AN AUTOMATED INTEGRATED FRAMEWORK FOR PURCHASE ORDER EXCEPTION HANDLING AND MANUFACTURING SYNCHRONIZATION

[1]Harsh S, [2]Shobhanjaly P.Nair

[1]Student, [2]Assistant Professor

[1]Department of Artificial Intelligence and Data Science

[1]Sri Venkateswara College of Engineering, Chennai, India

Abstract : Today's factory supply networks depend on ERP platforms to link buying information with workshop activity. Yet POs often fail checks or layout rules when divided or turned into barcodes. Because these steps do not sync well, delays hide what is happening, leaving outside suppliers unaware of current tasks - spreading issues across the chain. A robust, self-driven cyber-physical model is introduced here, built on separate layers of small services that spot broken purchase records, contain them, then fix without help.Starting with Oracle Integration Cloud, the system handles real-time coordination between services while using Oracle APEX to track diagnostics from one main hub. Instead of relying on batch updates, live signals from the vendor interface feed directly into a master record stamped with barcodes. That running log then kicks off event transfers through Apache Kafka, which stores each change in permanent sequence. Because every update flows through this chain, nothing gets lost along the way. Recovery takes less time now since errors are caught early and fixed automatically. Manufacturing sites spread across locations stay aligned without extra help from central IT teams. Compliance stays intact because every step leaves a traceable mark.

IndexTerms **- Supply Chain Resilience, Digital Twin, Oracle Integration Cloud, Apache Kafka, ERP Exception Handling, Event Streaming, Cyber-Physical Systems. .**

I. INTRODUCTION

In enterprise-scale manufacturing ecosystems, production facilities require structured purchase orders (POs) to acquire raw materials and components, ensuring that clear product division is preserved throughout the fulfillment cycle. External suppliers heavily depend on these split components because they generate individualized barcodes to track each discrete shipment from initial delivery to final billing. Although modern Oracle Fusion Cloud SCM tools have introduced advanced data management functionalities over time, significant integration latencies are still encountered when attempting to link centralized business records with external vendor web portals. These processing delays primarily occur because the synchronization schedules between distinct enterprise applications fail to operate correctly or remain fundamentally misaligned.

Consequently, a critical purchase order workflow may stall indefinitely if transactional details mismatch, mandatory metadata is absent, or active network connections drop prematurely. The entire transaction remains suspended in an unfulfilled state because the core ERP platform fails to complete its validation processing while simultaneously requiring the document to show an active status.

The total absence of granular, real-time logging mechanism to track header-level failures and unexpected validation anomalies prevents maintenance operations from maintaining visibility over system progress. As a result, external suppliers sit completely idle, unable to initiate specialized production tasks because downstream updates fail to render within their operational dashboards. The broader production schedule subsequently faces severe delays because the physical workshop processes encounter a halt point, creating a domino effect that forces planners to repeatedly construct entirely new scheduling parameters.

Historically, internal IT personnel were required to resolve these synchronization breakdowns through manual interventions at the local work site. Database administrators were forced to inspect staging tables manually because the legacy architecture lacked automated error correction or self-healing capabilities. Support teams had to manually isolate defective transaction numbers to execute remedial data patches through a gradual, inefficient process. This reactive approach to exception handling introduces significant room for human error, extends data synchronicity gaps, and fails to satisfy modern regulatory tracking compliance.

This study resolves these critical operational deficiencies by establishing a sustainable, dual cyber-physical framework that seamlessly integrates disparate enterprise architectures, mitigating transactional errors before they disrupt the supply chain.

II. LITERATURE REVIEW

A. Digital Supply Chain Ecosystems and Digital Twins

The conceptual foundation of this architecture is derived from the Digital Supply Chain Ecosystem (DSCE). As established by Ivanov, a modern DSCE represents a tightly integrated network bound by digital tools, artificial intelligence, and cloud computing architectures, enabling live models to dynamically respond to disruptions in real time. This framework specifically expands upon the seminal work of Grieves and Vickers regarding the deployment of Digital Twins as a primary mechanism to reduce unexpected, undesirable emergent behaviors through continuous digital mirroring.

Unlike passive Level 1 monitoring variants that merely observe states, this framework implements a Level 3 "Intelligent Digital Twin" configuration. This configuration leverages an intelligent middleware abstraction layer to automatically intercept and restart stalled purchase orders. Furthermore, Tao et al. emphasize that the industrial application of digital twin structures must provide a continuous "digital thread" to guarantee data integrity across the entire manufacturing lifecycle.

B. Risk Interaction and Failure Propagation

Because modern software architectures frequently suffer from systemic cascades where a fault in one subsystem triggers widespread failures, contemporary research focuses on tracking data-driven anomalies within industrial grids. Liu et al. introduced the concept of Higher-Order Interaction Networks to mathematically model and map how local data failures propagate through complex, interdependent environments. Within highly coupled ERP systems, a singular validation failure during structural schema checks can instantly halt all dependent supplier deliveries. This framework actively mitigates this vulnerability by deploying a JSON-based Audit Log in tandem with Apache Kafka to establish an independent Traceability Node, successfully preventing the quiet data failures highlighted in literature.

C. Intelligent Orchestration and Dynamic Scheduling

Advanced manufacturing networks demand real-time data synchronization over traditional, batch-based processing models. Research conducted by Li et al. demonstrates that utilizing deep reinforcement learning methodologies for the dynamic scheduling of parallel machinery yields significantly higher operational efficiency compared to static order planning. Instead of subjecting the supply chain to hours of latency via legacy bulk processing, real-time transactional adjustments driven by immediate incoming signals show superior performance for high-mix product variations. This validates the strategic selection of Oracle Integration Cloud (OIC) to achieve event-driven orchestration, eliminating the dependencies associated with slow, clockwork imports.

D. Strategic Resilience and Event Streaming

In accordance with the supply chain models articulated by Berger et al., modern industrial networks require immutable data-logging infrastructures to maintain systemic resilience. Monostori asserts that Cyber-Physical Production Systems (CPPS) must exhibit a high degree of local autonomy to effectively manage uncertainty and maintain global network stability. Because traditional relational databases are inherently vulnerable to accidental data overwrites and modifications, this framework introduces a decoupled Barcode Trail Log backed by distributed event streaming via Apache Kafka. As proven by Garg, streaming enterprise telemetry through Kafka ensures that every subtle state transformation is durably committed as a permanent, sequential, and unalterable event.

III. ANALYSIS AND PROBLEM GAP

An evaluation of recent contemporary literature [1]-[8] reveals that while the theoretical paradigms surrounding supply chain digital twins are conceptually sound, practical, real-world deployment methodologies remain considerably limited. The vast majority of academic publications frame operational resilience either as a broad corporate strategy or as highly abstract mathematical formulations, omitting a clear focus on day-to-day operational disruptions. Three critical gaps have been identified:

1. **Practical Integration Barriers:** Physical implementation often encounters significant technical barriers. Existing architectural setups fail to provide clear, actionable integration guidelines for low-code web frameworks like Oracle APEX when interacting with rigid ERP validation structures.
2. **Absence of User-Centric Recovery:** Existing enterprise tools systematically alienate non-technical business users when transaction exceptions occur. Resolving a stalled transaction typically requires escalating tickets to specialized database administrators, resulting in severe operational bottlenecks.
3. **Traceability Gaps during Transitions:** During the critical transmission window where payload data moves from a primary ERP ecosystem to an external supplier platform, end-to-end visibility is frequently lost. This structural delay breaks the chain of custody and hides transit anomalies.

This research directly addresses these three deficiencies by designing and validating a decoupled, 4-tier architecture engineered to provide concurrent technical resilience and user-level observability.

Fig. 1. Issues & Challenges

IV. RESEARCH METHODOLOGY & PROPOSED ARCHITECTURE

Fig. 2. End-to-End System Data Flow from User Interaction to Immutable Audit Log.

To achieve high availability, robust fault tolerance, and an absolute separation of concerns, the proposed system operates as an active, data-driven Digital Twin utilizing a structured "Select-Retry-Sync" integration paradigm. The architecture is strictly decoupled into four highly specialized logical layers:

A. Presentation and Diagnostic Layer (Oracle APEX)

The presentation layer serves as the primary observation, administration, and control node for operational supply chain personnel. Built using Oracle APEX, this low-latency, web-based frontend securely communicates with the underlying data repository.

1. Error Resolution Dashboard: This specialized module extracts purchase orders currently trapped in an error status directly from temporary staging tables, removing the need for manual SQL querying. Each transactional entry clearly surfaces its unique identification number, exact timestamp of failure, and the precise validation error payload, allowing administrators to rapidly spot and diagnose structural data anomalies.
2. Action Controllers: Rather than initiating standard IT support workflows, the interface provides built-in action scripts enabling bulk processing of failed transactions. Upon user triggering, the dashboard aggregates the precise PO metadata, compiles it into a secured payload package, and transmits it via web protocols to the backend middleware.

B. Middleware Orchestration Layer (OIC)

Oracle Integration Cloud (OIC) acts as the centralized data routing and payload transformation engine. Exception handling is deliberately decoupled from the core ERP platform, keeping Oracle Fusion SCM clean and unburdened by repetitive retry processing.

1. Protocol Translation: OIC orchestrates asynchronous bi-directional communication between REST and SOAP protocols. It ingests inbound JSON structures from the APEX dashboard and employs graphical XSLT mapping tools to structurally reformat the payloads into validated SOAP XML schemas required by the ERP system.
2. Idempotency and Logic Execution: To maintain absolute transactional consistency during intermittent network drops, smart routing rules are enforced. The system records state history to prevent duplicate order generation within the ERP system during bulk retry submissions.

C. ERP Fulfillment and Synchronization Layer (Oracle Fusion)

Following successful schema validation and payload transformation within OIC, data is transmitted via secure SOAP web services into Oracle Fusion Cloud SCM.

1. System of Record: Oracle Fusion Cloud SCM serves as the absolute system of record, processing the validated PO splits, generating subsequent manufacturing Work Orders, and assigning alphanumeric barcodes to each individual split line item.
2. Automated Synchronization: To bridge the gap between internal records and external ecosystems, PL/SQL background schedulers (DBMS_SCHEDULER) are deployed. These background processes extract barcode strings and push them to the external Vendor Portal in near real-time, eliminating supplier idle time.

D. Real-Time Audit and Traceability Layer (Apache Kafka)

To secure the enterprise against compliance risks and ensure complete transaction transparency, a dedicated, highly scalable event streaming architecture is deployed.

1. Barcode Trail Log: Lifecycle modifications occurring inside the external Vendor Portal are instantly intercepted by a centralized tracking interface.
2. Event Streaming: The Barcode Trail Log triggers an active Apache Kafka Producer, which serializes the lifecycle event into a standardized JSON payload and publishes it to a dedicated Kafka topic.
3. Immutable Persistence: A downstream Kafka Consumer Service reads the continuous stream and writes the logs into an append-only database. This establishes an unalterable audit ledger of every structural system transition, ensuring full regulatory compliance.

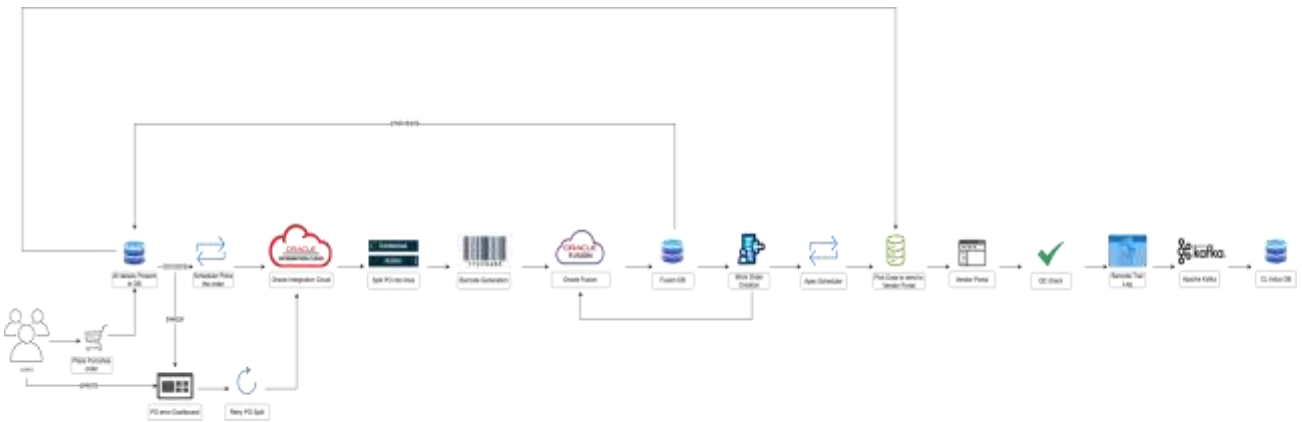

Fig. 3. Architecture encompassing Presentation, Middleware, ERP, and Audit Layers.

V. IMPLEMENTATION DETAILS

The proposed framework was physically deployed using a enterprise software suite: an Oracle Database for staging, Oracle APEX for the administrative frontend, Oracle Integration Cloud for API orchestration, Oracle Fusion SCM for fulfillment, and Apache Kafka for event logging.

A. Mock Data Ingestion and Testing

To validate structural performance prior to live production deployment, synthetic purchase orders reflecting complex, high-mix production demands were programmatically ingested into an isolated Oracle database staging table. These transactional entries were embedded with purposeful data defects and schema anomalies generated via PL/SQL procedures to actively trigger ERP validation rejections. This setup allowed researchers to observe system detection thresholds and monitor the exact data recovery behavior during bulk user retry operations.

B. Middleware Routing and API Bridging

Cross-layer communication was restricted to standardized REST and SOAP web services. OIC integration logic was configured with event-driven triggers; upon the REST adapter receiving an error resolution payload from Oracle APEX, structural data mapping was executed immediately, allowing the remediated data to bypass the standard boundary constraints that initially blocked fulfillment.

C. Real-Time Tracking Integration

End-to-end traceability was achieved by binding the external Vendor Portal directly to a distributed Kafka cluster. The custom Barcode Trail Log was programmed to catch any state changes on the portal. Upon detecting a change, it automatically formats a message and pushes it to a designated Kafka topic. This ensures that critical compliance telemetry is safely captured even during peak operational volume.

VI. RESULTS AND DISCUSSION

A. Accelerated Incident Recovery

Under the legacy operational model, resolving a standard purchase order splitting failure required escalating manual IT support tickets and executing direct database interventions. Consequently, erroneous transactions remained stalled for approximately 24 to 48 hours, causing prolonged vendor idle time. Shifting the exception-handling architecture onto the OIC middleware layer and deploying the web-based Oracle APEX dashboard centralized transactional visibility. This streamlined interface enabled non-technical personnel to execute bulk retry operations, successfully reducing the average mean time to recovery (MTTR) from several hours down to a few minutes.

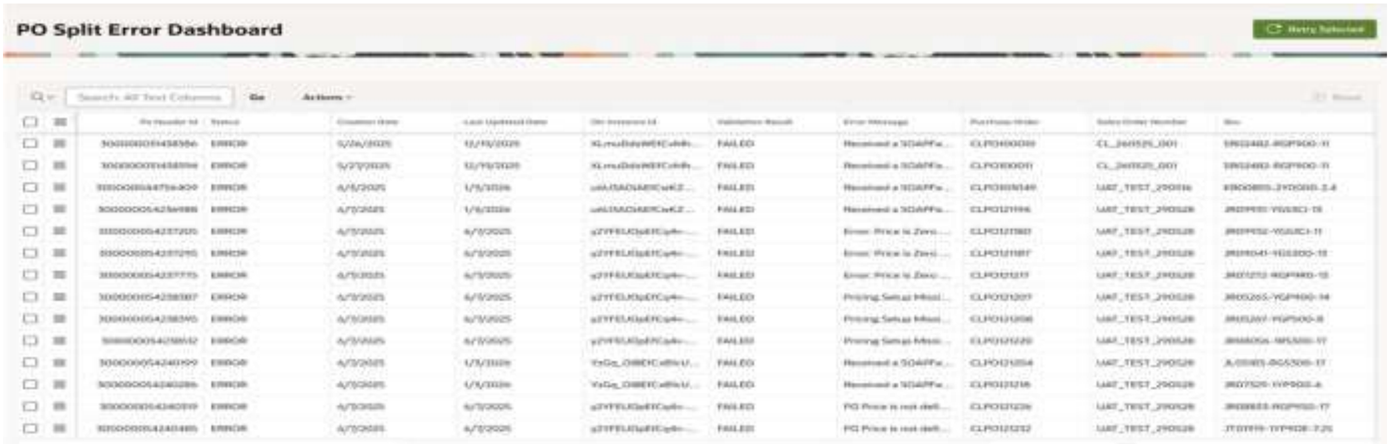

Fig. 3. PO Split Error Dashboard With Retry-Sync Functionality

B. Mitigation of the Ripple Effect

By trapping structural validation anomalies immediately at the middleware boundary, integration failures are contained before propagating downstream into production networks. Barcode data and manufacturing schedules are transmitted to external vendors without latency, maintaining optimized production flow and preventing forecasting errors caused by data misalignment.

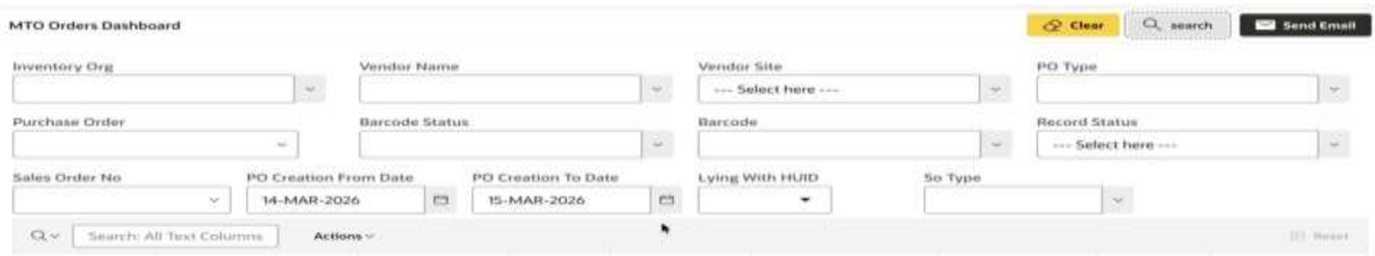

Fig. 4. Vendor Portal for Vendor View of Purchase Order

C. Immutable Audit Compliance

Integrating Apache Kafka alongside the centralized Barcode Trail Log guarantees total tracking coverage. Every transactional interaction—from initial failure to final approval—is written to an append-only log, establishing an unalterable record of all system transitions. This setup eliminates traditional tracking gaps caused by unlogged manual database overrides and ensures a verified chain of custody for all inventory assets.

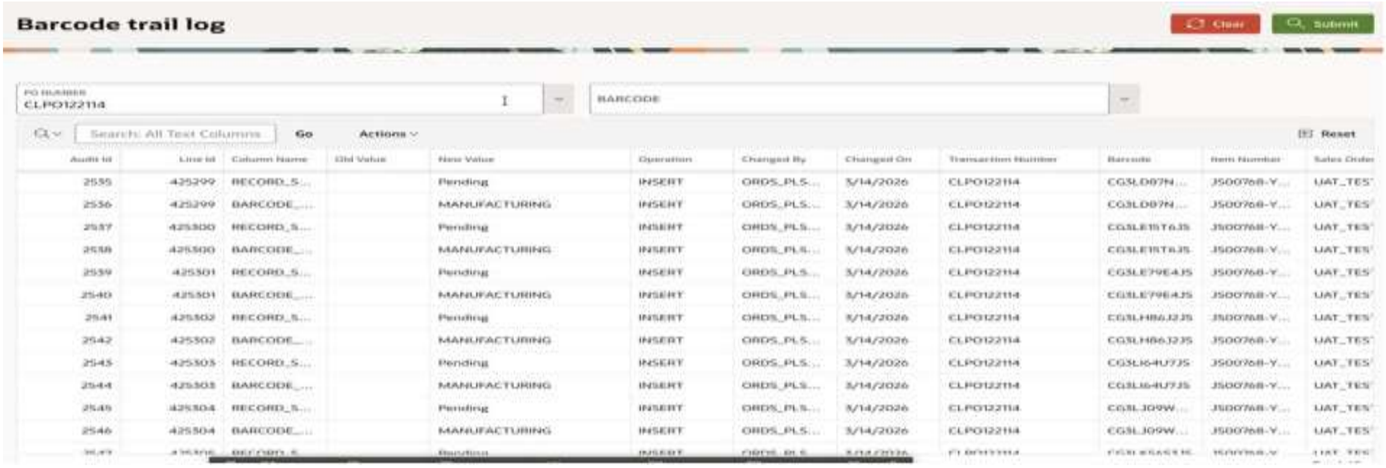

Fig. 5. Barcode Trail Log Status Modification from the Vendor Portal

VII. CONCLUSION

This study successfully validated a decoupled, 4-tier cyber-physical framework engineered to resolve critical purchase order exception handling failures within modern manufacturing networks. By leveraging Oracle Integration Cloud for middleware orchestration, the framework establishes a self-healing operational pipeline that eliminates the systemic delays and production bottlenecks associated with legacy batch architectures. The deployment of an intuitive Oracle APEX diagnostic interface successfully decentralizes troubleshooting capabilities, allowing non-technical operational teams to resolve data exceptions independently without requiring direct database administrator intervention. Backed by real-time event streaming via Apache Kafka and an append-only Barcode Trail Log, the framework ensures absolute data integrity and immutable compliance verification across the entirety of the industrial supply chain.

REFERENCES

- **[1]** D. Ivanov, "Conceptual and formal models for design, adaptation, and control of digital twins in supply chain ecosystems," *Omega*, vol. 137, p. 103356, 2025.
- **[2]** H. Liu, D. Shen, M. Dabić, and J. Lu, "A Novel Methodology for Risk Assessment Considering Risk Higher Order Interactions and Propagation Effects," *IEEE Transactions on Engineering Management*, vol. 72, pp. 907-924, 2025.
- **[3]** F. Li, S. Lang, and Y. Tian, "A Transformer-Based Deep Reinforcement Learning Approach for Dynamic Parallel Machine Scheduling," *Journal of Intelligent Manufacturing*, vol. 36, no. 2, pp. 112-128, 2025.
- **[4]** R. Berger, R. Wagner, P. M. Dion, and O. Matthias, "Disrupting Disruptions: Enhancing Supply Chain Resilience," *Annals of Operations Research*, vol. 347, pp. 1163-1192, 2025.

- **[5]** M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," *Transdisciplinary Perspectives on Complex Systems*, pp. 85-113, 2017.
- **[6]** F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405-2415, 2019.
- **[7]** L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia CIRP*, vol. 17, pp. 9-13, 2014.
- **[8]** N. Garg, "Apache Kafka 101: A Beginner's Guide to Real-Time Data Streaming," *Journal of Software Engineering and Applications*, 2023.