# An Intelligent ERP Framework for Foundry Inventory Optimization Using Human-Centric AI

**J. Nevil Prabhu, Dr. G. Maria Priscilla**

PG & Research ,Asso. Professor

HOD of Computer Science

Sri Ramakrishna College of Arts and Science

Coimbatore – 641006

nevilprabhu@gmail.com
mariapriscilla@srcas.ac.in

## Abstract

Foundry manufacturing environments require continuous synchronization between material availability, production planning, and procurement operations. Conventional inventory control mechanisms and generalized enterprise resource planning (ERP) platforms often lack adaptability to foundry-specific operational constraints such as heat-based traceability, fluctuating material consumption, and delayed replenishment decisions. This paper introduces an intelligent ERP framework tailored for foundry inventory optimization, combining real-time transaction processing, rule- assisted intelligence, automated alerting, conversational system interfaces, and visual analytics. The system employs a modular service-oriented backend architecture and a lightweight web-based frontend to ensure scalability and usability. Human-in-the- loop intelligence is embedded through assisted reorder logic and interactive dashboards, aligning the framework with Industry 5.0 principles. Experimental observations indicate enhanced inventory accuracy, reduced operational latency, and improved managerial responsiveness.

## Keywords

## 1. Introduction

### 1.1 Manufacturing Context

Foundries constitute a core segment of industrial manufacturing, producing essential cast components used in transportation, infrastructure, energy systems, and heavy machinery. These operations depend on uninterrupted access to raw materials such as ferrous and non- ferrous metals, alloying elements, moulding media, and auxiliary consumables. Any disruption in material flow directly impacts productivity and delivery commitments.

## 1.2 Challenges in Inventory Governance

Inventory governance within foundries is inherently complex due to material diversity, consumption variability, and traceability requirements. Many facilities continue to rely on fragmented digital tools or manual processes that fail to provide synchronized, real-time insight. While commercial ERP solutions exist, they often impose high customization costs and do not adequately reflect foundry workflows.

## 1.3 Transition Toward Industry 5.0

Industry 5.0 shifts the focus from pure automation toward collaborative intelligence, ethical system behavior, and resilience. ERP platforms in this paradigm must serve as decision companions—enhancing human judgment through analytics and contextual awareness rather than enforcing autonomous actions.

## 1.4 Contribution of This Work

This paper presents a redesigned ERP framework developed specifically for foundry inventory management. The framework integrates intelligent monitoring, transaction transparency, conversational access, and visual insight to support operational and strategic decisions.

# 2. System Architecture and Operational Workflow

## 2.1 Architectural Design Philosophy

The system is structured using a layered architecture to isolate user interaction, business processing, and data persistence. This approach ensures maintainability, extensibility, and fault isolation.
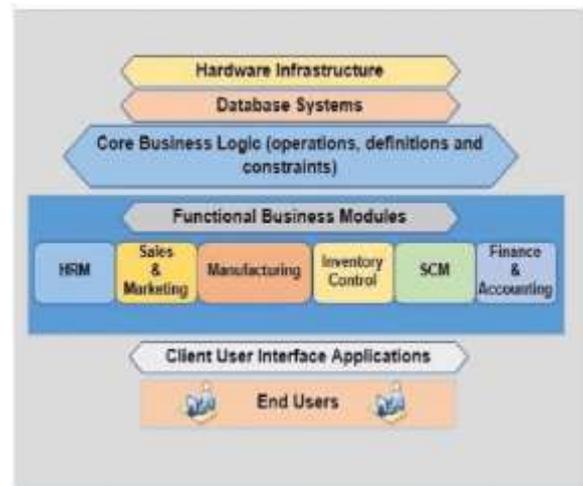
## 2.2 User Interaction Layer

The interaction layer delivers web-based access through dashboards, tables, notification panels, and a conversational assistant. Interfaces are optimized for rapid comprehension and minimal navigation overhead.

## 2.3 Business Processing Layer

The processing layer is implemented using a high-performance API framework that manages inventory computation, rule evaluation, alert generation, and conversational intent handling. Each functional domain operates as an independent service, promoting modular evolution.

## 2.4 Data Persistence Layer

A relational database schema supports transactional integrity and historical traceability. All inventory changes are recorded as immutable events, enabling auditability and rollback analysis.

## 2.5 End-to-End Workflow

Operational workflows begin with authenticated user actions, followed by transaction validation, data persistence, rule evaluation, and real-time UI updates. Conversational queries follow a parallel processing path through intent resolution and data retrieval.

# 3. Implementation Methodology

## 3.1 Backend Engineering

The backend employs an asynchronous web framework to expose RESTful services. Database operations are abstracted using an object–relational layer, ensuring schema consistency and cross-platform compatibility.

## 3.2 Frontend Engineering

Client-side functionality is implemented using modern web standards. Dynamic rendering allows inventory views, alerts, and analytics to update instantly without page reloads.

## 3.3 Security and Role Enforcement

User identity is verified using token-based authentication, while authorization rules restrict access to sensitive operations based on role assignments.
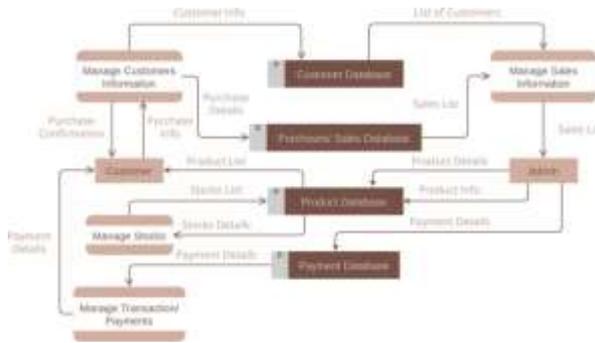
## 3.4 Conversational Interface Design

A deterministic conversational engine processes structured natural-language inputs related to inventory inquiries, alerts, and operational summaries. This interface reduces dependency on complex navigation paths.
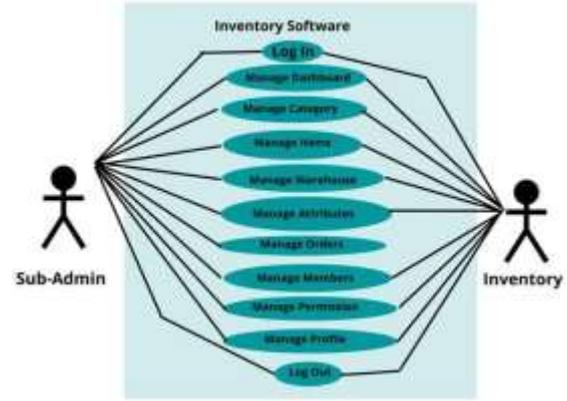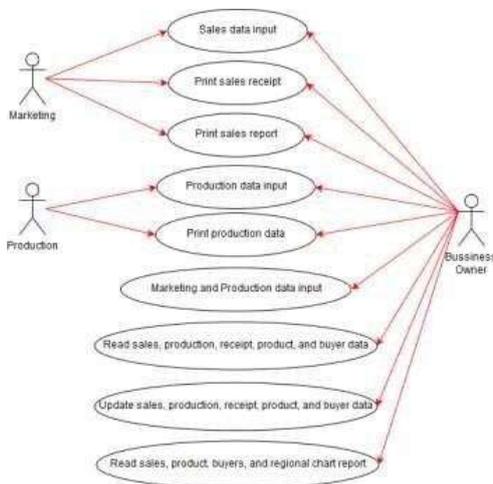
# 4. Data Modeling and Process Representation

## 4.1 Data Flow Modeling



Data flow representations illustrate how inventory data propagates through transaction services, rule evaluators, alert engines, and reporting components.

## 4.2 Use Case Representation





Distinct user roles interact with system capabilities such as stock handling, monitoring, approval workflows, and analytical review.

# 5. Experimental Results and Evaluation

## 5.1 Evaluation Environment

The framework was evaluated using synthetic datasets emulating real foundry operations, including multi-material inventories and variable consumption patterns.

## 5.2 Accuracy and Consistency

Inventory values reflected immediate consistency following each transaction, demonstrating the effectiveness of event- based updates.

## 5.3 Alert Effectiveness
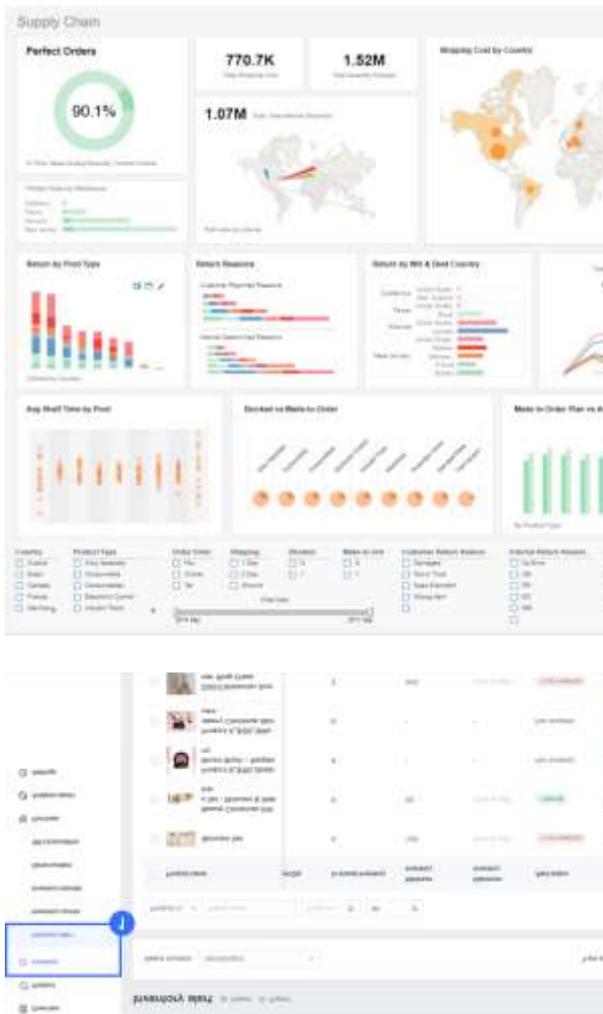




Risk conditions triggered timely notifications, allowing intervention before operational disruption.

## 5.4 Analytical Visibility

Graphical analytics enabled early identification of abnormal consumption trends and emerging shortages.

## 6. Discussion

### 6.1 Operational Impact

The framework improved situational awareness and reduced dependency on manual oversight. Decision latency decreased significantly compared to legacy practices.

### 6.2 Human-in-the-Loop Intelligence

Instead of autonomous control, intelligence components provide recommendations and contextual cues, preserving human authority and accountability.

### 6.3 Scalability and Adaptability

The modular architecture supports expansion to additional plants, materials, and analytics modules without system redesign.

## 7. Applications and Use Cases

### 7.1 Foundry Deployment Scenarios

The framework supports raw material control, production consumption tracking, and procurement coordination across single or multi-plant foundry operations.

### 7.2 Cross-Domain Adaptation

With minor adjustments, the architecture can serve other resource-intensive industries such as forging, metal fabrication, and process manufacturing.

## 8. Future Work

Planned extensions include predictive demand modeling using machine learning, supplier-side system integration, mobile access, and sustainability metrics.

# 9. Conclusion

This paper presented a newly articulated intelligent ERP framework for foundry inventory optimization. Through real-time data processing, transparent transaction handling, conversational access, and analytical insight, the framework addresses limitations of traditional ERP deployments. The results demonstrate improved operational resilience, decision support, and system usability, positioning the framework as a practical foundation for Industry 5.0-aligned manufacturing systems.

# References

[1] European Commission, *Industry 5.0: A Human-Centric Industrial Transformation*, 2021.

[2] S. Chopra and P. Meindl, *Supply Chain Management*, Pearson, 2019.

[3] I. Sommerville, *Software Engineering*, Pearson, 2016.

[4] FastAPI Documentation, https://fastapi.tiangolo.com

[5] SQLAlchemy Documentation, https://www.sqlalchemy.org