The American Journal of Applied Sciences

ISSN 2689-0992 | Open Access

TYPE Original Research
PAGE NO. 65-72
DOI 10.37547/tajas/Volume07Issue09-09

Scalable Personalization in E-commerce Platforms: Balancing Customer Experience with System Complexity

Sathiya Veluswamy
Software Engineering Manager Seattle, Washington, USA

OPEN ACCESS

SUBMITED 08 August 2025
ACCEPTED 18 August 2025
PUBLISHED 28 September 2025
VOLUME Vol.07 Issue09 2025

CITATION

Sathiya Veluswamy. (2025). Scalable Personalization in E-commerce Platforms: Balancing Customer Experience with System Complexity. The American Journal of Applied Sciences, 7(09), 65–72. https://doi.org/10.37547/tajas/Volume07Issue09-09

Abstract: The article analyzes the strategic transition of a large-scale gift card platform for e-commerce from a monolithic architecture to a microservices model. The relevance of the study is driven by the need for established technology organizations to overcome constraints related to technical debt, such as slow development cycles and high keep-the-lights-on (KTLO) operational costs, in order to improve system scalability, reliability, and business agility. The objective of the study is to examine contemporary architectural approaches, algorithmic concepts, and operational models that can contribute to achieving an optimal balance between system flexibility and fault tolerance. The results obtained indicate that a phased migration strategy aimed at creating a standardized, configurable purchase management page in standard retail platform (DPX) is the most effective. This approach not only improved core technical metrics—such as achieving 99.99% availability and a significant reduction in KTLO, equivalent to 2.5 full-time employees per year—but also equipped stakeholders with tools for rapid layout updates without writing code. The presented findings will be useful to chief technology officers, heads of engineering, and software architects managing the modernization of business-critical legacy systems.

Keywords: e-commerce, personalization, scalability, microservice architecture, customer experience, agile development, recommendation systems, high load, system complexity.

Introduction

The current stage of the digital economy's development is characterized by the unprecedented expansion of the e-commerce segment. The volume of the digital commerce market in 2024 is estimated at 5.5 trillion USD, and it is expected to reach 12 trillion USD by 2033, with an average annual growth rate of 9.5% during the period from 2026 to 2033 [1]. The intensifying competition in this domain creates a critical dependence of platform success on the ability to provide a highly integrated and personalized customer experience (Customer Experience, CX). One of the most effective means of achieving such a level of engagement is the adaptation of content and offerings to individual preferences, behavioral patterns, and the user's current context in real time.

The necessity of personalization is supported by extensive empirical evidence. Research shows that personalization most often leads to a revenue increase of 10–15% (although depending on the industry and the company's capabilities, growth may range from 5 to 25%). The better a company is at using data to increase its knowledge of customers and strengthen its connection with them, the higher its profits. For companies that are digitally native and operate based on a data-driven direct-to-consumer model, personalization is not merely a promotional tool but the foundation of operations [2]. Accordingly, mechanisms such as recommendation systems, dynamic content generation, and adaptive user interfaces are shifting from the category of competitive advantages to fundamental requirements for audience retention and increasing customer lifetime value.

Nevertheless, the creation and operation of an infrastructure capable of serving millions of users in real time is associated with growing systems and computational complexity. The processing of petabyte-scale data volumes, execution of resource-intensive machine learning algorithms, and provision of high reliability and fault tolerance of the architecture require significant resources. The transition from elementary segmentation rules to hybrid neural network models leads to an exponential increase in workloads, growth in stored data volumes, and complications in inter-component interactions.

As a result, a critical challenge arises when legacy monolithic architectures, common in established e-commerce platforms, become a bottleneck to innovation. These systems, often burdened with significant technical debt, lead to slow development cycles, high operational costs (often referred to as "Keep The Lights On" or KTLO), and an increased risk of failures during deployment. The inability to quickly iterate on the customer experience (CX)—whether for international expansion, A/B testing, or introducing new features—directly hinders business growth and competitiveness. In extreme cases, the architectural rigidity nullifies any attempts at personalization, making the platform difficult to adapt and vulnerable to change.

Academic research in the field of recommender systems has traditionally focused on the optimization of algorithmic approaches—from classical collaborative filtering to deep neural networks [3, 4]. In parallel, a line of research has emerged exploring scalable architectural solutions: microservice structures and serverless computing [5]. However, existing literature still lacks a systematic, holistic examination of how the level of content personalization correlates with architectural patterns and operational models in large-scale e-commerce ecosystems.

The aim of this study is to conduct a comprehensive analysis of existing strategies and architectural frameworks that allow for balancing the depth of customer experience personalization with the manageability of technological complexity in e-commerce platforms.

The scientific novelty of this work lies in the proposal of an integrated methodological framework that connects the selection of personalization algorithms with specific scalable architectural patterns and levels of operational process maturity.

The author’s hypothesis suggests that the application of a federated microservice architecture, in combination with hybrid recommender models and an agile operational culture focused on enhancing the customer journey, constitutes the optimal compromise between the level of personalization, system scalability, and its long-term maintainability.

Materials and Methods

In recent years the topic of scalable personalization in electronic commerce has attracted increasing attention, driven both by the rapid growth of the global digital commerce market and by rising user demands for an individualized shopping experience. According to a report by Verified Market Reports, the volume of the global digital commerce market is rapidly expanding across all business models and geographic regions [1], and a McKinsey study emphasizes that proper personalization tuning can multiply the value of customer engagement, whereas errors in this process lead to serious losses both financial and reputational [2].

A key research direction comprises algorithmic personalization methods, among which classical approaches based on collaborative filtering harmoniously blend with more modern deep learning techniques and graph models. Thus, Ahmed E., Letta A. [4] apply collaborative filtering for recommendations, demonstrating that even basic models can achieve high accuracy with appropriate data preprocessing and consideration of the specificity of product categories. Boka T. F., Niu Z., Neupane R. B. [6] in their review sequentially examine methods of recurrent neural networks, Markov chains and hybrid models for constructing sequential recommendations, clearly highlighting the “cold start” problem and the need to balance between the recency and the relevance of user sessions. Gao C. et al. [3] demonstrate the advantages of graph neural networks (GNN) for recommendations, allowing complex relationships between users and products to be taken into account, which improves recommendation quality in sparse data and contributes to more flexible contextualization of interactions. Finally, Frey J., Ferraz L., Hofer M. [9] propose the Pots approach, based on meta-ontologies and hyper-level vector spaces, which provides fine-grained management of term context and can augment recommendation models, particularly in semantic search tasks for products with rich descriptions.

Alongside algorithmic improvements, the issue of constructing reliable and fault-tolerant architectures capable of scaling under increasing load and ensuring low response latency remains crucial. Bushong V. et al. [5] in a systematic review analyze the evolution of microservice architectures, emphasizing that transitioning to microservices requires a carefully devised strategy for versioning, transaction management and orchestration, otherwise system complexity may grow exponentially. Kjorveziroski V., Filiposka S., Trajkovic V. [7] consider serverless edge computing, identifying open challenges in resource balancing and distribution of computing functions between the cloud and client devices, which is especially relevant for reducing latency of real-time personalized offers. Moreover, Lekkala C. [10] describes principles for building high-performance Kafka clusters for streaming processing of large volumes of transactional data, a key element of low-latency recommendation systems ensuring consistency of event stream state.

An equally important aspect in implementing personalization is ensuring user privacy and regulatory compliance. Gotsch M. L., Schögel M. [8] in their review of organizational strategies for overcoming the “privacy paradox” show that companies often face a contradiction between the desire to collect maximum data for personalization and the necessity to protect user information, proposing hybrid data management models and internal policies capable of aligning these interests .

Additionally, a significant body of research in Human-Computer Interaction (HCI) and web development highlights the critical role of the user interface in realizing the full value of personalization. Studies emphasize that factors such as page load speed, adaptive layouts for different devices, and the seamless integration of personalized elements into the user flow directly impact engagement and conversion rates. The technical implementation of the front-end, therefore, is not merely a delivery mechanism but a crucial component of the overall personalization strategy, responsible for translating data-driven insights into a tangible and positive customer experience.

Thus it can be observed that various research directions emphasize the contradiction between the potential benefits of deep personalization and the high complexity of both algorithmic solutions and infrastructure components. On the one hand, modern methods (GNN, sequential models, semantic ontologies) enable significant improvement in recommendation quality, but they require substantial computational resources and complicate the system. On the other hand, architectural patterns (microservices, edge computing, streaming) provide scalability and fault tolerance, yet their integration with personalization algorithms remains nontrivial and requires further investigation. Moreover, tension persists between the effectiveness of personalization and the protection of privacy, reflecting insufficiently developed methods for ensuring transparency and control over user data. Among the least explored areas are cross-channel personalization accounting for heterogeneous behavior sources, online model training in conditions of high-frequency assortment and demand changes, as well as combining edge computing with centralized personalization algorithms to optimize latency and network load.

Results and Discussion

Achieving deep and scalable personalization requires a comprehensive consideration of three fundamental aspects: structural design, algorithm selection methodologies, and the maturity level of operational processes.

The key element of any large-scale e-commerce platform is its system architecture. When using the classical monolithic model, in which all business modules are tightly integrated into a single codebase, bottlenecks quickly emerge. An increase in load and functional complexity leads to the fact that implementing new personalization mechanisms becomes extremely labor-intensive. Any modification of the recommendation mechanism requires a complete rebuild and redeployment of the entire application, which significantly slows down the innovation cycle and increases the likelihood of deployment errors.

As a solution, the transition to the microservices paradigm is increasingly practiced, where the system is divided into autonomous services, each focused on a specific business function. In the context of personalization, these can include the user profile management service, the product catalog service, and, most critically, the recommendation service. This decomposition makes it possible to scale the Recommendation Service independently of the overall platform and to apply a specialized technology stack to it (for example, Python packages for machine learning, while other services are maintained in Java) [5, 9].

Nevertheless, the division into microservices by itself does not guarantee the required level of flexibility. To avoid fragmentation, it is advisable to implement a federated model of microservice architecture (see Figure 1), in which a set of dedicated domain services is combined under unified logical control while maintaining deployment autonomy. This approach ensures both data consistency and independent scaling.

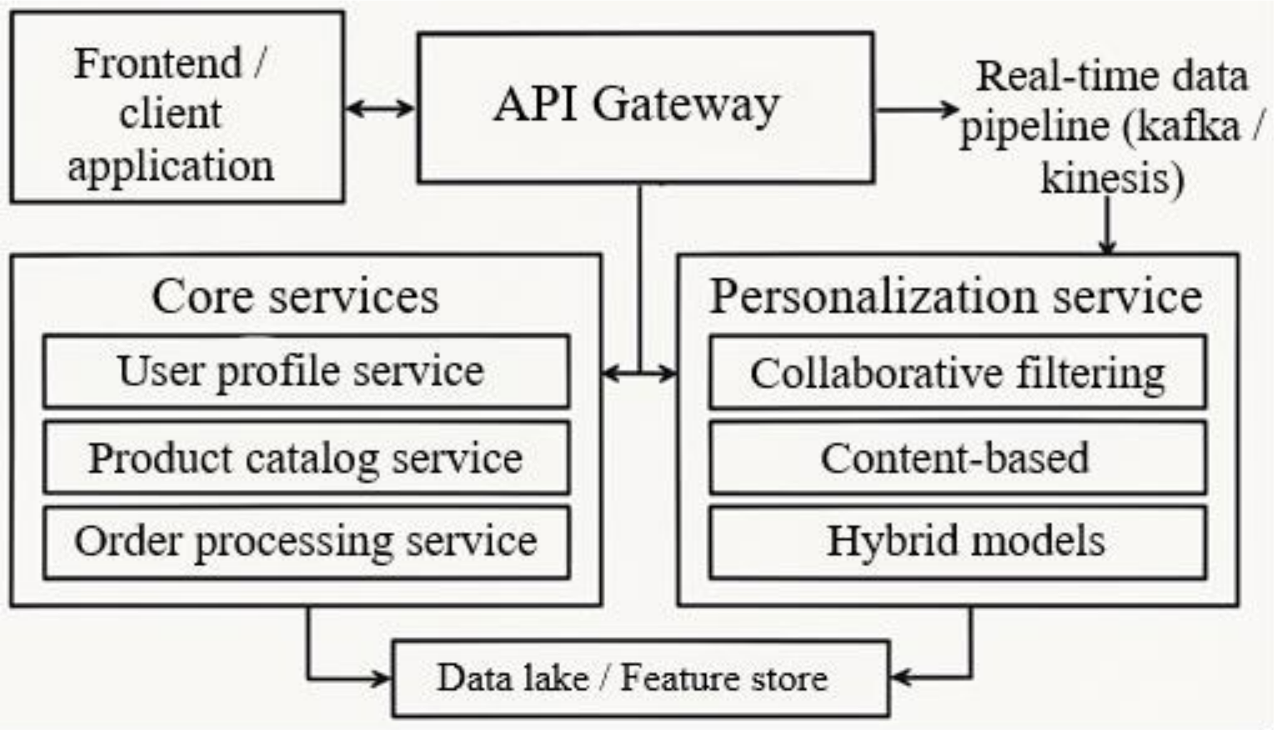

Fig.1. Federated microservice architecture for personalization [5, 9]

The key components of the proposed architecture are designed to ensure flexibility, fault tolerance, and the ability to independently scale individual modules:

API Gateway. Serves as a unified front access point for all external requests. In addition to routing calls to the corresponding microservices, it is capable of centrally performing authentication, managing response caching, and rate limiting, thereby preventing system overload.

Frontend/Client Application. Represents the user interface (whether a web portal or a mobile application) that interacts with the server layer exclusively through the API Gateway. This separation allows the client side and backend to evolve independently, minimizing mutual dependencies.

Core Services. Contain business-critical functional blocks: user profile management, product catalog, order processing, and other fundamental subsystems. Each of these services is deployed separately and has its own data store, which increases system resilience in case of a module failure.

Personalization Service. A specialized service responsible for generating personalized recommendations for users. Due to its modular structure, it is divided into subservices focused on different algorithmic approaches (collaborative filtering, content-based models, hybrid methods), which facilitates the addition of new strategies without affecting the rest of the logic.

Real-time Data Pipeline. An asynchronous event processing stream (e.g., click tracker, views, adding items to the cart) implemented using systems such as Apache Kafka or AWS Kinesis. This pipeline collects and delivers user action data to the personalization service for rapid model retraining and real-time adaptation of recommendations [6, 10].

Data Lake / Feature Store. A unified storage that accumulates both raw event data and precomputed features used in the training and inference of ML models. The presence of a single source of truth enhances the reproducibility of experiments and simplifies research activities.

As a result, such an organization of components ensures strict separation of concerns: the business logic core remains conservative and reliable, while the personalization block can dynamically expand and be modified without the risk of unintended consequences for the core services. This directly contributes to the reduction of technical debt and accelerates the release of new functional capabilities.

The practical application of the described architectural principles can be demonstrated through a case study of a large-scale modernization project. The project focused on three key strategic initiatives to transform the platform and deprecate monolithic system:

Initiative 1: Phased Migration of the Core Purchase Journey. The primary initiative involved the migration of the entire electronic gift card purchase journey from the monolith to a microservice architecture based on Java and AWS. This high-risk, high-impact project entailed shifting approximately 17 million daily customer requests from one technology stack to another while adhering to a strict service level objective of 99.99% availability. The migration was executed in carefully managed phases to minimize risk and ensure a seamless customer experience across 22 global marketplaces. A key outcome of this phase was the successful transition and stabilization of the new architecture, which now manages a traffic volume that grew from an initial 10 million to 17 million daily hits over a three-year period.

Initiative 2: Implementation of a Configurable, Experiment-Friendly CX Platform. To address the inherent rigidity of the monolith, where UI or layout changes required significant engineering overhead, a second initiative focused on migrating gift card detail pages to a standardized platform known as DPX (Detail Page Experience). This platform provided business and product stakeholders with a configuration tool, "Themis," which enabled the creation and modification of page layouts without direct engineering intervention. This initiative proved transformative for processes such as international expansion and marketing campaigns. The direct result was a quantifiable reduction in "Keep The Lights On" (KTLO) operational costs, equivalent to the effort of 2.5 full-time employees annually.

Initiative 3: Architecturally-Enabled Business CX Improvements. The new architectural flexibility enabled subsequent business-driven enhancements to the customer experience. In partnership with product stakeholders, the gift card discovery journey was redesigned. Instead of a single, monolithic detail page, the system was reconfigured to present categorized pages based on specific occasions (e.g., "Birthdays," "Holidays"). This change, made feasible by the decoupled architecture, led to a measurable increase in customer conversion speed. Furthermore, the project introduced new ingress points for gift cards on the platform, which generated an additional ~42k clicks per day per marketplace, thereby increasing product visibility to new customer segments.

When selecting methods for generating recommendations, four key criteria inevitably arise: prediction accuracy, computational cost, required data volume, and model transparency. There is no universally optimal algorithm; therefore, Table 1 provides a comparison of the main classes of methods according to the aforementioned parameters.

Table 1. Comparative Analysis of Personalization Algorithms [7, 8, 10]

Algorithm	**Accuracy**	**Scalability (Inference)**	**Data Requirements**	**Cold Start**	**Interpretability**
Collaborative Filtering (CF)	Medium	High	Interaction data only	Problematic	Low
Content-Based Analysis	Low-Medium	High	Item attributes	Solves	High
Hybrid Models	High	Medium-High	Interactions and attributes	Partially solves	Medium
Deep Learning (DNN, GNN)	Very High	Low-Medium	Large volumes of data	Problematic	Very Low

As the analysis shows, models based on deep learning methods demonstrate the highest accuracy; however, their use in inference mode is associated with critical computational resource costs and the necessity of working with colossal training datasets. This makes their implementation for real-time personalization for each user economically unjustified [3, 6].

As a solution, it is advisable to use a multi-level hybrid recommendation architecture, structurally presented in Figure 2.

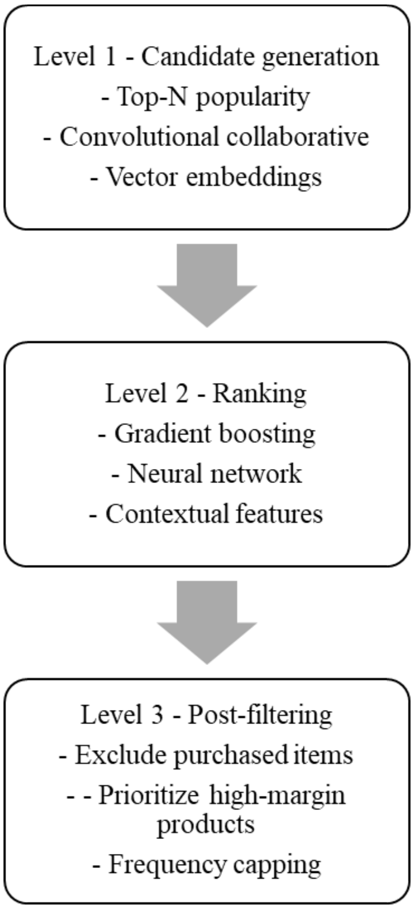

Fig.2. General scheme of the multi-level hybrid recommendation model [4, 7, 9]

Let us consider in more detail the levels reflected in Figure 2:

Level 1 – Candidate Generation. At this stage, the fastest and most easily scalable methods are employed: basic heuristics (e.g., top-N by popularity), convolutional collaborative filtering mechanisms, or models based on vector representations of users and items. The objective is to select from the entire thousand-item catalog of gift cards several hundred options that are most likely to interest the user.

Level 2 – Ranking. The resulting pool undergoes thorough analysis using more sophisticated and accurate algorithms—for example, gradient boosting or a compact neural network. Beyond basic features (user and item identifiers), the model accounts for additional contextual factors: temporal characteristics (hour, day of the week), demographics, session history, current events (e.g., upcoming holidays), etc. The output is a precisely ordered top-N list.

Level 3 – Post-filtering. At the final stage, corporate business rules are applied: exclusion of recently viewed or purchased items to diversify the results, prioritized display of high-margin or newly added products, adherence to constraints on impression frequency (frequency capping), and so on.

A key, though often underestimated, component for the successful integration of personalization is the operational model and the maturity of the development lifecycle. The goal is to create a tight feedback loop between data insights, feature development, and the end-user's reaction, which is a cornerstone of modern web development and customer experience management[10].

To achieve this, companies adopt Agile methodologies. These practices are crucial for several reasons:

- Faster Time-to-Market for CX Improvements. Automated Continuous Integration and Continuous Deployment (CI/CD) pipelines for both backend services and frontend applications significantly reduce the time from ideation to launching a new personalization feature or UI enhancement. This allows teams to quickly iterate based on user feedback.
- Proactive Monitoring of User Experience. Beyond traditional technical metrics (CPU, memory), the focus shifts to business and CX-oriented metrics. This includes monitoring conversion rates, click-through rates on recommendations, and user session duration. A/B testing platforms are integrated directly into the deployment pipeline, allowing for data-driven decisions on which features truly improve the customer experience.

When machine learning models are a part of this lifecycle, these principles extend into the domain known as MLOps, which addresses specific challenges like data versioning and model drift monitoring [5, 7].

For these principles to function effectively, an appropriate organizational structure is required. Cross-functional teams are necessary, including not only ML and data engineers but also backend and frontend developers, UI/UX designers, and product managers. This composition fosters deeper knowledge exchange and shared responsibility for the end-to-end customer journey—from the data pipeline to the final pixel rendered in the user's browser.

It is precisely the harmonious combination of architectural patterns, advanced algorithms, and a well-designed operational model that lays the foundation for a sustainable balance between exceptional user experience and long-term operational reliability of the technology platform.

Conclusion

This study demonstrates that modernizing a high-load e-commerce platform by migrating from a legacy monolith to a federated microservice architecture is a critical strategy for sustainable growth. The success of such a transformation hinges not on implementing complex algorithms, but on a disciplined, phased approach to architectural change. The key findings confirm that this approach provides a robust foundation for scalability, achieving 99.99% availability for core services serving over 17 million daily users. It also yields significant operational efficiencies, drastically reducing KTLO costs and empowering business teams with configurable tools that accelerate time-to-market for new customer experiences. The practical value of this work lies in the presented case study, which serves as a roadmap for technology leaders undertaking similar modernization efforts. A focus on architectural fundamentals over premature algorithmic complexity builds a resilient, adaptable platform prepared for future business challenges.

For future research, it is advisable to focus on several directions: first, developing quantitative models to measure the impact of architectural modernization on developer velocity and business agility; second, exploring best practices for managing data consistency and migration in a phased decomposition of a monolith; and third, investigating the interplay between standardized backend platforms and modern frontend frameworks in delivering a seamless, configurable user experience at scale.

References

1. Verified Market Reports. (2025). Global Digital Commerce Market Size By Business Model (B2B (Business-to-Business), B2C (Business-to-Consumer)), By Product Type (Physical Goods, Digital Goods), By Payment Method (Credit and Debit Cards, Digital Wallets), By Customer Demographics (Age Group, Income Level), By Distribution Channel (Online Sales, Offline Sales (Pharmacies and Retailers)), By Geographic Scope And Forecast. Retrieved June 25, 2025, from https://www.verifiedmarketreports.com/product/digital-commerce-market/
2. McKinsey & Company. (2021). The value of getting personalization right—or wrong—is multiplying. Retrieved June 15, 2025, from https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/the-value-of-getting-personalization-right-or-wrong-is-multiplying
3. Gao, C., et al. (2022). Graph Neural Networks for Recommender System. Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 1623–1625. https://doi.org/10.1145/3488560.3501396
4. Ahmed, E., & Letta, A. (2023). Book Recommendation Using Collaborative Filtering Algorithm. Applied Computational Intelligence and Soft Computing, 1–12. https://doi.org/10.1155/2023/1514801
5. Bushong, V., Abdelfattah, A. S., Maruf, A. A., Das, D., Lehman, A., Jaroszewski, E., Coffey, M., Cerny, T., Frajtak, K., Tisnovsky, P., & Bures, M. (2021). On Microservice Analysis and Architecture Evolution: A Systematic Mapping Study. Applied Sciences, 11(17), 7856. https://doi.org/10.3390/app11177856
6. Boka, T. F., Niu, Z., & Neupane, R. B. (2024). A survey of sequential recommendation systems: Techniques, evaluation, and future directions. Information Systems, 125, https://doi.org/10.1016/j.is.2024.102427
7. Kjorveziroski, V., Canto, C. B., Roig, P. J., Gilly, K., Mishev, A., Trajkovik, V., & Filiposka, S. (2021). IoT Serverless Computing at the Edge: Open Issues and Research Direction. Transactions on Networks and Communications, 9(45), 1–33.
8. Gotsch, M. L., & Schögel, M. (2021). Addressing the privacy paradox on the organizational level: Review and future directions. Management Review Quarterly, 73, 263-296.
9. Frey, J., Ferraz, L., & Hofer, M. (2025). POTS—A Polyparadigmatic Ontology Term Search with Fine-Grained Context Steering using Hyper-Llevel Vector Spaces. WWW 25: Companion Proceedings of the ACM on Web Conference 2025, 2831-2834.
10. Lekkala, C. (2021). Designing High-performance, Scalable Kafka Clusters for Real-time Data Streaming. European Journal of Advances in Engineering and Technology, 8(1), 76-82.