireland offers a close match to our interests, i.e., the DevOps Infrastructure Ontology Catalogue [1]. Their generated ontologies are complete in terms of the objective and coarsely linking the entities; however, the catalogue does not present a readily available ontology to integrate into NDT. For instance, the relationship of mapping application components or microservices to a segment element, and the attributes that measure the impact of this mapping, are out of scope in these ontologies, while this is important for ECC stakeholders. Research from the Ontology Engineering Group [10] is closer to our interest, as it details ontology down to the microservices level. However, it still needs to be enhanced to characterize the impact of each microservice, such that different views on consumption data (computing, memory, energy) can be rendered for stakeholders.

III. Data Model

A. High Level Ontology

The proposed data model is primarily represented using a high-level ontology, developed in Protege [2], as shown in Figure 1. This ontology organizes the key entities/classes of a segment in the ECC in a structured manner and plans a reusable vocabulary of interrelating classes. NDT class in this ontology serves as the central entity, representing the entire Edge-Cloud infrastructure and allowing stakeholders to monitor network states and input specific parameters/concerns. The ontology includes both physical and software infrastructure classes. The physical infrastructure class captures components like CPU and storage, helping service providers manage resource availability and scalability. The software infrastructure class characterizes virtual resources, such as virtual machines and containers, and also includes the orchestration layer, which curates data from orchestration solutions like the Kubernetes control plane. Additionally, the software infrastructure class acts as an abstraction for service providers, enabling the deployment of componentized applications and supporting Service Function Chaining. This includes user applications linked with specific system microservices, such as a packet parser virtual network function (VNF), which can be a key component in the chain of control of any application.

B. Detail Ontology

The detailed ontology is specifically structured to characterize the components of the segments in terms of their consumption, depicted in Figure 2. This is essential for capturing the impact of the placement of various functions in any segment involved in the ECC. It includes classes that represent the hierarchical structure of data center management, such as site, room, rack, pods, clusters, and their specific attributes. The detailed ontology focuses on capturing the identification and

[1] https://oeg-upm.github.io/devops-infra/index.html
[2] https://protege.stanford.edu/

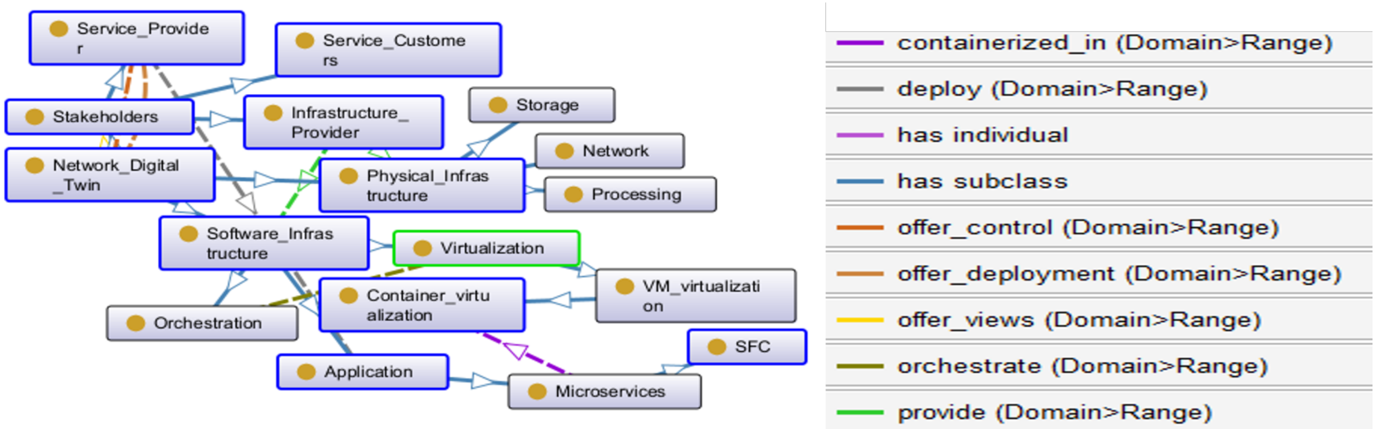

Fig. 1. High-level abstract representation of NDT

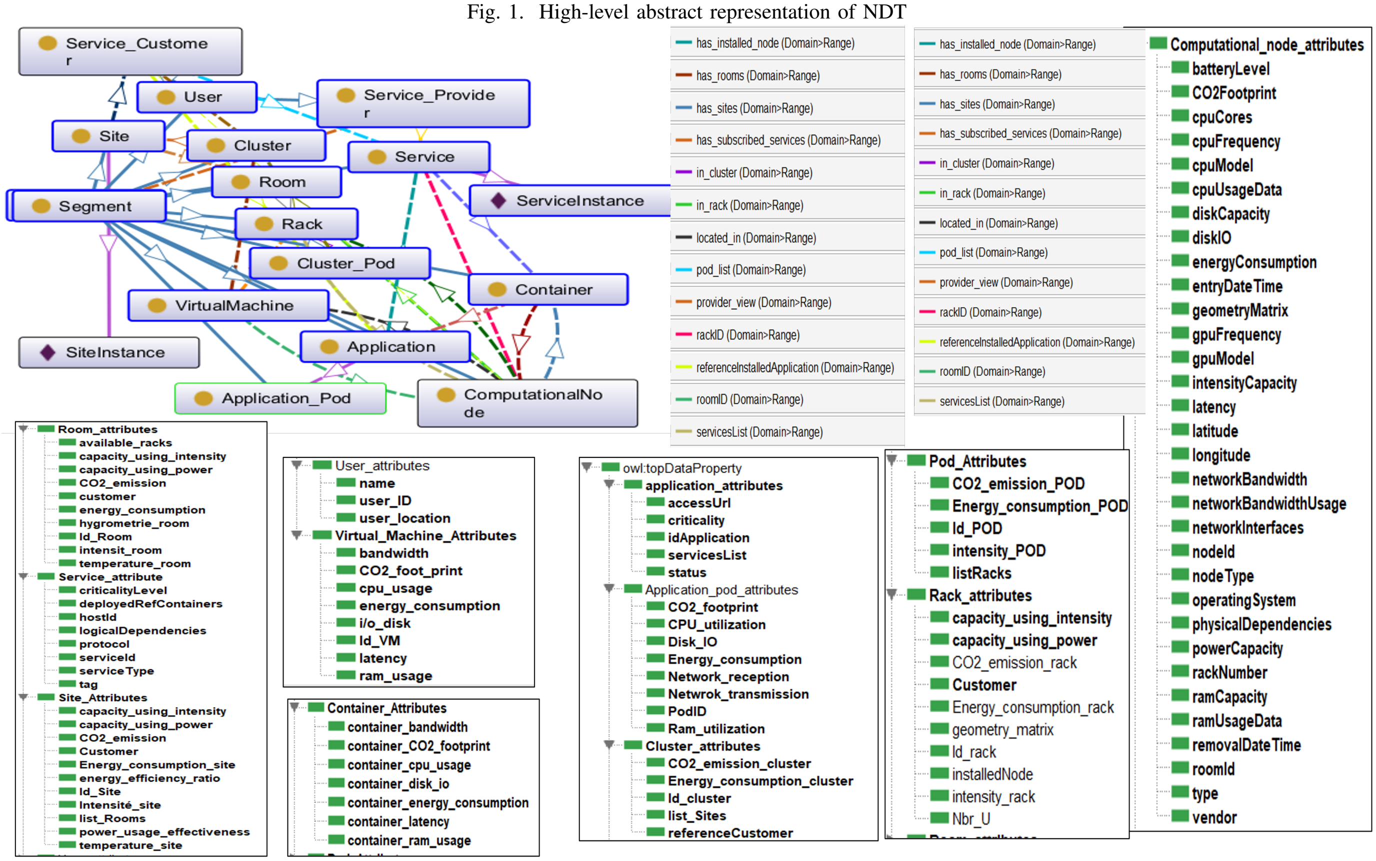

Fig. 2. Detailed Ontology representation

consumption (e.g., CPU, memory, power, carbon emissions) of microservices deployed in containers on computational nodes. It also considers the geo-location attributes of these nodes, allowing stakeholders to assess the proximity of individual functions or entire services (comprising various microservices and standalone applications). For example, it includes geometric attributes of the computational nodes.

C. Ontology Transformation into Smart Data Model

The transformation of the above ontology into a smart data model, following FIWARE's data development guidelines, is ongoing[3]. This transformation extends the EGM-provided vocabulary [11] for contextualizing segment resources in the ECC. Stellio Broker, a Context-Broker that operates on the publish-subscribe model, is utilized to control the data flow and perform create, read, update, and delete (CRUD) operations over HTTP in compliance with NGSI-LD standards [12].

[3]https://github.com/easy-global-market/c2jn-data-models/tree/main/ngsild-payloads

D. Data Collection and Characterising it using Proposed Data Model

To collect the real-time data from the Kubernetes deployed cluster according to specific ontology, we have developed an Edge topology of six computational nodes, consisting of two servers and four Raspberry Pi. Note that, due to space limitations, the used topology and configurations are not illustrated here. K3s [4] Kubernetes is installed along with Prometheus distributed tracing solutions [5]. Various exporters such as Kube State metric [6], node exporters, and benchmark applications i.e., Sock-Shop [4] are deployed for testing and validating the approach.

Pod Name	Time	CPU Utiliz	Node Nam	emory Util	rk Receive	rk Transmi	isk IO Utili	Labels	Location
api-gatew	2024-08-3	0.0007814	imtserver	452.8125	0	0	546.13333	app=stelli	IMTBS -Edge
carts-666b	2024-08-30 21:05:12		cube14		0	0		app=mohs	IMTBS -Edge
carts-db-6	2024-08-30 21:05:12		cube05		0			app=mohs	IMTBS -Edge
catalogue	2024-08-30 21:05:12		cube01		11.4	0		app=mohs	IMTBS -Edge
catalogue	2024-08-30 21:05:12		cube14		0	0		app=mohs	IMTBS -Edge
continuou	2024-08-3	0	imtserver	0.1914062	0	0		app=jupyt	IMTBS -Edge
coredns-5	2024-08-3	0.0023443	imtserver	61.003906	442.13333	438.83333	0	k8s-app=k	IMTBS -Edge
csi-attach	2024-08-3	0.0003880	cube03	9.5742187	103.96666	21.866666	0	app=csi-at	IMTBS -Edge

Fig. 3. Sample data collection of Pod consumption using software agent that fetches data from Prometheus endpoints

{ "id": "urn:ngsi-ld:ApplicationPod:carts-db-644ff6b576-kh7qx", "type": "https://vocab.egm.io/ApplicationPod", "name": { "type": "Property", "value": "carts-db-644ff6b576-kh7qx" }, "https://vocab.egm.io/user": { "type": "Relationship", "object": "urn:ngsi-ld:ServiceProvider:Mohsan_IMTBS" }, "https://vocab.egm.io/diskIo": { "type": "Property", "value": "", "unitCode": "AD", "observedAt": "2024-09-03T19:34:51Z" }, "https://vocab.egm.io/deployedIn": { "type": "Relationship", "object": "urn:ngsi-ld:Server:cube05" }, "https://vocab.egm.io/applicationRef": { "type": "Relationship", "object": "urn:ngsi-ld:Application:APP123" }, "https://vocab.egm.io/cpuUtilization": { "type": "Property", "value": "", "unitCode": "P1", "observedAt": "2024-09-03T19:34:51Z" }, "https://vocab.egm.io/ramUtilization": { "type": "Property", "value": "", "unitCode": "AD", "observedAt": "2024-09-03T19:34:51Z" }, "https://vocab.egm.io/networkReception": { "type": "Property", "value": 0, "unitCode": "AD", "observedAt": "2024-09-03T19:34:51Z" }, "https://vocab.egm.io/networkTransmission": { "type": "Property", "value": 0, "unitCode": "AD", "observedAt": "2024-09-03T19:34:51Z" }, "temporalResolution": { "type": "Property", "value": "PT1H" }, "@context": "https://easy-global-market.github.io/c2jn-data-models/jsonld-contexts/c2jn-compound.jsonld" }

Fig. 4. Stellio Broker response over HTTP for a specific application pod (Cart microservice of Sock-Shop Application)

A sample data collection from a specific time instance is shown in Figure 3. The data is collected by scraping the endpoint of microservices in the Sock-Shop application using a Prometheus exporter. After transforming this data and posting it to the broker according to integrity and linking information rules in the proposed data model, a simple Python application is developed. This application interacts with the data through CRUD operations and fetches the Cart microservice consumption (a component of the Sock-Shop application in underlying computational nodes. A response from the context broker, triggered by the developed Python application, is depicted in Figure 4.

IV. Conclusion and Future work

This paper presents a data modeling approach for the efficient management of the Edge-Cloud Continuum. It provides granular contextualization of the entities and classes in a segment involved in the Edge-Cloud Continuum. Additionally, it demonstrates a use case involving the collection of sample application resource consumption data, sending it to the broker according to the proposed data model, and retrieving the highly linked data by a third-party application. In this way, the proposed data model will be capable of supporting a full view of ECC components in a Network Digital Twin, once the model is fully functional.

In the future, the topology will be expanded to three clusters located in different locations in the Paris region to validate the effectiveness of this data model. Various smart city applications being developed for the C2JN project, as well as benchmark applications will be tested for their resource consumption, i.e., KPIs. The specific logic to render the views concerning stakeholders will be implemented.

Acknowledgment

The research work is funded by CLOUD CONTINUUM SOUVERAIN ET JUMEAUX NUMÉRIQUES (C2JN), under grant number AMI CLOUD-1 C2JN (DOS0179613/00,DOS0179612/00). Authors are thankful to the project partners for their collaboration.

References

[1] C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, C. Jacquenet, Network Digital Twin: Concepts and Reference Architecture, Internet-Draft draft-irtf-nmrg-network-digital-twin-arch-06, Internet Engineering Task Force, work in Progress (Jul. 2024). URL https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/06/

[2] Y. S. . Y.3090, Digital twin network - requirements and architecturedigital twin network - requirements and architecture, https://www.itu.int/rec/T-REC-Y.3090.

[3] Nokia, Network digital twin, https://www.nokia.com/networks/solutions/network-digital-twin/.

[4] S. M. Raza, R. Minerva, N. Crespi, M. Karech, Definition of digital twin network data model in the context of edge-cloud continuum, in: IEEE International Conference on Network Softwarization (NetSoft 2023), 2023.

[5] N. Apostolakis, L. E. Chatzieleftheriou, D. Bega, M. Gramaglia, A. Banchs, Digital twins for next-generation mobile networks: Applications and solutions, IEEE Communications Magazine (2023).

[6] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong, et al., Network digital twin: Context, enabling technologies, and opportunities, IEEE Communications Magazine 60 (11) (2022) 22–27.

[7] S. M. Raza, R. Minerva, B. Martini, N. Crespi, Empowering microservices: A deep dive into intelligent application component placement for optimal response time, Journal of Network and Systems Management 32 (4) (2024) 84.

[8] D. Borsatti, W. Cerroni, L. Foschini, G. Y. Grabarnik, L. Manca, F. Poltronieri, D. Scotece, L. Shwartz, C. Stefanelli, M. Tortonesi, et al., Kubetwin: A digital twin framework for kubernetes deployments at scale, IEEE Transactions on Network and Service Management (2024).

[9] E. Karabulut, S. F. Pileggi, P. Groth, V. Degeler, Ontologies in digital twins: A systematic literature review, Future Generation Computer Systems (2023).

[10] O. Corcho, D. Chaves-Fraga, J. Toledo, J. Arenas-Guerrero, C. Badenes-Olmedo, M. Wang, H. Peng, N. Burrett, J. Mora, P. Zhang, A high-level ontology network for ict infrastructures, in: International Semantic Web Conference, Springer, 2021, pp. 446–462.

[11] A. Abid, C. Dupont, F. Le Gall, A. Third, F. Kane, Modelling data for a sustainable aquaculture, in: 2019 Global IoT Summit (GIoTS), IEEE, 2019, pp. 1–6.

[12] A. Lazidis, E. G. Petrakis, S. Chouliaras, S. Sotiriadis, Open-source publish-subscribe systems: A comparative study, in: International Conference on Advanced Information Networking and Applications, Springer, 2022, pp. 105–115.

[4] https://github.com/k3s-io/k3s

[5] https://github.com/prometheus/prometheus

[6] https://github.com/kubernetes/kube-state-metrics