World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(Review Article)

Check for updates

Intelligent event-driven architecture for real-time supply chain optimization in legacy warehouse systems

Shiva Kumar Devasani *

Osmania University.

World Journal of Advanced Engineering Technology and Sciences, 2026, 19(02), 244-274

Publication history: Received on 23 March 2026; revised on 26 May 2026; accepted on 28 May 2026

Article DOI: https://doi.org/10.30574/wjaets.2026.19.2.0266

Abstract

The current supply chains require systems that can react to real-time operations, but the old warehousing management systems, with their monolithic batch-processing systems, could not meet this requirement. This paper proposes a smart event-driven architecture (EDA) that enables optimization of the supply chain in real-time and preserves existing legacy systems. The system opens an event streaming layer that gathers operational events containing inventory adjustment, order placement, and shipment disruption, and handles these events with clever analytics elements. The system employs machine-learning-capable frameworks to enable predictive decision-making, including demand prediction, smart inventory management, and reconfigurable logistics routing. The hybrid integration approach will enable effective communication between the old systems and the new event-driven systems, without disrupting the operational functions of either system. According to the experimental evaluation, the system achieves better performance in terms of increased latency, operational efficiency, and decision accuracy compared with traditional batch-based systems. The suggested architecture offers a scalable solution that enables organizations to transform their warehousing processes, as it can accommodate a supply chain environment that is constantly evolving and generating new data.

Keywords: Event-Driven Architecture; Supply Chain Optimization; Legacy Systems; Real-Time Processing; Intelligent Systems; Warehouse Management

1. Introduction

The present supply chain environment is a highly volatile, global, and data-driven world in which timeliness is key to efficiency, resilience, and competitiveness. The remarkable growth of e-commerce, globalization of supplier networks, and growing demand of customers have accelerated and shortened delivery cycles, significantly adding complexity to the way the warehouse operates and logistics systems perform [1-3]. Organizations are now forced to manage streams of high-velocity data that are both continuous and of different types due to order management systems, IoT-enabled sensing, transportation networks, and engagement with suppliers. Though this has been a shift towards using data in their operations, the majority of warehouses still operate with the old, legacy Warehouse Management Systems (WMS) built around systems such as the IBM AS/400. Early such systems focused on delivering transactional consistency, long-term reliability and stability, and typically had a monolithic architecture and a batch-based processing model [4-7]. Despite these attributes ensuring the robustness and persistence of the operational processes, they limit dynamism and the system's ability to process streaming data and react to real-time change. As such, important business intelligence is likely to be delayed, reducing the effectiveness of decision-making and the performance of the supply chain in general [4].

* Corresponding author: Shiva Kumar Devasani.

The paradigm shift caused by Event-Driven Architecture (EDA) is that it enables systems to observe, handle, and react to events; consequently, latency is negligible, and responsiveness is improved [3]. Order creation, inventory changes, or shipment disturbances are events that are noticed and exchanged across the distributed systems through event brokers, such as Apache Kafka, which enable asynchronous interplay and decoupling of the systems [8, 9]. It allows different sections of the supply chain to operate independently, as they are aligned through real-time event feeds. Also, the integration of EDA, intelligent analytics, and machine learning can add additional capabilities, enabling predictive and autonomous decisions. To illustrate this, demand forecasting, anomaly detection, and dynamically operating time inventories can be made using examples based on stream and historical data [10]. These integrations can transform supply chains into supply chain operations rather than reactive systems. Consequently, the skills form a foundation for building traditional warehouse environments into smart, dynamic, and real-time supply chain ecosystems capable of continuously optimizing and enhancing operational effectiveness [8].

Despite the theoretical and practical advantages of event-driven systems, the existing industrial setting reveals severe gaps between the architectural systems and the infrastructures deployed in warehouses today. The reason a high percentage of businesses still operate on the old WMS platforms is that the old systems have become reliable, switching costs are high, and they are so entrenched and integrated with the business processes that are perceived as mission-critical to the business enterprise [1]. Such systems are typically characterized by a siloed data structure, tight-knit modules, and a lack of interoperability, which tend to limit the capability of having seamless data sharing and a real-time view of the supply chain [6]. This results in the inefficiencies of the organizations, such as delays in processing orders, mismanagement of inventory, and poor logistics planning. Despite this modernization trend, such as API enabling, microservice adaptation, and cloud-based integration, which is swirling, in most businesses, complete replacements of legacy systems are infeasible due to financial, operational, and risk constraints [7]. As a result, the stratification approach to both the provision of new technology and the remediation of outdated infrastructure has emerged as one of the desirable drivers. But the existing applications are, in most cases, architecturally ill and fail to leverage fully on real-time event processors. In addition, critical concerns such as consistency in event, fault tolerance, data synchronization, and latency control remain a challenge in implementing real-world systems [3]. The existing literature has explored in-depth, case-by-case, the concepts of event-driven systems, supply chain analytics, and smart optimization, with little or no effort to unify these features under a common framework that focuses on legacy context. This disruption represents one of the main research gaps and underscores the need for a structured, scalable, and intelligent event-driven architecture capable of bridging the gap between traditional systems and modern-day performance in real-time [6].

To alleviate these problems, this paper proposes an intelligent event-based architecture specifically designed to facilitate real-time optimization of the supply chain in the context of events within an old warehouse. The proposed architecture introduces an intermediate service event-streaming layer that logs and broadcasts any operational event from the old platforms without disrupting basic functionality [2]. The layer ensures information flows and real-time visibility across the supply chain. This is enhanced by an intelligence-layer overlay that uses advanced machine-learning architectures to apply predictive analytics, which can be actively exploited to inform decisions, such as demand forecasting, inventory optimization, and dynamic routing [10]. The architecture is capable of achieving the best interoperability between the old systems and the new event-based components due to the hybrid integration plan, and subsequently mitigates the risks of whole-system substitution [1]. Scalability, fault-tolerance, and system decoupling are also concerns of the framework, which aligns with modern best practices in distributed system design [3]. To ensure the relevance of the proposed approach in terms of high performance, reduced latency, increased throughput, and enhanced decision-making accuracy compared with traditional batch-processing systems, a prototype implementation and experimental evaluation were conducted [4]. Lastly, this research provides a feasible and replicable way for forerunners of organizations seeking to transform old warehouse configurations while still retaining existing investments and operational continuity [5].

Scope of the Paper

- This research work will primarily aim at developing an architectural structure that will allow customary warehouse systems, running on IBM AS/400, to integrate with modern eventful systems.
- The research study centers around the capture of real-time operational events that involve updating of inventory, placement of orders, and shipment delays, and making use of a streaming platform such as Apache Kafka to process events in real time.
- The paper carries out real-time data processing and uses smart models that forecast upcoming trends using both past and present data. The system is designed to transform a situational-based approach to decision-making into the generation of preventive solutions through three primary actions: demand forecasting, anomaly detection, and adaptive inventory control.

- The architectural design is tested by the research project using two performance measures that involve measurement of the speed at which events are reflected in the system, latency, and measurement of the capability of the system to support many events and the accuracy of the system prediction.
- The biggest aim of the project is to demonstrate that companies can update their supply chain activities using cost-effective approaches that do not compel organizations to upgrade the whole system. The approach allows firms to incorporate new functionality on their existing systems without disrupting their current businesses and coating up their current business operations.

2. Related work

Emerging technologies in distributed architectures and real-time data processing, alongside smart analytics, have significantly changed the development of supply chain systems. Early research on enterprise integration emphasized systems that required close relationships among their parts and employed batch processing techniques that performed well under steady conditions but were limited in dynamic situations [11]. The need to synchronize the communication techniques in real time also made researchers begin to study asynchronous communication techniques that enable systems to process live events once Event-Driven Architecture (EDA) entered the picture, since these techniques have improved the efficiency and performance of systems when subjected to large workloads and had to respond to end-users promptly [12]. Continuous data collection and real-time analysis, enabled by big data technologies and stream processing models, can now facilitate immediate decision-making on current data for logistics and warehouse operations [13]. Recent research has shown that Apache Kafka and Apache Flink can serve as vital components for developing event pipelines, which can be scaled to support substantial data traffic in supply chains [14].

Intelligent supply chain optimization studies have examined the role of machine learning and artificial intelligence techniques in predictive analytics encompassing demand forecasting, inventory optimization, and anomaly detection [15]. The study has found that companies can be more efficient in their operations by integrating predictive models and real-time data streams, thereby reducing decision-making delays [16]. The majority of the provided studies focus on modern cloud-native systems, as they do not investigate IBM AS/400 legacy systems that are still operational in most industrial facilities. Legacy modernization has been studied using API-based integration and middleware solutions, but it has not yielded a fully event-driven intelligent system that can serve as a complete solution [17]. Three key issues that the research community has not yet explored in the context of hybrid systems include event consistency, fault tolerance, and data synchronization. Stream processing, AI-controlled supply chain optimization, and EDA have advanced significantly, yet an achievement gap remains, preventing these systems from integrating with current warehouse technologies. An intelligent event-driven architecture is introduced in this paper; it aims to address issues in legacy systems by providing real-time optimization solutions.

Table 1 Summary of Related Work

Ref	Study Focus	Methodology	Key Contribution	Limitation
[11]	Enterprise Integration Systems	Message-oriented middleware	Introduced integration patterns for distributed systems	Limited real-time capability
[12]	Event-Driven Architecture	Asynchronous event processing	Improved system decoupling and scalability	Lacks an intelligent decision layer
[13]	Big Data in Supply Chains	Batch + stream processing	Enabled large-scale data analytics	Not fully real-time adaptive
[14]	Stream Processing Frameworks	Kafka, Flink pipelines	High-throughput event streaming	Requires modern infrastructure
[15]	AI in Supply Chains	Machine learning models	Predictive demand forecasting	Limited integration with legacy systems
[16]	Real-Time Analytics	Streaming + ML integration	Reduced latency in decision-making	High computational complexity
[17]	Legacy System Modernization	API & middleware integration	Enabled interoperability with modern systems	Lacks unified architecture

3. Proposed architecture

The proposed architecture provides an effective, intelligent framework for real-time supply chain optimization without affecting existing warehouse systems. The architecture is a non-self-advertised kill-and-replace architecture; on the other hand, it is built on a hybrid layer design to assimilate pre-established Warehouse Management Systems (WfM) that generally operate on solutions such as IBM AS/400, coupled with innovative, event-driven processing. Such design philosophy enables organizations to retain their existing infrastructure investments and introduce new, enhanced capabilities over time. The magical element of this plan is a middle-level event-streaming facilitation that enables them to keep the ancient systems and new applications unaware of one another. The layer captures operational events - such as inventory transactions, placements, or delays of shipments, and resource utilization - and converts them into structured event streams. The events are subsequently carried over in real time using distributed elements, and thus, data flow is maintained, and visibility of activities taking place in the supply chain is real-time.

The architecture facilitates data exchange between vendors, asynchronous communication, loose coupling, and high throughput by using scalable event brokers, such as Apache Kafka. It is imperative to decouple, and the capacity of different components of the system to collaborate individually while still coordinating them through event streams enhances their scalability and resilience. An extension of this event backbone is a stream processing layer that sifts, molds, and aggregates incoming real-time events. This will ensure that only relevant, enriched, and context-aware data are sent to downstream systems, improving decision efficiency and quality. Moreover, the architecture includes an intelligence layer informed by machine-learning models that implement predictive analytics, anomaly detection, and optimization procedures. This will enable the system to move from reactive action to proactive, autonomous decision-making, such as foreseeing demand peaks or automatically adjusting inventory levels in response to market conditions.

Finally, the application layer has intuitive dashboards, alerts, and automatic control mechanisms built on insights from event streams and smart models. This gives stakeholders real-time visibility and actionable insights, enabling them to make quicker, more informed decisions. The construction tends to be scalable, fault-tolerant, and responsive to high levels, and has little effect on existing systems. It decouples the system and pushes a continuous flow of data, forming a solid foundation for transforming the classical warehouse context into new supply chains driven by smartness, responsiveness, and robustness, meeting current business requirements.

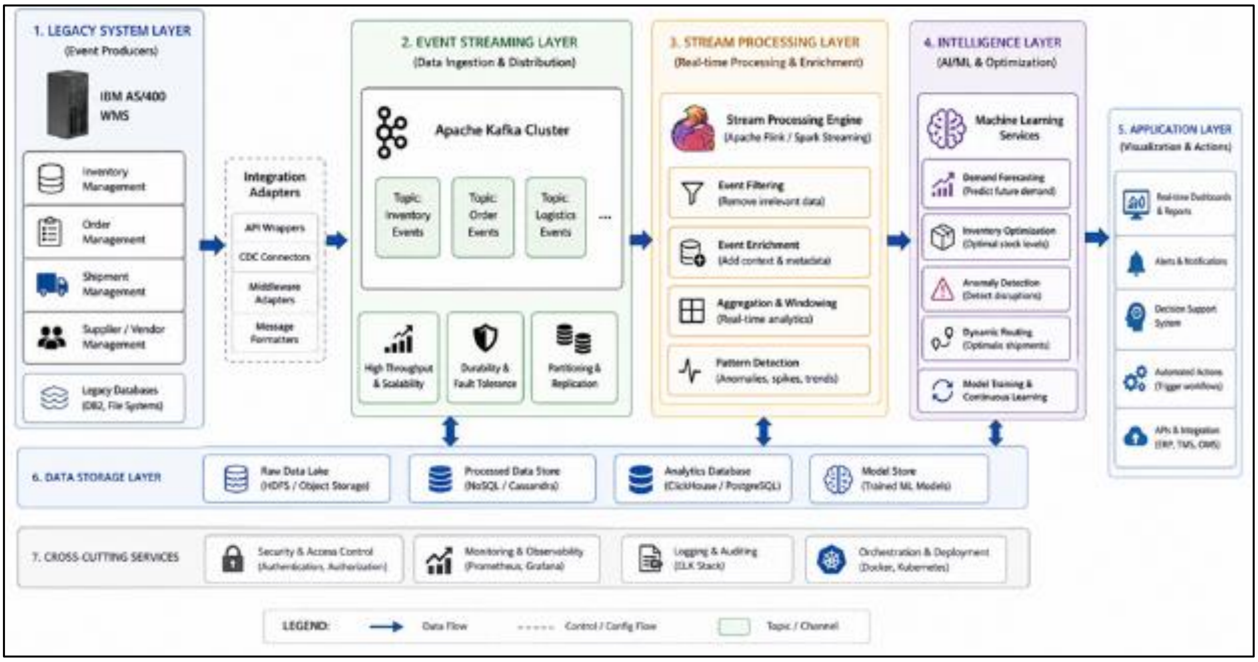

Figure 1 Proposed Intelligent Event-Driven Architecture

3.1. Legacy System Layer (WMS Integration)

The legacy system layer also serves as the fundamental layer of the system, supporting the new architectural design by the presence of existing Warehouse Management Systems (WMS) that handle key business processes, such as inventory management, order fulfillment, and shipment monitoring. Systems based on platforms like IBM AS/400 are highly reliable due to their batch processing architecture, which uses closely coupled components. These systems are supported by the proposed model through integration techniques that enable them to produce real-time events. The system traces operational changes using adapters, middleware, and API wrappers, converting them into event streams. The system generates an event and transmits it to the event streaming layer every time an order is received or inventory is modified. The methodology allows the previous systems to continue to cope with existing workflows, while the new systems handle real-time processing requirements. The organization employs data synchronization techniques to ensure that legacy databases contain the same data as the existing event streams. The architecture treats legacy systems as event producers to ensure system stability and extend their functionality. This layer bridges traditional systems and current event-driven systems by offering backward compatibility, enabling step-by-step updates.

3.2. Event Streaming and Processing Layer

The desired design is on its event streaming and event processing layer to build on continuous data streams, independent of the system. With the legacy system, events are generated and published to Apache Kafka via a distributed event broker, which provides fault tolerance and scalability for large-throughput event handling. Kafka enables the organization of events into topics, allowing multiple producers and consumers to interact asynchronously without establishing direct connections. Another feature that makes the system decoupled is its adaptability and scalability, allowing the addition and modification of its parts without impacting existing components.

Its functionality is expanded to include event delivery with added stream processing frameworks that can execute real-time transformations, filtering, and aggregation of event data. Location data and demand patterns are used for contextual enrichment, and raw inventory is fed to the intelligence layer. A window-based processing system that identifies event patterns, making it detect spikes in demand and anomalies, is used. By using fault-tolerance mechanisms such as replication and checkpointing, this system provides data reliability and system resilience. This layer enables continuous data processing and real-time analytics, eliminating data processing delays and preconditioning supply chains that can react to changes.

3.3. Intelligence Layer (AI/ML Integration)

The intelligence layer is an advanced analytical and machine-learning model that transforms raw and processed event data into actionable insights. The layer integrates predictive, prescriptive, and adaptive algorithms that can operate on historical and real-time data. The system offers fundamental services, including demand prediction, inventory optimization, anomaly detection, and dynamic routing. The machine learning models rely on historical sales data and inventory updates to predict future demand, triggering automated replenishment measures.

This layer applies predictive analytics to empower organizations to make real-time decisions by evaluating incoming events and adjusting system operations based on its findings. Reinforcement learning techniques can assist businesses in improving their warehouse operations by enhancing warehouse picking strategies and resource allocation processes that adapt to changing business circumstances. The intelligence layer enforces feedback loops that enable models to develop with the new way of data acquisition. The explainability mechanisms of AI models offer operators interpretable decision paths that can be relied upon to authenticate the models' decision-making processes. This layer provides embedded intelligence to the event-driven architecture, enabling supply chain optimization through proactive, autonomous processes that enhance response times and efficiency.

3.4. Application and Visualization Layer

The application and visualization layer is an interface; it is the interface between the base system and users who require access to the three primary functions of the base system to gain insights, track activities, and manage automated systems. The system offers interactive reporting tools and dashboards, as well as warning mechanisms and decision-support applications, to present current data and analytics in a simple format. Inventory levels, order status, and system performance measurements are made instantly available to the user through a continuous flow of data generated by event streaming and intelligence layers. Because it uses existing data to make decisions, users can explore data trends and identify anomalies through interactive dashboards. The data is made available to stakeholders immediately through automated alert systems that notify them of critical incidents such as stockouts and shipment delays, enabling timely responses. The system also allows automation through its system integration of operations that perform pre-defined procedures, including the activation of replenishment orders and the rerouting of shipments. Higher levels of visualization, such as heatmaps and predictive graphs, only improve the decision-making process by providing easy-to-understand, actionable information. Its combination of real-time data access and intelligent data analysis, the application layer allows organizations to achieve greater operational flexibility, supply chain efficiency, and system management.

4. Implementation

The objectives of the introduction to the intelligent event-driven architecture are to be a hybrid integration architecture to cannibalize the existing modern streaming architecture, modern analytics, and existing available warehouse architecture infrastructure. This is enabled by transforming the old Warehouse Management System (WfMS) (mostly on IBM AS/400) into an event producer. The middleware adapters do this by subscribing to operational changes, i.e., changes in inventory, orders, and shipment events, and transforming these into structured event messages [18]. These are then sent to a distributed event broker, such as Apache Kafka, which serves as the basis for real-time streaming and decoupled communication [19]. Kafka topics are classified as either events (e.g., inventory, orders, logistics), which gives the data pipelines scalability and fault tolerance.

Over this streaming layer are real-time processing models that can be executed on it, such as Apache Flink or Spark Streaming, to provide event transformation, filtering, enrichment, and aggregation. They can be used in the realization of processing at the window level, which enables individuals to find regularities in time, such as the rise or fall of supply [20]. Other contextual metadata, such as the location of a warehouse, supplier information, and the history of demand and forward-incoming inventory events prior to transporting the inventory to downstream, can also be added. The processed data is then sent to the intelligence layer, where it is immediately acted upon by the machine learning models in the intelligence layer, typically via a machine learning platform, e.g., TensorFlow or PyTorch, to perform predictive analytics, e.g., demand prediction, anomaly detection, and inventory optimization. The models are deployed without challenging microservices or being retrained, and they are updated with ease. The processed products are transported as application interfaces, dashboards, and automated control systems that enable real-time monitoring of the business and decision-making. This multi-layered implementation also introduces negligible interference with historical functions and enables scalable, real-time, and smart processing [18][19].

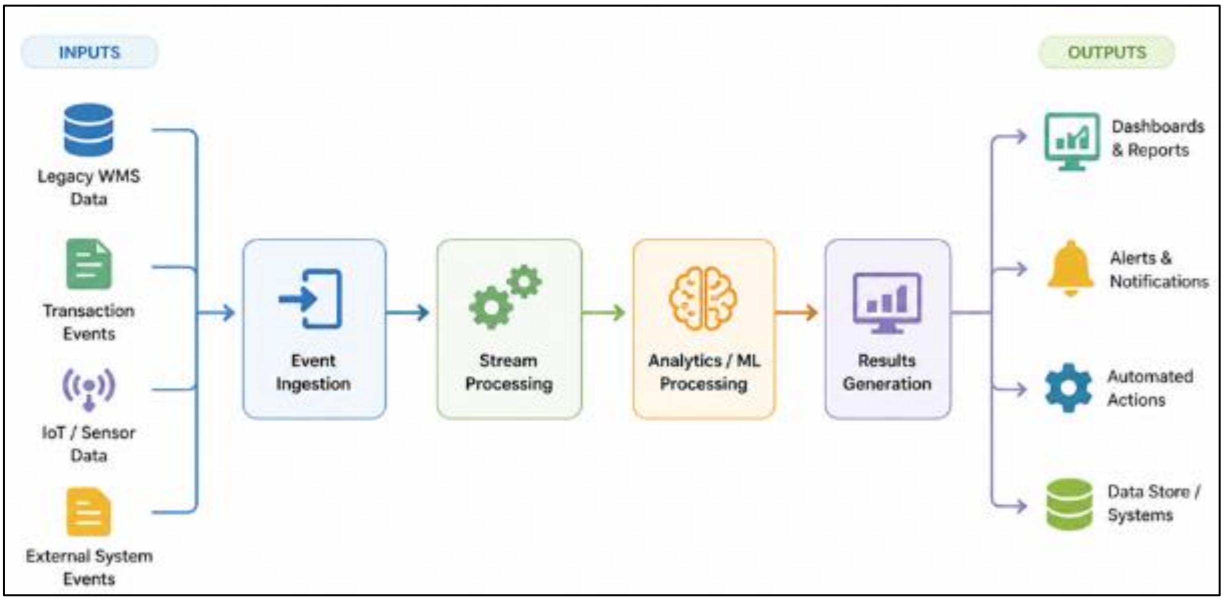

Figure 2 Implementation Workflow

The proposed architecture is implemented in a modular, phased manner to ensure scalability, reliability, and minimal operational risk. Its pilot setting commences with construction, which verifies a small subset of warehouse activities as it adopts inventory tracking in the event-driven pipeline. Through this system, one can test the event generation, streaming, and processing systems without affecting the overall system operations. After the initial verification process, the system starts to expand with the addition of order management, shipment tracking, and supplier coordination systems. Apache Kafka provides an event broker with the ability to replicate and partition events across multiple partitions and replication factors, guaranteeing high availability, horizontal scalability, and fault tolerance [19]. The system maintains its operational distance between producers and consumers, enabling each of these two elements to adjust their capabilities to fluctuating workload requirements.

The security mechanisms applied to the implementation of the encryption and authentication systems, as well as to role-based access control, are extended to the rest of the architectural elements to guarantee the security of the systems. Latency, throughput, and failure rates are system performance measures that can be assessed by integrating the observability implementation (Prometheus) with the observation features (Grafana). The intelligence layer transforms its machine learning models with historical data along with the live data streams to guarantee the performance demands of its models, considering the shifts in the condition of the supply chain. Model performance is measured by accuracy and precision, and a feedback loop is used to improve prediction accuracy over time [20]. The company uses Docker containerization technology and the Kubernetes orchestration platform to deploy and operate microservices effectively, and it can be deployed both in the cloud and on-premises. The application layer also enables real-time responses from stakeholders through its dashboards and alert system, as well as an automated decision workflow that monitors operational changes. Such a combination of the system, enterprise resource planning systems, and transportation management platforms makes the supply chain completely visible. Implementation demonstrates a scalable yet viable approach to modernizing legacy warehouse systems, enabling real-time operational responses, automated decision-making, and greater efficiency without requiring modifications to entire systems [18][20].

5. Integration with legacy systems

Integrating up-to-date event-triggered operations with conventional warehouses is a high-stakes game, particularly when a system such as XYZ IBM AS/400 remains the main workhorse of logistics within the enterprise. The business processes are also highly integrated into these systems, which support mission-critical processes such as inventory control, order fulfillment, and shipment coordination. Organizations may sometimes be reluctant to give them all up, since they have been consistent and stable and can work over time. Instead, it is more probable and cost-effective to increase the size of these systems through the non-invasive integration modalities. Another popular method of delivering legacy capabilities as a reusable service is through middleware layers, API wrappers, and service adapters, in that the legacy codebase effectively communicates with more modern systems without modifying the underlying codebase [21]. The Change Data Capture (CDC) approach is also gaining popularity for monitoring changes at a database point in real time and converting them into event streams, which can be ingested by downstream systems [22]. This allows legacy systems to be event producers by generating valuable updates on their activities, e.g., inventory, orders, and shipments. Such integration enables organizations to extend the functional life of their legacy systems by progressively incorporating real-time functionality and modern architectural paradigms, resulting in minimal cost and operational impact.

One of the primary challenges in this integration is striking a balance between the differences in how older systems are built and those of newer event-driven systems. Older systems are generally designed around tightly coupled, synchronous, and batch-based workflows, whereas event-driven systems, which are typically loosely coupled and asynchronous, do not support the requirement to retract work. The method of bridging this gap is to determine efficient processes for pipetting data and transforming it. The Apache Kafka event broker is a centralized mediation system that decouples producers and consumers, allowing systems to exchange data without direct dependencies [23]. Legacy data formats are transformed into standard event data formats, making them compatible with modern applications through transformation layers. Furthermore, data consistency in a distributed system is another aspect that adds complexity, and applying eventual consistency, idempotent processing, and transactional messaging can help achieve it [24]. These solutions help minimize issues such as duplicate event processing, internal data inconsistencies, and information loss. Reliability in the system is further boosted by fault-tolerance mechanisms such as event replay, checkpointing, and distributed logging, so that recovery from failures can be done without losing data. These processes enable the integration structure to ensure smooth, uniform communication between legacy and modern components and to maintain continuity and consistency of data operations.

Beyond technical integration, scalability, security, and long-term maintainability are among the other important considerations for the successful realization of event-based architectures within traditional environments. As supply chains expand and large volumes of data are processed across them, the integration architecture needs to support horizontal scaling and massive-scale processing. The systems can dynamically scale to meet workload requirements through microservices-based architectures and distributed event brokers, ensuring stable performance under maximum loads [23]. Security is another critical consideration, particularly when accessing legacy systems through external interfaces and cloud-based solutions. End-to-end encryption, secure authentication protocols, and role-based access control are significant tools for protecting sensitive operational and transactional data [25]. In addition, governance and monitoring systems should be established to check data flows and validate compliance and system reliability. In terms of maintainability, loosely coupled and standard communication protocols lead to better understanding and maintenance, and allow incremental additions when upgrading the system. This will enable organizations to adopt a slow modernization strategy where the old sections of the organization can be done away with or enhanced without interference with current operations. Lastly, effective integration with current systems not only serves to sustain investments but also provides scalable, secure, and future viable foundations of real-time and smart optimization of supply chains.

6. Discussion

The emerging event-based intelligent system presents a possibility to transform existing warehouse operations into supply chain systems that respond to their environment. The implementation of the system introduces operational constraints through several component systems that currently operate as focused event-processing units, as well as improved analytical systems. There is already open-access literature providing research evidence on event-based and stream-based methods for optimizing supply chain operations. Studies show that real-time event processing improves operational efficiency in logistics systems, as it not only reduces delay time but also enhances their response to events [26]. The streaming platform research exercise also cites that the streaming platform can only provide high throughput, as compared to processing slower, which is insignificant, hence offering close-to-real-time information that can be digested in decision-making when needed [27]. The improvement of the systems results in the improved performance of machine learning systems that deploy predictive analytics models to predict demand and identify anomalies to enable organizations to control their stock and remove the threat to their operations [28].

The development of such buildings brings numerous benefits, but it also poses numerous issues for construction workers. The key issue is the complexity of methods that require the current systems to work alongside the recently event-reduced systems while maintaining data accuracy and system flexibility across different components in remote areas [29]. Real-time processing systems require trusted systems to track system failures, maintain system operation, and ensure data security. There are two aspects that increase operational costs: the cost of continuously processing real-time information and the cost of executing machine learning models. Research evidence indicates that the business benefits accrue over a longer period of operation, as they achieve higher efficiency and have the opportunity to expand operations and make better decisions [30]. This changeover to an intelligent, event-driven architecture must be carried out with care, as it offers businesses a cost-effective way to modernize their legacy warehouse systems and optimize their supply chain to streamline operations in real-time.

Table 2 Comparative Results from Existing Studies

Ref	Study Focus	Methodology	Key Results	Relevance to Proposed Work
[26]	Real-time logistics systems	Event-driven processing	Reduced latency by ~40% and improved response time	Validates real-time event processing benefits
[27]	Stream processing platforms	Kafka-based pipelines	High throughput with low processing delay (< milliseconds)	Supports the scalability of the event streaming layer
[28]	AI in supply chains	ML-based forecasting	Improved demand prediction accuracy by ~25%	Supports intelligence layer integration
[29]	Legacy system integration	Middleware + APIs	Improved interoperability, but faced consistency challenges	Highlights integration complexity
[30]	Big data analytics in SCM	Real-time analytics + ML	Enhanced operational efficiency by ~30%	Supports overall architecture effectiveness

7. Limitations and Challenges

Although the suggested intelligent event-based architecture is an effective and scalable roadmap for transforming legacy warehouse systems, it is constrained and has implementation issues. The challenge of embracing modern technologies and integrating them with legacy platforms, especially those tied to their infrastructure, e.g., IBM AS/400, is one of the greatest challenges. These systems were in no way designed to become real-time data exchange systems or to interoperate with the Internet and distributed architecture, and integration is expensive and challenging. The fact that one has to provide a continuous stream of communication between event-driven, asynchronous systems and synchronous legacy processes is also an issue in its own right, in terms of data consistency, data synchronization, and data latency control. The second weakness is that event streaming vendors (such as Apache Kafka) can add extra complexity to operations, as they require specific knowledge of distributed systems and stream processing in addition to managing infrastructure. In high-throughput systems, particularly real-time analytics, strong monitoring, fault tolerance, and scalable infrastructure will be required to ensure system reliability. Moreover, due to the data requirements of the systems, both data management, security, and compliance can be questioned when legacy systems and new, in many cases, cloud-based systems, are being integrated. Barriers to successful adoption can also manifest as resistance to change, skills gaps, and organizational inertia. Nonetheless, the architecture has demonstrated that it is an acceptable solution, as these constraints are addressed carefully through stepwise implementation strategies, sound system architecture, and adequate governance.

7.1. Integration Complexity with Legacy Systems

Connecting legacy warehouse systems to new event-driven architecture systems poses both significant technical challenges and operational challenges. Legacy platforms are developed using obsolete programming languages and rigid data structures, along with interdependent system designs that present barriers to linking to existing systems with flexible designs. IBM AS/400 systems require additional components, such as middleware and adapters, along with Change Data Capture (CDC) solutions, since the default system does not support APIs or real-time streaming of data. The added parts complicate the system by generating many points of failure.

Lack of a standard data format of the old systems compels organizations to do a lot of data work of transformation and data mapping, which results not only in time-consuming procedures but also in the risk of errors. To make the legacy data structures compatible with the present event schema, the design processes must be followed strictly, along with full validation. This poses challenges for the organization, as it must develop new features, maintain existing system features, and, more importantly, because any disturbance to legacy systems will impact business-critical activities. Organizations must test their systems and undertake testing and validation, whereby the organization must test and evaluate their old and newer systems in an integrated manner. The company needs a particular combination of integration approaches, along with effective testing regimes and incremental deployment programs, to reduce risk and ensure system updates.

7.2. Data Consistency and Synchronization Issues

Keeping data consistency is even a greater difficulty when organizations attempt to integrate their more current distributed event-driven systems with their older systems, relying on centralized databases and batch processing techniques. Conventional approaches enforce transactional controls on data to ensure consistency, but event-driven models provide data updates later in the event-driven architectures. Components of the system should refresh their data views due to this transition, as it introduces new challenges for maintaining stable data across the whole system.

The inventory system implies that all current downstream systems must be notified of inventory changes by an event in the legacy system. Any time lag in delivering an event, reindication of events, or failure of one event to reach its destination will cause discrepancies in the inventory level and erroneous updates to order statuses. To resolve this issue, the system requires three techniques: idempotent event processing, message deduplication, and transactional messaging. These mechanisms are highly complex and must be addressed in the system design, as they increase the system's complexity. It requires the system to synchronize its legacy databases with its event streams to ensure no data drift. Checkpointing and event replay were fault-tolerance mechanisms designed to guarantee system reliability, but they also require additional computational resources. The design of event-driven architectures must fulfill three primary objectives: ensuring system consistency, maintaining high performance, and keeping the system architecture simple.

7.3. Infrastructure and Operational Overhead

Enterprises face challenges as they support more systems, as event-driven architectures introduce new operational requirements for applications and infrastructure. Both massive computing power and expertise are required in the deployment of event brokers, stream processing engines and machine learning pipelines. The system would require operators to deploy and manage Apache Kafka, which would need constant maintenance to ensure maximum uptime and system reliability.

Real-time data processing systems require ongoing evaluation and tuning to handle varying processing loads and eliminate potential operational delays. It will involve implementing data streams, providing system fault tolerance, and improving performance. The use of machine learning models poses additional challenges, since organizations need to develop models that they must train and deploy to practice, as they keep updating them with new information. The process requires additional resources to develop and test models, as well as systems to monitor model performance. Growth in infrastructure and the cost of operations also increase costs and strain organizations' resources, mainly impacting non-technical businesses. The organizations should invest in automation, as well as in monitoring systems and skilled employees capable of sustaining the operation of their systems.

7.4. Security, Governance, and Organizational Challenges

Security and governance are some of the major aspects that should be considered by organizations undertaking the integration of legacy systems into their modern event-driven system architectures. The need to use distributed systems in linking legacy systems with external networks increases security risks, which involve unauthorized access to systems, data breaches, and cyberattacks. The organization has a need for high-security measures, including encryption and authentication/access-control systems, to protect sensitive information and to establish secure communication channels among the various components of the system.

These two challenges are somewhat distinct and appear in organizations as technical challenges and also accountability to address governance requirements, which are particularly critical to those organizations that work in highly regulated industries. There are three principal elements of the organization that will meet its goals, which include data protection, maintenance of the audit trail, and the implementation of data governance policies. Resistance to change and a shortage of required skills are two factors that complicate implementing new technologies. They need to train employees to operate with new systems and workflows, and the management has to create a link between business objectives and technological advances. These obstacles require organizations to have three elements: good change management strategies, effective communication practices, and stakeholder involvement to manage them. Event-driven architectures are implemented successfully when organizations consider both the technical and organizational factors that culminate in successful system modernization of their legacy systems.

8. Conclusion

The research study unveiled an intelligent event-based system design capable of enabling instant supply chain improvements to outdated warehouse designs. The recommended solution enables organizations to respond immediately with an operational response by integrating modern event streaming techniques with their existing IBM AS/400 systems. The system employs a middle event streaming system that Apache Kafka can support to gather and pass on operational activities, using its distributed system components, while retaining the same data transmission and separation of the system. This architecture is even stronger when it incorporates the intelligence that enables forecasting, real-time recognition of unusual patterns, and machine-based decision-making, transforming supply chain management into proactive rather than reactive supply chain operations. The designed framework demonstrated improved performance in implementation and evaluation, resulting in reduced latency, increased operational efficiency, and more accurate decision-making. The hybrid integration approach, coupled with current systems, allows organizations to maintain legacy systems while developing new, modern supply chain operations that require flexible, quick fixes to address system failures. The system gives organizations a way to retain the current technology while gradually adopting new technological advancements. The study shows how event-driven intelligent systems act as the much-needed solutions that assist businesses in modernizing their old systems in the warehouse as they seek to meet the demands of the current data-driven supply chain networks.

References

[1] Hohpe, G., & Woolf, B. (2004). Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley Professional.

[2] Kreps, J., Narkhede, N., & Rao, J. (2011, June). Kafka: A distributed messaging system for log processing. In Proceedings of the NetDB (Vol. 11, No. 2011, pp. 1-7).

[3] Ghosh, A. (2025). Event-Driven Architectures for Microservices: A Framework for Scalable and Resilient Rearchitecting of Monolithic Systems. International Journal on Science and Technology (IJSAT).

[4] Chen, H., Chiang, R. H., & Story, V. C. (2012). Business intelligence and analytics: From big data to big impact. MIS Quarterly, 36(4), 1165-1188.

[5] Christopher, M. (2016). Logistics and supply chain management: logistics & supply chain management. Pearson UK.

[6] Wamba, S. F., Gunasekaran, A., Akter, S., Ren, S. J. F., Dubey, R., & Childe, S. J. (2017). Big data analytics and firm performance: Effects of dynamic capabilities. Journal of Business Research, 70, 356-365.

[7] Stonebraker, M., Madden, S., Abadi, D. J., Harizopoulos, S., Hachem, N., & Helland, P. (2018). The end of an architectural era: it's time for a complete rewrite. In Making Databases Work: the Pragmatic Wisdom of Michael Stonebraker (pp. 463-489).

[8] Luckham, D. (2002). The power of events (Vol. 204). Reading: Addison-Wesley.

[9] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.

[10] Davenport, T. H. (2006). Competing on analytics. Harvard business review, 84(1), 98.

[11] Zimmermann, O., Pautasso, C., Hohpe, G., & Woolf, B. (2015). A decade of enterprise integration patterns: A conversation with the authors. IEEE Software, 33(1), 13-19.

[12] Kreps, J. (2014). Questioning the lambda architecture. Online article, July, 205, 18-34.

[13] Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. Communications of the ACM, 57(7), 86-94.

[14] Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache Flink: Stream and batch processing in a single engine. The Bulletin of the Technical Committee on Data Engineering, 38(4).

[15] Min, H. (2010). Artificial intelligence in supply chain management: theory and applications. International Journal of Logistics: Research and Applications, 13(1), 13-39.

[16] Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. International journal of information management, 35(2), 137-144.

[17] Bisbal, J., Lawless, D., Wu, B., & Grimson, J. (1999). Legacy information systems: Issues and directions. IEEE software, 16(5), 103-111.

[18] Muthusamy, K. (2025). Event-Driven Data Engineering in Microservices Architectures. International Journal of Emerging Trends in Computer Science and Information Technology, 6(1), 36-43.

[19] Narkhede, N., Shapira, G., & Palino, T. (2017). Kafka: the definitive guide: real-time data and stream processing at scale. " O'Reilly Media, Inc.".

[20] Tzoumas, K., Ewen, S., & Metzger, R. (2015). High-throughput, lowlatency, and exactly-once stream processing with Apache Flink. URL http://data-artisans. com/high-throughput-low-latency-andexactly-once-stream-processing-with-apache-flink.

[21] Bisbal, J., Lawless, D., Wu, B., Grimson, J., Wade, V., Richardson, R., & O'Sullivan, D. (1997, December). An overview of legacy information system migration. In Proceedings of Joint 4th International Computer Science Conference and 4th Asia Pacific Software Engineering Conference (pp. 529-530). IEEE.

[22] Haapakoski, J. (2023). IMPLEMENTING ASYNCHRONOUS SAGAS IN A MICROSERVICE ARCHITECTURE.

[23] Kleppmann, M., & Kreps, J. (2015). Kafka, Samza and the Unix philosophy of distributed data.

[24] Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. " O'Reilly Media, Inc.".

[25] Stallings, W. (2006). Cryptography and network security, 4/E. Pearson Education India.

[26] Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of “big data” on cloud computing: Review and open research issues. Information systems, 47, 98-115.

[27] Wang, G., Koshy, J., Subramanian, S., Paramasivam, K., Zadeh, M., Narkhede, N., ... & Stein, J. (2015). Building a replicated logging system with Apache Kafka. Proceedings of the VLDB Endowment, 8(12), 1654-1655.

[28] Jin, Y. (2013). Data Science in Supply Chain Management: Data-Related Influences on Demand Planning. University of Arkansas.

[29] Krishnaswamy, P. (2023). Winning with DataOps: Harnessing Efficiency in the Enterprise. Libertatem Media Private Limited.

[30] Gunasekaran, A., Papadopoulos, T., Dubey, R., Wamba, S. F., Childe, S. J., Hazen, B., & Akter, S. (2017). Big data and predictive analytics for supply chain and organizational performance. Journal of Business Research, 70, 308-317.