www.ijarp.com ISSN 2456-9992 Page: 01-13

International Journal Advanced Research Publications

Research Article

Volume: 02
Issue: 04

AN EVENT-DRIVEN BAKERY OPERATIONS PLATFORM INTEGRATING ORDER LIFECYCLE, KITCHEN EXECUTION, DELIVERY HANDOFF, AND POLICY-CONTROLLED OPERATIONS MANAGEMENT

***Sachin Gupta, Priyanshu Patel, Sonam Rohilla**

Department of Computer Science & Engineering (Artificial Intelligence) IIMT College of Engineering, Knowledge Park 3, Greater Noida, Uttar Pradesh, India.

Article Received: 16 March 2026, Article Revised: 06 April 2026, Published on: 26 April 2026

***Corresponding Author: Sachin Gupta**

Department of Computer Science & Engineering (Artificial Intelligence) IIMT College of Engineering, Knowledge Park 3, Greater Noida, Uttar Pradesh, India.

DOI: https://doi-doi.org/101555/ijarp.9083

ABSTRACT

Bakery operations operate within a highly time-sensitive environment influenced by perishable inventory, fluc-tuating customer demand, and the simultaneous handling of walk-in orders, scheduled pickups, customized production, and delivery processes. Despite the growing adoption of digital solutions, many small and medium-scale bakeries continue to rely on fragmented systems for order management, kitchen coordination, customer interaction, and administrative control. This lack of integration often results in delayed synchronization, limited visibility of order states, workflow inefficiencies, and reduced traceability of operational activities.

This paper proposes an event-driven bakery operations platform that integrates order processing, kitchen execution, delivery coordination, loyalty management, and policy-driven administrative control within a unified architecture. The system is developed using the MERN stack and employs Socket.io to enable real-time communication and dynamic workflow updates. The proposed design separates user interaction, transaction processing, and event handling into distinct layers to enhance scalability, maintainability, and system responsiveness.

Furthermore, the platform incorporates role-based access control, audit-enabled governance, and a structured lifecycle model to enforce valid order transitions and improve operational

clarity. Performance evaluation through simulated workflows demonstrates improved synchronization, reduced latency, enhanced visibility, and reliable event propagation. The results indicate that an event-driven approach provides an effective and scalable solution for modern bakery operations by replacing disconnected systems with a cohesive real-time coordination framework.

***INDEX TERMS*:** Event-driven architecture, bakery operations, kitchen display system, order lifecycle, MERN stack, real-time systems, role-based access control, workflow orchestration.

I. INTRODUCTION

Bakery operations constitute a hybrid environment that combines elements of retail service, production workflows, delivery coordination, and customer relationship management. Unlike traditional restaurant systems, bakery businesses often handle multiple parallel processes, including walk-in purchases, advance order scheduling, customized product preparation, batch production, packaging, and delivery logistics. The coexistence of these activities introduces significant complexity, making efficient coordination, real-time visibility, and accurate timing essential for maintaining operational efficiency and service quality.

In many real-world bakery settings, digital infrastructure remains fragmented. Order entry, kitchen preparation, and delivery tracking are frequently managed through separate tools or semi-manual processes. As a result, there is often a disconnect between order confirmation and actual execution within the kitchen and dispatch systems. During peak hours, such fragmentation can lead to delayed updates, repeated manual verification, miscommunication between staff roles, and increased likelihood of operational errors. These challenges ultimately reduce system reliability and hinder effective decision-making.

To address these limitations, modern distributed systems increasingly adopt event-driven architectures that enable realtime coordination between system components [1], [2]. Instead of relying on continuous polling mechanisms, event-driven systems propagate state changes instantly to relevant subscribers, thereby improving responsiveness and reducing system overhead [3], [4]. In the context of bakery operations, such an approach is particularly beneficial, as it ensures that order status, preparation progress, and dispatch readiness are consistently synchronized across all stakeholders.

This paper presents an event-driven bakery operations platform designed to explicitly model order progression and deliver real-time updates across system interfaces. The proposed solution integrates point-of-sale operations, kitchen workflow management, readiness

validation, delivery coordination, customer loyalty mechanisms, and administrative governance within a unified full-stack architecture. By structuring the system into distinct layers for user interaction, transaction processing, and event handling, the design enhances modularity, scalability, and maintainability.

The primary contributions of this work are summarized as follows:

- Development of a bakery-specific event-driven architecture organized into experience, transaction, and event layers.
- Design of a structured order lifecycle model that enforces valid state transitions and prevents workflow inconsistencies.
- Integration of role-based access control and audit-oriented governance for improved system security and traceability.
- Evaluation of system performance demonstrating the benefits of real-time synchronization in reducing delays and improving operational visibility.

The remainder of this paper is organized as follows. Section II discusses related work and identifies existing research gaps. Section III outlines the research methodology. Section IV presents system analysis and requirement specifications. Section V describes the proposed architecture and implementation details. Section VI provides experimental evaluation results. Section VII discusses findings, limitations, and future research directions. Finally, Section VIII concludes the paper.

II. Related Work and Research Gap Analysis

A. Digital Transformation in Bakery and Food-Service Operations

The adoption of digital technologies in small and medium-sized food-service businesses has increased significantly in recent years. These systems support order management, production planning, and customer engagement. However, the presence of digital tools alone does not ensure improved operational performance. When systems operate in isolation, issues such as inconsistent information flow, redundant tasks, and lack of coordination across operational roles persist. Prior studies indicate that effective workflow management depends not only on automation but also on the timely propagation of state changes and consistent system-wide synchronization [5], [6].

B. Event-Driven Architecture and Real-Time Coordination

Event-driven architectures have gained prominence in distributed computing environments

where rapid responsiveness is required. Instead of relying on continuous polling, such systems notify relevant components immediately when state changes occur. This mechanism enhances responsiveness and reduces system overhead while promoting loose coupling between components [1], [3]. Additionally, this architectural approach improves scalability and extensibility, making it suitable for applications that require real-time coordination. In retail and service domains, event-driven communication has proven effective in maintaining order visibility, enabling instant notifications, and ensuring process consistency [7], [8], [9].

C. Kitchen Display Systems and Workflow Orchestration

Kitchen Display Systems (KDS) have evolved from basic digital order displays into advanced coordination tools that support task prioritization, preparation tracking, and completion monitoring.

These systems enhance communication

between front-end ordering interfaces and kitchen operations. However, most existing implementations are tailored toward conventional restaurant environments. Bakery workflows, which often involve batch processing, customized production, and delayed pickup or delivery, require more flexible and specialized coordination mechanisms. This highlights the need for workflow models that accurately represent bakery-specific operational processes and state transitions [10].

D. Governance, RBAC, and Auditability

Role-Based Access Control (RBAC) is widely used to regulate access to system resources and enforce operational security [11], [12]. While RBAC effectively restricts unauthorized actions, modern systems require additional mechanisms to ensure accountability. Administrative configurations, permission changes, and exceptional actions must be traceable to maintain system integrity. Research in enterprise systems suggests that combining RBAC with audit logging and role-aware governance significantly improves transparency and control in operational environments [13], [14].

E. Research Gap

Although existing commercial platforms provide strong capabilities in specific areas such as point-of-sale systems, inventory management, or customer interaction, they often lack an integrated and transparent architectural framework. In particular, there is limited focus on combining bakery-specific lifecycle modeling, real-time event-driven communication, and

policy-based governance within a single unified system. This research addresses this gap by proposing and evaluating a cohesive event-driven platform tailored specifically for bakery operations, emphasizing real-time coordination, structured workflow management, and comprehensive governance.

III. Research Methodology

This study adopts the Design Science Research (DSR) paradigm, which is well suited for developing and evaluating artifacts intended to solve practical problems [15], [16]. The primary objective of this approach is to design a functional software system while ensuring that the solution is grounded in established principles of software engineering, system coordination, and enterprise-level control mechanisms.

The research process was carried out in a structured sequence of stages. Initially, the problem domain was analyzed by examining the limitations of existing bakery workflows, particularly the issues arising from fragmented systems and lack of real-time coordination. Based on this analysis, both functional and non-functional requirements were defined to address the needs of order processing, kitchen operations, and delivery management.

Following requirement identification, a prototype system was designed and implemented using modern web-based technologies that support real-time communication and scalable system behavior. The architecture was structured to separate user interaction, transaction processing, and event handling, enabling efficient coordination across system components.

Subsequently, the developed prototype was evaluated through controlled simulations representing typical bakery operations. These simulations included scenarios such as order placement, preparation updates, and delivery handoff processes. System performance was analyzed using quantitative metrics, while workflow effectiveness was assessed through role-based interaction scenarios.

The evaluation approach incorporated both technical and operational perspectives. From a technical standpoint, key performance indicators such as API response time, database transaction latency, WebSocket communication delay, and overall system resource utilization were measured under simulated load conditions. From a workflow perspective, the study examined how effectively the platform provided real-time visibility of order status, improved coordination between roles, and minimized ambiguity in operational handoffs through the use of a structured lifecycle model.

IV. System Analysis and Requirements

A. Stakeholders

The proposed platform is designed to support a diverse set of operational stakeholders, each interacting with the system according to their specific responsibilities:

- **Customer:** Interacts with the system to browse available products, place orders, apply promotional offers, and monitor the status of ongoing orders.
- **Operations Administrator:** Responsible for managing the product catalog, configuring system settings, assigning user roles, and maintaining operational traceability.
- **Kitchen Preparation Staff:** Handles order execution by receiving assigned tasks, confirming order acceptance, and updating preparation progress in real time.
- **Quality-Control Staff:** Ensures that prepared items meet quality standards and confirms order readiness before dispatch or pickup.
- **Dispatch Staff:** Oversees the final stage of order fulfillment by coordinating delivery or customer pickup once the order is marked ready.
- **Managerial User:** Monitors overall system performance, reviews operational reports, and evaluates workflow efficiency and governance compliance.

B. Functional Requirements

The system is required to provide a unified and integrated workflow that supports the complete lifecycle of an order. Core functionalities include centralized order ingestion from multiple channels, real-time tracking of order progression, and enforcement of valid state transitions through a structured lifecycle model. Additionally, the platform must support atomic processing of coupon and loyalty mechanisms to ensure consistency in pricing and rewards.

The system should also implement role-based visibility, ensuring that users can access only the information and actions relevant to their responsibilities. Administrative operations must be traceable to support accountability and system integrity.

C. Non-Functional Requirements

In addition to functional capabilities, the platform must satisfy several critical non-functional requirements. These include maintaining low latency for real-time updates, ensuring system stability under varying load conditions, and providing reliable audit mechanisms for tracking critical operations. The system should also be resilient to partial failures, ensuring that non-essential component failures do not disrupt core operational workflows.

These requirements are particularly important in bakery environments, where timely information updates and consistent interpretation of order status are essential for efficient coordination and service delivery.

V. Proposed Architecture and Implementation

The proposed bakery operations platform is structured into three primary architectural layers, namely the Experience Plane, the Transaction Plane, and the Event Plane. This layered design separates user interaction, core business logic, and asynchronous communication, enabling efficient coordination while maintaining system modularity and scalability.

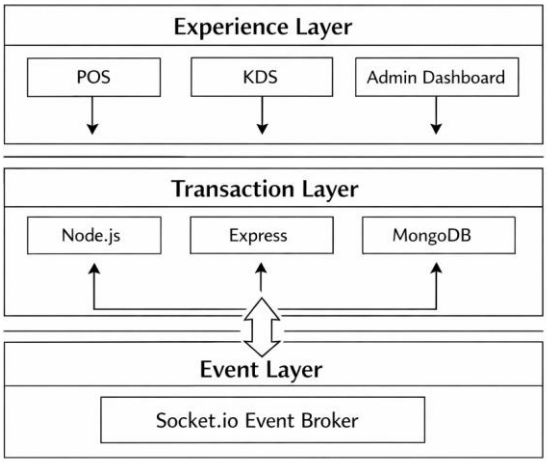

Fig. 1: Overall event-driven bakery operations architecture illustrating the experience, transaction, and event planes.

The *Experience Plane* represents the front-end layer through which different stakeholders interact with the system. It includes interfaces for customers, administrative staff, kitchen personnel, dispatch operators, and managerial users. Each interface is tailored to provide role-specific functionality, ensuring that users access only the relevant information and actions required for their tasks.

The *Transaction Plane* acts as the core processing layer of the system. It is responsible for validating incoming requests, managing order creation, maintaining system state, handling coupon and loyalty logic, and executing business rules. This layer ensures data consistency and enforces application-level constraints before any state changes are committed to the database.

The *Event Plane* facilitates real-time communication across system components. It propagates state changes using an event-driven mechanism implemented through Socket.io.

By broadcasting updates to subscribed interfaces, this layer eliminates the need for continuous polling and significantly improves the responsiveness and visibility of system operations. A key component of the proposed architecture is the lifecycle management engine, which governs the progression of orders through predefined states. Unlike loosely managed status flags, the system enforces a structured sequence of transitions to ensure consistency and traceability.

The order lifecycle is defined as a sequence of states, namely: *PENDING*, *APPROVED*, *PREPARING*, *READY*, *DISPATCHED*, and *DELIVERED*.

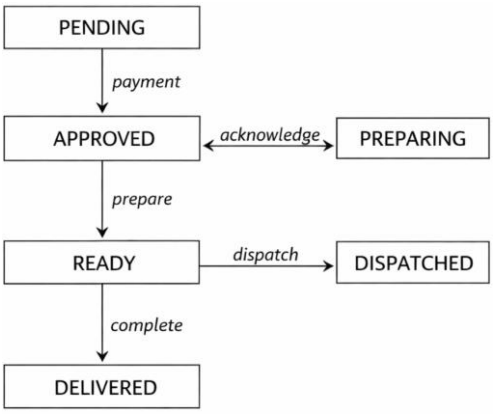

Fig. 2: Order lifecycle state machine representing valid transitions in bakery operations.

An order initially enters the *PENDING* state and transitions to *APPROVED* after successful validation of payment and associated transaction logic. Once acknowledged by the kitchen, the order moves to the *PREPARING* state. Upon completion of preparation and quality verification, the order is marked as *READY*. The assignment of delivery or pickup handling moves the order to the *DISPATCHED* state, and the final confirmation of fulfillment results in the *DELIVERED* state. Any attempt to bypass or violate these transitions is rejected and recorded, ensuring process integrity.

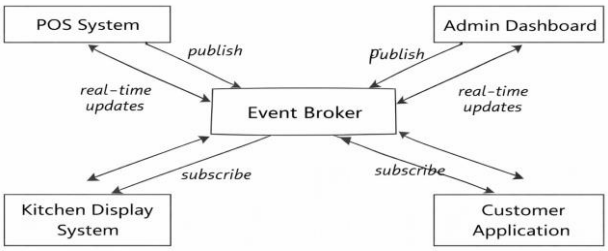

Fig. 3: Event-driven communication flow enabling real-time synchronization between system components.

In addition to lifecycle management, the platform integrates coupon validation and loyalty processing directly within the transaction layer. This ensures that pricing adjustments and reward mechanisms are applied consistently during order processing. Governance mechanisms are implemented through role-based access control, route-level authorization, and audit logging of critical administrative actions. These features enhance system security, improve accountability, and reduce ambiguity in user permissions.

VI. Experimental Setup and Evaluation

The proposed system was evaluated through a series of simulated bakery operation scenarios, including order placement, kitchen acknowledgment, preparation updates, and final dispatch handling. The primary objective of this evaluation was to analyze whether the platform enhances coordination across different operational roles and reduces ambiguity in order tracking.

From a technical standpoint, the system was tested under both normal and high-load conditions to assess its performance and stability. Key performance indicators included average API response time, database transaction latency, WebSocket-based event propagation delay, and overall server resource utilization.

TABLE I: Quantitative performance metrics under simulated load.

Performance Metric	Nominal Load	Peak Load	Target Threshold
REST API response time (avg.)	124 ms	385 ms	< 500 ms
Database transaction latency	45 ms	112 ms	< 200 ms
WebSocket event latency	28 ms	86 ms	< 150 ms
CPU utilization	14%	78%	< 85%

In addition to technical evaluation, a workflow-oriented comparison was conducted between the proposed system and a fragmented baseline setup. The baseline represents a conventional environment where order updates are partially manual or delayed across different stages of operation.

TABLE II: Operational workflow comparison between fragmented baseline and proposed platform.

Metric	Baseline	Proposed Platform
Order visibility delay	145 s	0 s
Routing error rate	8%	0%
State synchronization	Manual / delayed	Real-time
Kitchen handoff clarity	Partial	Explicit
Dispatch readiness visibility	Inconsistent	Immediate
Audit traceability	Weak	Strong

The observed performance metrics indicate that the system maintains stable responsiveness even under increased workload conditions. Furthermore, the workflow comparison demonstrates that the adoption of an event-driven architecture significantly enhances real-time visibility, reduces coordination delays, and improves clarity in operational processes across all participating roles.

VII. Results and Discussion

The evaluation results demonstrate that the proposed architecture enhances both system performance and operational efficiency. The division of the system into experience, transaction, and event planes introduces clear separation of responsibilities, allowing each layer to function independently while maintaining seamless interaction. This design enables rapid propagation of state changes without compromising system stability.

The implementation of a structured lifecycle engine significantly improves process reliability. By enforcing valid state transitions, the system prevents inconsistent or invalid order progression. This ensures that all stakeholders share a consistent understanding of the current order status, thereby reducing confusion and operational errors.

The event-driven communication model further contributes to improved coordination across roles. Real-time broadcasting of updates related to preparation progress, readiness confirmation, and dispatch status eliminates the need for manual status checks. As a result, delays in information flow are minimized, and operational decisions can be made more efficiently.

Additionally, the integration of governance mechanisms such as role-based access control and audit logging strengthens system accountability. These features ensure that sensitive actions are both restricted and traceable, thereby improving system integrity and

administrative control.

Overall, the findings suggest that bakery management systems should be designed as active coordination platforms rather than passive data storage solutions. The proposed approach demonstrates that real-time synchronization, combined with structured workflow enforcement, can significantly improve operational transparency and efficiency.

VIII. Limitations and Future Work

The current study is based on simulated operational scenarios and does not include deployment in a real-world multi-branch bakery environment. Consequently, certain practical factors such as network instability, integration with external delivery services, and large-scale coordination across multiple outlets have not been fully evaluated. These aspects may influence system performance and require further validation in real deployment conditions.

Additionally, the present implementation relies primarily on rule-based logic for operational decision-making. While this approach ensures consistency and control, it does not incorporate predictive capabilities that could enhance planning and optimization.

Future research can extend the proposed architecture by integrating distributed event-streaming platforms to support higher scalability and fault tolerance. The inclusion of predictive models for demand forecasting and intelligent inventory management can further improve operational efficiency. Moreover, expanding the system to support multi-branch coordination and cross-location synchronization would make the platform more suitable for large-scale bakery enterprises. These enhancements would strengthen both the scalability and decision-making capabilities of the system.

IX. Conclusion

This paper introduced an event-driven bakery operations platform designed to unify order management, preparation workflows, customer engagement, dispatch coordination, and administrative governance within a single architectural framework. By adopting a structured lifecycle model and real-time event propagation, the system addresses the limitations of fragmented bakery workflows and improves overall operational efficiency.

The proposed approach enhances visibility across all stages of order processing, ensures faster response to state changes, and enables consistent coordination between different operational roles. The integration of governance mechanisms further strengthens system reliability by enforcing controlled access and maintaining traceability of critical actions.

The findings of this study highlight the effectiveness of event-driven design in transforming

bakery management systems into coordinated, real-time platforms. Such systems not only improve current operational processes but also provide a scalable foundation for future advancements in food-service technology, particularly in environments that demand both responsiveness and structured control.

ACKNOWLEDGMENT

The authors would like to sincerely thank Ms. Sonam Ro-hilla, Assistant Professor in the Department of Computer Science & Engineering (Artificial Intelligence), IIMT College of Engineering, Knowledge Park 3, Greater Noida, Uttar Pradesh, India, for her continuous guidance, valuable suggestions, and support throughout the course of this research work.

The authors also express their appreciation to IIMT College of Engineering for providing the necessary academic resources, infrastructure, and environment that facilitated the successful completion of this study.

REFERENCES

1. B. Michelson, "Event-driven architecture overview," *Patricia Seybold Group*, 2006.
2. M. Fowler, "What do you mean by event-driven architecture?" 2017. [Online]. Available: https://martinfowler.com/articles/ 201701-event-driven.html
3. G. Pardo-Castellote, "OMG data-distribution service: Architectural overview," in *Proc. IEEE International Conference on Distributed Computing Systems*, 2003, pp. 200–206.
4. N. Kreps, J. Rao, and J. Narkhede, "Kafka: A distributed messaging system for log processing," in *Proc. NetDB*, 2011, pp. 1–7.
5. R. Kumar, S. Verma, and D. Singh, "Kitchen display systems and production coordination in hybrid service environments," in *Proc. International Conference on Smart Business Systems*, 2021, pp. 55–60.
6. H. Zhang, Y. Wang, and L. Chen, "Workflow synchronization in distributed retail systems," *IEEE Access*, vol. 11, pp. 34521–34534, 2023.
7. S. Newman, *Building Microservices*. Sebastopol, CA, USA: O'Reilly Media, 2015.
8. G. Hohpe and B. Woolf, *Enterprise Integration Patterns*. Boston, MA, USA: Addison-Wesley, 2003. P. Thönes, "Microservices," *IEEE Software*, vol. 32, no. 1, pp. 116–116, 2015.
9. D. Miller and J. Clark, "Digital kitchen coordination and order orchestration in hospitality systems," *Journal of Service Systems*, vol. 15, no. 3, pp. 88–102, 2021.
10. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.
11. D. F. Ferraiolo, J. F. Barkley, and D. R. Kuhn, "A role-based access control model and reference implementation within a corporate intranet," *ACM Trans. Information and System Security*, vol.

2, no. 1, pp. 34–64, 1999.

12. J. Lee and H. Park, "Role-based access control and operational governance in SME software platforms," *Journal of Enterprise Information Systems*, vol. 18, no. 1, pp. 44–58, 2021.
13. M. Bishop, *Computer Security: Art and Science*. Boston, MA, USA: Addison-Wesley, 2003.
14. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
15. K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.
16. J. Brook, "SUS: A quick and dirty usability scale," in *Usability Evaluation in Industry*, 1996.
17. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, MA, USA: Addison-Wesley, 1994.
18. T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic syntax," *RFC 3986*, 2005.
19. Robinson, J. Webber, and E. Eifrem, *Graph Databases*. Sebastopol, CA, USA: O'Reilly Media, 2015.
20. M. Richards, *Software Architecture Patterns*. Sebastopol, CA, USA: O'Reilly Media, 2015.