Rethinking AI-Powered Service Orchestration: The Case for Decentralization

Jesus Perez-Valero, Gines Garcia-Aviles, Anastasios E. Giannopoulos, Sotirios T. Spantideas, Antonio Skarmeta, Sławomir Kukliński

2025 IEEE 11th International Conference on Network Softwarization (NetSoft) | 979-8-3315-4345-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/NETSOFT64993.2025.11080596

***Abstract*—The evolution of cloud computing towards a cloud continuum, including cloud, edge, and far-edge resources, is revolutionizing the deployment, management, and orchestration of Network Services (NSs) and applications. Traditional, centralized orchestration approaches are increasingly inadequate for handling the complexity, scale, and dynamic nature of this continuum. In this paper, we present a data-driven approach for AI-powered service orchestration based on the European 6G-CLOUD project. Specifically, we introduce the Decentralized Service Orchestrator (DSO) framework, an AI-powered, decentralized orchestration model that leverages the capabilities of the Artificial Intelligence and Machine Learning Framework (AI/MLF) to enable intelligent, autonomous, and scalable service lifecycle management across heterogeneous environments. Key contributions include the detailed architecture of the DSO, its workflows, and its integration with the Cloud Continuum and with an AI/MLF that manage the AI lifecycle, enabling models provision to the different components. By enabling decentralized AI-driven decision-making, this framework enhances service reliability, scalability, operational efficiency, and innovation acceleration, paving the way for next-generation cloud continuum orchestration.**

***Index Terms*—6G, Cloud Continuum, Service Orchestration, Artificial Intelligence**

I. Introduction

The evolution of cloud computing towards a cloud continuum, spanning central cloud, edge, and far-edge resources, is transforming the way network services (NSs) and applications are deployed, managed, and orchestrated. In this context, orchestration has emerged as a key enabler to dynamically allocate resources, automate service provisioning, and optimize performance across distributed infrastructures. Specifically, service orchestration refers to the automated coordination and management of computing, networking, and storage resources, with a particular focus on scaling resources to ensure seamless service delivery. However, traditional centralized orchestration approaches [1] struggle to handle the complexity, scale, and dynamic nature of the cloud continuum, necessitating a distributed management and orchestration framework [2] capable of adapting to real-time changes and optimizing resource utilization, especially when orchestration is AI-powered such as [3]–[5].

To address these challenges, modern orchestration frameworks must enable scalable, flexible, and autonomous service management across heterogeneous and distributed environments. A distributed model—enhanced by AI and data-driven automation—supports intelligent lifecycle management of network services (NSs), leveraging real-time monitoring, predictive analytics, and intent-based networking to improve reliability, fault tolerance, and efficiency [6]. However, orchestrating services across the cloud continuum faces significant challenges, including resource fragmentation, complex workload balancing, interoperability issues, and the need for real-time decision-making to handle dynamic conditions and ensure effective service delivery [7], [8].

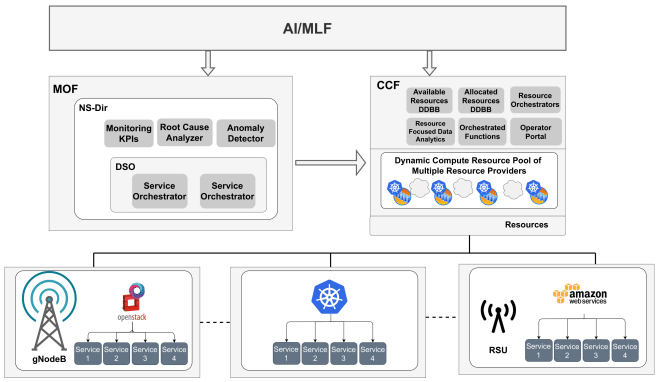

Fig. 1. High-level view of the 6G-CLOUD architecture, including the Management and Orchestration Framework (MOF), Cloud Continuum Framework (CCF) and Artificial Intelligence & Machine Learning Framework (AIMLF) which support different domains (e.g., RAN, Core and Edge)

A fundamental shift towards data-driven orchestration is key to overcoming these challenges. Traditional rule-based orchestration mechanisms, which rely on static policies [1], [9], are being replaced by AI/ML-driven approaches that continuously learn from operational data to optimize service performance. The Artificial Intelligence/Machine Learning Framework (AIMLF) is a key component in the transformation, providing a complete framework for models to be generated and trained from data exposed by the cloud continuum. This enables a continuous training of models that are available for the orchestration entities toward achieving an efiicient and effective operation.

On the other hand, one of the primary advantages of an orchestrable infrastructure is the ability to offer Orchestration as a Service (OaaS) [10], opening orchestration to third-

Jesus Perez-Valero, Gines Garcia-Aviles, and Antonio Skarmeta are with University of Murcia, Spain. Gines Garcia-Aviles is also with i2CAT Foundation. Anastasios Giannopoulos and Sotirios Spantideas are with the Research & Development Department of Four Dot Infinity (FDI), Athens, Greece. Sławomir Kukliński is with the Warsaw University of Technology.

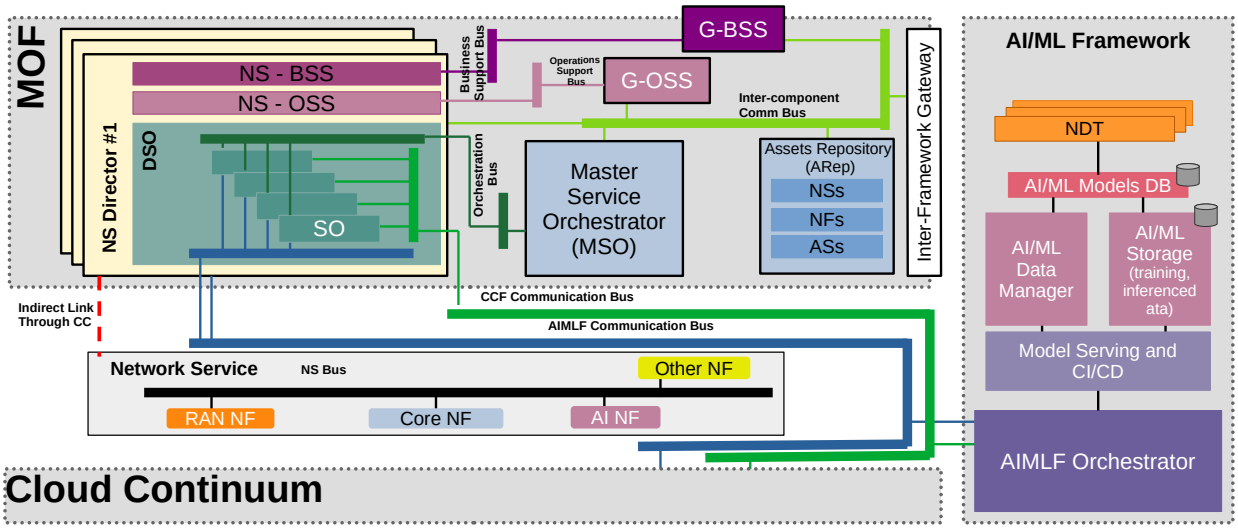

Fig. 2. Architectural building blocks of the MOF Architecture.

party providers, enabling the ability to request orchestration functions as needed and allowing the presence of orchestration marketplaces where specialized orchestration solutions will be available for the different frameworks of the architecture. This approach brings several benefits, including enhanced flexibility, scalability, fault tolerance, NS-customised orchestration and accelerated innovation.

Additionally, the decoupling of resource and service or chestration minimizes the impact of failures, ensuring greater resilience in distributed environments because failures in one layer do not directly propagate to the other. For example, if a resource (e.g., a server or network link) fails, the ser vice orchestration layer can dynamically adapt by reassigning workloads to available resources without disrupting the entire system. The Management and Orchestration Framework (MOF) illustrated in Fig. 1 implements the Service Orchestrators following a per-Network Service (NS) approach and enabling an effective and isolated operation. Section III further extend the description of the funcional blocks involved in this process.The architecture also highlights the role of the AIMLF and Cloud Continuum (CC) in facilitating AI-driven decision-making and real-time resource adaptation, which are critical for achieving the flexibility and resilience required for OaaS in 6G networks. The architecture also highlights the role of the AIMLF and CCF in facilitating AI-driven decision-making and real-time resource adaptation, which are critical for achieving the flexibility and resilience required for OaaS in 6G networks.

In this paper, we present data-driven approaches to orchestration in the cloud continuum, with a focus on defining the DSO, detailing its workflows, and presenting its interactions with other architectural components. In a nutshell, the main contributions of the paper are as follows:

- We present the DSO as a key component for intelligent, autonomous, and scalable service orchestration across heterogeneous cloud environments. We describe its architecture, workflows, and integration with the Artificial Intelligence/Machine Learning Framework (AIMLF).
- The proposed framework leverages AI/ML models to enable real-time decision-making, anomaly detection, predictive scaling, and fault management, enhancing service reliability, operational efficiency, and adaptability in the cloud continuum.
- To validate the workflows of the framework, we present a use case where we implement an AI-driven self-healing mechanism for latency-aware Virtual Network Function (VNF) scaling. This use case demonstrates how real-time telemetry, predictive analytics, and automated decision-making within the DSO can proactively detect and mitigate potential SLA violations.
- We explore AI-driven optimization in decentralized service orchestration, emphasizing its benefits for real-time decision-making, scalability, and fault tolerance. We introduce how advanced knowledge-sharing mechanisms—such as Federated Learning, Reinforce ment Learning, and Graph Neural Networks that enable secure collaboration among decentralized orchestrators. These techniques enhance localised intelligence and facilitate system-wide adaptability, addressing challenges like privacy, interoperability, and dynamic model management in heterogeneous environments.

The rest of the paper is structured as follows. In Section II, we present some of the challenges and requirements for data-driven service orchestration. In Section III, we introduce the Management and Orchestration Framework (MOF), along with the DSO and its key workflows. Additionally, we present a use case to illustrate its application. In Section IV, we provide the benefits of DSO for AI-enabled SO together with knowledge-sharing techniques and different challenges still to be addressed within this field. Finally, in Section V, we present some concluding remarks.

II. Challenges and key requirements for Orchestrable data-driven service orchestrators

One of the main challenges in orchestration frameworks is separating service orchestration from resource orchestration. Service orchestration focuses on managing workflows and business processes, while resource orchestration handles the allocation of computing and network resources. This separation improves scalability and flexibility, especially in dynamic environments like the cloud, where demands frequently shift. It allows each layer to adapt independently—for example, resources can be scaled up or down based on current needs, while services continue functioning smoothly.

Separating these two layers also enhances fault isolation and system resilience. In monolithic systems, failures in one part can cascade into others, causing widespread issues. With decoupling, faults are confined to their own domain —so a problem with resource allocation doesn't necessarily affect service operations. This leads to more robust and reliable orchestration systems.

In addition to improving fault tolerance, this separation enables faster and more flexible innovation. Since the layers can evolve independently, organizations can update or enhance one without disrupting the other. For example, integrating AI into resource orchestration can be done without altering service orchestration, allowing for quicker experimentation and adaptation in fast-changing tech environments.

III. Decentralized Service Orchestration

A. Management and Orchestration Framework: the 6G Cloud vision

The Management and Orchestration Framework (MOF) in the 6G-Cloud project is a key architectural enabler designed to provide scalable, distributed and intelligent orchestration across the cloud continuum. This need for a more advanced orchestration model arises because current NSs are increasingly complex, and a single SO may struggle to efficiently manage distributed workloads, inter-domain dependencies, and dynamic scaling requirements. For example, a network service composed by a 5G virtual network may benefit from using two service orchestrators where one can be dedicated to the core network, which is instantiated in a multi-domain architecture and the other to the RAN, which follows an Open RAN architecture. As a result, the decentralized orchestration approach comes into the scene to enable a more granular and adaptive approach. The key features are the following:

- **Service and Resource Separation:** The MOF implements a programmable Orchestration-as-a-Service (OaaS) model where service and resource orchestration are split into independent operations. This enables dynamic resource allocation for specific NSs while reducing the operational overhead, and, at the same time, service orchestration can be decentralized and orchestrable.
- **AI-Driven Orchestration and Automation:** The MOF integrates AI/ML-based intelligence to support intelligent resource allocation, anomaly detection, fault management and real-time optimizations over the NSs.

The main building blocks start with the Master Service Orchestrator (MSO), which oversees the global orchestration process and initializes the creation process of network servicess. Then, the Network Service Director (NSDir) is designed to manage a specific NS, comprising the Service Orchestrators (SOs) within a controlled logical component (Decentralized Service Orchestrator) to perform pure management operations and real-time optimizations through AI-assisted models. After that, the Assets Repository (ARep) stores Network Functions (NFs) and NS templates for efficient deployment and lifecycle management, and the NS-OSS/BSS are specific instances of operations and business interfaces specifically created for each NS. In a more general way, the MOF also includes instances for the OSS/BSS that operate at framework level, exposing general operations that are not limited by the characteristics of an NS.

B. Architectural design of a DSO

To provide efficient management of complex NSs under distributed and heterogeneous workloads, the NSDir adopts a decentralized AI-powered approach for service orchestration where one or more SOs will be in charge of orchestrating the NS or a subset of elements of the NS as depicted in Figure 2.

The MSO manages the global orchestration vision, and hence, the different SOs conforming to the DSO are susceptible to receiving orchestration directives. The Orchestration BUS included in the figure (dark-green lines) enables direct communication among the MSO and DSO so that global orchestration can effectively be applied, directives that may involve the deployment or replacement of SOs dedicated to specific NSs (or elements within the NSs). Then, for the DSO to effectively orchestrate the NSs, it is also connected to the CCF Communication BUS (green lines), which acts as the endpoint for communicating with the NSs. Finally, the AIMLF also contributes to improving the overall performance of both NSs and the Cloud Continuum by providing both the MSO and the NSDir with the appropriate trained AI/ML models with different purposes. For instance, the AIMLF is connected to the SO and the CCF via the CCF Communication Bus (green lines) and the AIMLF Communication Bus (blue lines). The AIMLF supports end-to-end model lifecycle management in line with MLOps principles [11], following a structured pipeline that includes: (i) **Data Adaptation:** preprocessing and transforming real-time and batch data for training; (ii) **Model Training:** using diverse ML techniques in scalable, distributed setups; (iii) **Testing and Validation:** evaluating model performance and generalization; and (iv) **Deployment:** using containerized/serverless environments with continuous monitoring for drift and retraining.

While providing models is the most basic function of the AIMLF, the framework is designed to support the full lifecycle management of AI functions by enabling the exchange of orchestration information and AI/ML models through its

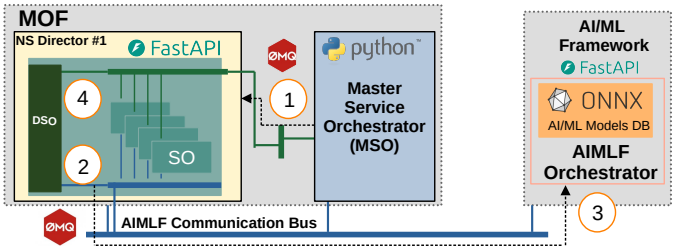

Fig. 3. Architecture AIaaS workflow for the DSO

integration with the MSO, NSDir, and DSO via a communication bus. Building on this, the OaaS paradigm enhances the decentralized orchestration approach by allowing the DSO to deliver dynamic, programmable orchestration across heterogeneous environments, with the NSDir offering service-agnostic orchestration functionalities (SOs) decoupled from specific network services.

C. Decentralized Intelligent Orchestration Model

As stated previously, one of the key advantages of the proposed orchestration approach is the OaaS combined with a decentralized and AI-enabled SO approach to enhance their performance. The process of orchestrating AI models required the utilization of the abovementioned communication buses and a set of interactions among modules described in Figure 3. First, the MSO notifies the DSO that there could be a situation that requires the addition of an AI model to prevent or enhance a specific situation (Step 1). For example, the MSO can predict congestion situations in certain parts of the continuum at which NSs are placed, energy consumption above a certain threshold or security concerns that may require these AI-trained models to follow the situation and put specific actions to prevent/solve it. Once the DSO has been notified, it requests AI models to the AIMLF through the AIMLF Communication Bus (Step 2), indicating a specific model if the MSO previously notified this information or models that meet concrete criteria informed by the MSO. Based on this information, the AIMLF Orchestrator will select the most suitable model and reply back to the DSO (Step 3). Finally, the DSO will inject the received model into the specific SO/SOs through the orchestration bus (Step 4).

Similarly, the AIMLF can also notify the MSO or the NSDir of any anomaly or situation that requires specific actions (e.g. anomaly detected in a certain NS or some predictions that require updates on the NS). This communication will go through the Inter-Framework Gateway, and if it requires the deployment of new models within the DSO, the subsequent steps are the same as the ones previously introduced (Figure 3).

D. Use case and preliminary results

To validate the architecture and workflows proposed in Section III, we present a use case that demonstrates the practical application of AI-driven self-healing mechanisms in 6G cloud networks. Specifically, this use case focuses on latency-aware Virtual Network Function (VNF) scaling, leveraging real-time latency monitoring, machine learning (ML)-based predictions, and a threshold-based decision-making framework to ensure service-level agreement (SLA) compliance while optimizing resource utilization.

The functionality of the system is depicted in Fig. 4, where each step of the process is illustrated. Below, we provide a detailed description of each step.

- **(1) Latency Prediction Model Training:** The AIMLF uses real-time and historical data from the Resource Database to train latency models, which are stored for deployment as shown in Step #1 of Fig. 4.
- **(2) Real-Time Prediction and Anomaly Detection:** SO_1 in the DSO uses the deployed model to predict SLA violations and detect anomalies in real-time (Step #2 in Fig. 4).
- **(3) Root Cause Analysis:** SO_2 performs AI-based diagnosis to identify causes of latency issues, such as traffic spikes or VNF inefficiencies (Step #3 in Fig. 4).
- **(4) Corrective Actions:** The NSDir initiates VNF scaling or resource reallocation based on SO insights to restore SLA compliance (Step #4 in Fig. 4).
- **(5) Continuous Monitoring and Retraining:** The AIMLF monitors model performance and triggers retraining with updated data when accuracy drops (Step #5 in Fig. 4).

By following these steps, the system ensures proactive latency management, optimal resource allocation, and autonomous network adaptation. The resulting benefits are detailed below.

- **SLA Compliance and Service Reliability:** Proactively predicts and prevents latency issues to meet URLLC requirements, ensuring reliable performance for critical applications.
- **Energy Efficiency and Resource Optimization:** Optimizes VNF usage to reduce energy consumption and costs, activating resources only when needed.
- **Self-Healing and Autonomous Network Operation:** Uses AI to detect, predict, and resolve issues automatically, enhancing resilience and reducing downtime.
- **Improved Scalability and Adaptability:** Supports dynamic scaling and adapts to network changes through continuous learning and retraining.

To validate the feasibility of the proposed workflow, we have developed a small-scale PoC using state-of-the-art technologies as depicted in Figure 3. The implementation consists of a microservices-based approach where each functional block is implemented using *Python FastAPI*, which provides high performance and scalability with a minimal overhead. The AI/ML Framework includes a direct connection with a database of *ONXX* pre-trained models that can be retrieved from the DSO following the abovementioned workflow, and all the components are connected through *ZeroMQ* using the PUB/SUB communication model. We run an initial set of experiments where five different models are requested from the DSO once the MSO identifies and notifies the anomaly. We depict the results in Table I. Results show that for "m-

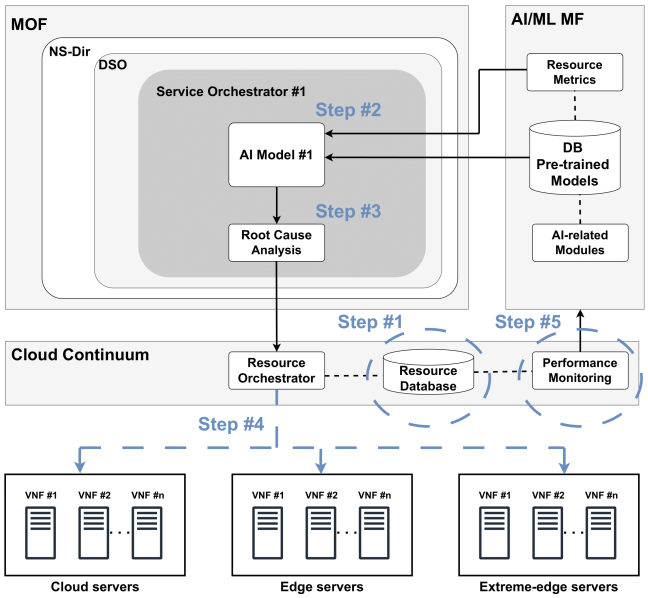

Fig. 4. Use case architecture illustrating each step of its functionality.

check" and "m-replacement" the the latency remains low, since they are communication points in software. In contrast, "m-request" incurs substantially higher latency due to the need to retrieve a typically large model. The last column shows the total latency accumulated across all checkpoints, with *resnet50* having the highest latency and *ResNet-preproc* the lowest. These findings support the effectiveness of our proposed approach, demonstrating that it introduces minimal overhead in the model communication process.

	Estimated Latency (ms)			
Model	m-check	m-request	m-replacement	Total
Model #1	0.000565	1.052840	0.001858	1.055262
Model #2	0.000602	0.389908	0.000853	0.391363
Model #3	0.000418	0.494493	0.000821	0.495732
Model #4	0.000388	0.496574	0.001025	0.497986
Model #5	0.000353	0.242890	0.000818	0.244061

TABLE I
ESTIMATED LATENCY OF FIVE DIFFERENT MODELS.

IV. AI-Driven Optimization in Decentralized Service Orchestration

The key benefits of decentralization for AI-powered service orchestration reside in the low-latency and real-time decision-making, scalability, flexibility and enhanced fault tolerance, resiliency and security. While a traditional centralized SO may introduce significant overhead (e.g. due to data aggregation or processing bottlenecks), the decentralized approach allows for distributed AI-driven operation that focuses on specific parts of the NS, significantly reducing the time needed (e.g. service scaling actions or failure detection). Moreover, enhanced scalability is intrinsically linked to decentralization, allowing the AI-powered SOs to scale better by enabling a better workload distribution. Decentralization also allows for self-operating SOs that can proactively detect anomalies either individually or in a collaborative manner and perform continuous learning, enabling model updates that may only affect part of the SOs within the DSO. Finally, fault tolerance, resiliency and security are widely enhanced given that failures may be contained locally (preventing cascading effects), and isolation among SOs also prevents sensitive data exposure while, at the same time, providing an improved framework toward complying with the multiple and non-homogeneous regulatory requirements.

Decentralized intelligent orchestration at the DSO is not only about localized intelligence but rather, it may require mechanisms for knowledge sharing within the DSO and in the overall system.

Before discussing the benefits of decentralized AI-driven service orchestration, it's important to clarify that only the inference stage is decentralized. The AIML Framework (AIMLF) remains centralized for tasks like data aggregation and model training, but it distributes trained models to local Service Orchestrators (SOs) within each Decentralized Service Orchestrator (DSO). This allows for localized, real-time decision-making at the edge, improving responsiveness and scalability. Knowledge sharing refers to the AIMLF sending AI insights and updates to DSOs and their SOs, maintaining coordination across the system.

A. Intra-DSO Knowledge Sharing

The different SOs conforming to a DSO may be responsible for orchestrating different virtual network functions within an NS without even sharing the optimization goals. However, information received by each of them may be vital for other SOs to achieve a non-envisioned optimization level. Hence, a knowledge-sharing model is required so that different AI models placed in SOs within a DSO can share information without affecting fault tolerance and security. Fortunately, there are AI/ML approaches that can be used toward trusted and secure knowledge sharing:

- Federated Learning (FL): Each SO runs its own AI model, which is locally trained, and instead of sharing raw data with a central entity, it shares model updates so that aggregated knowledge can be built and shared with other SOs to allow their models to improve [4].
- Reinforcement Learning (RL): This approach uses RL on each SO for optimization purposes but additionally, each SO shares reward signals or learned policies with other SOs to allow their RL agents to improve their operation [12].
- Graph Neural Network (GNNs): This approach represents the relationships among SOs as a graph, and the agents on each SO share knowledge (e.g. policies such as "*this orchestration action reduces latency*" with other SOs according to the defined relations [13].

- Convolutional Neural Networks (CNNs): CNNs can be adapted in orchestration scenarios to extract hierarchical features from time-series telemetry or multivariate network performance metrics. This enables SOs to detect e.g., patterns and anomalies in localized data streams [14].

B. Inter-DSO AI Knowledge Sharing

Intra-DSO knowledge sharing enhances the orchestration of a single domain by enabling collaboration among AI-driven SOs within a trusted environment. Extending this sharing across DSOs poses challenges due to anonymization requirements and limited applicability across domains, though similar approaches can be adapted to enable secure and trusted inter-DSO knowledge exchange.

- MOF Assets Repository: Pre-trained models from different SDOs can be stored in the Assets Repository (ARep) of the MOF, allowing other DSOs to reuse them.
- AI model exchange through transfer learning: Trained models could be reused in other DSOs, enabling the usage of models optimized in one domain into other related domains as black-box but with the ability to further fine-tune it during operation.
- Federated Learning (FL) between DSOs: This approach consists of sharing model updates instead of complete models, contributing to the generation of federated models that can also be stored in the ARep.

C. Challenges and Future Directions in Knowledge Sharing

Knowledge sharing enhances system performance by providing insights across different orchestration domains. However, it faces challenges such as privacy concerns and interoperability issues. One key challenge is the sensitivity of model training data, which could be leaked if malicious users gain access. Techniques like federated learning (FL) or differential privacy can mitigate this risk.

The diversity and lack of standardization in orchestration domains also hinder seamless knowledge exchange. To overcome this, defining exchange points for model sharing or using transfer learning to adapt models can improve compatibility. Additionally, managing decentralized models requires balancing inference time, accuracy, and optimal update intervals. Hierarchical deployment strategies or adaptive update schedules can help reduce the overhead.

Lastly, security and trust are essential in knowledge sharing. Models must be authenticated, and frameworks like zero-trust can ensure that shared models are reliable, preventing biased or compromised models from degrading system performance.

V. Conclusion

This paper introduces the DSO and its integration with the Management and Orchestration Framework (MOF) and AI/ML Framework (AIMLF). By decoupling service and resource orchestration, the framework enhances scalability, fault tolerance, and flexibility. AI/ML-driven decision-making supports intelligent resource allocation, anomaly detection, and real-time optimization. We show preliminary results in a use case which demonstrates an AI-powered self-healing mechanism for latency-aware VNF scaling. Finally, we discuss how AI-driven optimization is key in decentralized orchestration, enabling better decision-making, fault tolerance, and scalable workload distribution.

Acknowledgment

This work has been partly funded by the European Commission through the SNS JU project 6G-CLOUD (Grant Agreement no. 101139073).

References

[1] P. J. Denning and T. G. Lewis, "An ai learning hierarchy," *Commun. ACM*, vol. 67, no. 12, p. 24–27, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3699525

[2] R. Weingärtner, G. B. Bräscher, and C. B. Westphall, "A distributed autonomic management framework for cloud computing orchestration," in *2016 IEEE World Congress on Services (SERVICES)*, 2016, pp. 9–17.

[3] L. SMART, "Ai-powered container orchestration: Optimizing kubernetes workflows for efficiency," 2023.

[4] J. Péerez-Valero, A. Virdis, A. G. Sánchez, C. Ntogkas, P. Serrano, G. Landi, S. Kukliński, C. Morin, I. L. Pavón, and B. Sayadi, "Ai-driven orchestration for 6g networking: the hexa-x vision," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1335–1340.

[5] S. Kukliński, R. Kołakowski, L. Tomaszewski, L. Sanabria-Russo, C. Verikoukis, C.-T. Phan, L. Zanzi, F. Devoti, A. Ksentini, C. Tselios, G. Tsolis, and H. Chergui, "Monb5g: Ai/ml-capable distributed orchestration and management framework for network slices," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021, pp. 29–34.

[6] S. Kukliński, R. Kołakowski, L. Tomaszewski, L. Sanabria-Russo, C. Verikoukis, C.-T. Phan, L. Zanzi, F. Devoti, A. Ksentini, C. Tselios *et al.*, "Monb5g: Ai/ml-capable distributed orchestration and management framework for network slices," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. IEEE, 2021, pp. 29–34.

[7] A. Ullah, T. Kiss, J. Kovács, F. Tusa, J. Deslauriers, H. Dagdeviren, R. Arjun, and H. Hamzeh, "Orchestration in the cloud-to-things compute continuum: taxonomy, survey and future directions," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1–29, 2023.

[8] M. A. Habibi, A. G. Sánchez, I. L. Pavón, B. Han, P. Serrano, J. Pérez-Valero, A. Virdis, and H. D. Schotten, "The architectural design of service management and orchestration in 6g communication systems," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2023, pp. 1–2.

[9] C. Rotsos, D. King, A. Farshad, J. Bird, L. Fawcett, N. Georgalas, M. Gunkel, K. Shiomoto, A. Wang, A. Mauthe *et al.*, "Network service orchestration standardization: A technology survey," *Computer Standards & Interfaces*, vol. 54, pp. 203–215, 2017.

[10] F. Righetti, N. Tonellotto, N. Barsanti, and C. Vallati, "Energy-efficient orchestration strategies for function-as-a-service platforms," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2024, pp. 290–295.

[11] 3GPP, ""Management and orchestration; Artificial Intelligence/ Machine Learning (AI/ML) management (Release 17)"," 3GPP, Tech. Rep. TS 28.105, 2021.

[12] L. L. Schiavo, G. Garcia-Aviles, A. Garcia-Saavedra, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Cloudric: Open radio access network (o-ran) virtualization with shared heterogeneous computing," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 558–572.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[14] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, 2024.