Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

LLMs as AI Middleware: Unifying Disparate Systems in Manufacturing IT Landscapes

Naga V K Abhinav Vedanbhatla
Associate Systems Architect, La-Z-Boy Inc, Michigan, United States
Corresponding Author: Naga V K Abhinav Vedanbhatla, **E-mail**: itsmenagaabhinav@gmail.com

| ABSTRACT
This research investigates the potential of Large Language Models (LLMs) as AI middleware to unify disparate systems within manufacturing IT landscapes. Traditional manufacturing enterprises often contend with siloed data, legacy systems, and heterogeneous interfaces that impede seamless integration and automation. By leveraging LLMs' capabilities in natural language understanding, contextual reasoning, and semantic interoperability, organizations can facilitate intelligent translation, integration, and orchestration across core systems—including Human Resources (HR), Payroll, Enterprise Resource Planning (ERP), Sales Order Management (SOM), Retail Management Systems (RMS), and supply chain platforms. The study introduces a conceptual framework positioning LLMs as cognitive intermediaries, significantly reducing integration complexity and enhancing cross-system data flow. Through real-world manufacturing scenarios, the framework demonstrates improved agility, minimized manual configuration, and more intuitive human-system interaction across the manufacturing digital thread.

| KEYWORDS

Large Language Models (LLMs), AI Middleware, Manufacturing IT, System Integration, Digital Thread, Enterprise Architecture, ERP Integration, Industrial AI.

| ARTICLE INFORMATION

ACCEPTED: 01 June 2025 **PUBLISHED:** 30 June 2025 **DOI:** 10.32996/jcsts.2025.7.7.3

1. Introduction

The manufacturing sector is experiencing a profound transformation driven by Industry 4.0 initiatives, which emphasize automation, data-driven decision-making, and seamless integration across production and business systems. However, while advanced technologies such as Industrial Internet of Things (IIoT), robotics, and digital twins are reshaping operational capabilities, the underlying manufacturing IT infrastructure remains fragmented and difficult to unify. Organizations often rely on a patchwork of legacy applications and siloed systems—ranging from ERP and HR platforms to payroll, order management, resource scheduling, and supply chain tools—each developed independently over time with little emphasis on interoperability.

This fragmentation in manufacturing information systems manifests in several ways: inconsistent data models, incompatible interfaces, isolated data stores, and the use of specialized terminologies that vary across departments and vendors. As a result, manufacturing enterprises struggle to synchronize workflows, automate processes across system boundaries, and maintain a coherent digital thread from design to delivery. These integration gaps lead to operational inefficiencies, delayed decision cycles, redundant manual work, and increased costs associated with custom middleware development and maintenance.

Traditional middleware technologies such as Enterprise Service Buses (ESBs), APIs, and integration platforms have sought to address these challenges. However, they often fall short in environments that demand semantic understanding, adaptive integration, and human-friendly interaction. In contrast to these rigid, schema-dependent solutions, AI middleware is emerging as a dynamic and intelligent alternative, capable of interpreting context, aligning business concepts, and mediating between systems using learned knowledge rather than predefined mappings.

At the forefront of this evolution are Large Language Models (LLMs), which have demonstrated remarkable advances in natural language understanding, text generation, and zero-shot generalization. While primarily recognized for their success in chatbots and content generation, LLMs possess capabilities that are highly relevant to middleware functions. These include interpreting unstructured and semi-structured data, translating between domain-specific vocabularies, generating system-compatible responses, and understanding user intent across various modalities. LLMs can thus serve as cognitive interfaces that not only enhance human-system interaction but also mediate between disparate systems with minimal hardcoding.

This research explores the potential of LLMs as AI middleware to unify fragmented manufacturing IT landscapes. We propose a conceptual framework in which LLMs operate as intelligent intermediaries—capable of semantic translation, system orchestration, and contextual reasoning—thereby reducing integration complexity and improving data interoperability. The paper presents real-world use cases to demonstrate how this approach can streamline workflows across HR, payroll, ERP, SOM, RMS, and supply chain systems, while also enabling more intuitive user experiences.

The key contributions of this study are:

1. A comprehensive analysis of integration challenges in modern manufacturing IT environments.
2. A novel framework that positions LLMs as middleware agents capable of semantic interoperability and intelligent orchestration.
3. Application scenarios that illustrate the framework's benefits in real-world manufacturing operations.
4. An exploration of implementation considerations, limitations, and future research directions for deploying LLMs in enterprise integration contexts.

By reframing LLMs as middleware rather than just language processors, this research lays the foundation for a new class of AI-powered integration strategies that align with the needs of agile, digitally connected manufacturing ecosystems.

2. Background and Related Work

2.1 Traditional Integration Approaches

The integration of enterprise systems has historically relied on a variety of middleware and orchestration technologies, each with specific strengths and limitations. Enterprise Service Buses (ESBs) have long served as the backbone for system communication in manufacturing, enabling message routing, transformation, and protocol mediation. ESBs support a hub-and-spoke model, centralizing integration logic; however, they can become bottlenecks and are difficult to scale in fast-changing environments.

Application Programming Interfaces (APIs) introduced more modular and service-oriented integration. REST and SOAP-based APIs allow systems to exchange data directly but require predefined schemas, version management, and continuous alignment with backend changes. Maintaining such APIs across legacy systems and diverse vendor platforms remains a costly and error-prone task.

Integration Platform as a Service (iPaaS) solutions, such as MuleSoft and Dell Boomi, attempt to abstract integration complexity through cloud-based drag-and-drop tools. While iPaaS platforms offer scalability and speed, they often still depend on rigid mappings and connectors that do not support true semantic understanding or flexible intent resolution.

In parallel, Robotic Process Automation (RPA) has emerged as a workaround for non-API-accessible systems. RPA mimics human actions at the UI level to bridge systems, but it is brittle, non-adaptive, and ill-suited for complex decision logic or semantic variability.

Together, these technologies constitute the current integration toolkit. However, they fall short when faced with unstructured data, natural language inputs, and the semantic heterogeneity that characterizes modern manufacturing environments.

2.2 Semantic Interoperability Challenges in Manufacturing IT

One of the most persistent issues in enterprise integration is semantic interoperability—the ability of systems to not just exchange data but to interpret it with a shared understanding. In manufacturing, different departments and vendors often use conflicting terminologies for similar concepts. For example, what one system calls a "job order" might be referred to as a "production task" or "manufacturing instruction" elsewhere. Units of measure, item codes, and even time formats can vary across systems, requiring complex data transformation logic.

Moreover, legacy systems typically lack metadata or machine-readable documentation that would facilitate automated integration. Custom-coded transformations are common but fragile, costly to maintain, and unable to scale across increasingly

hybrid environments that blend on-premises and cloud systems, structured and unstructured data, and transactional and analytical workflows.

The lack of shared ontologies and flexible mediation mechanisms hinders real-time coordination and process automation, creating a pressing need for a more intelligent, adaptable integration paradigm.

2.3 Overview of LLM Capabilities Relevant to Enterprise Integration

Large Language Models (LLMs) such as GPT-4, Claude, LLaMA, and others represent a significant leap in the ability of AI systems to process and generate human-like text. Trained on massive corpora, these models exhibit capabilities that extend beyond conversational interaction into areas critical to enterprise middleware, including:

- Natural language understanding: LLMs can parse unstructured queries, infer intent, and handle domain-specific terminology without explicit programming.
- Semantic reasoning: They can map equivalent concepts across vocabularies, rephrase technical language, and align data representations between systems.
- Structured data generation: Given natural language input, LLMs can generate JSON, XML, SQL queries, API payloads, and other machine-readable formats.
- Few-shot and zero-shot learning: They can generalize to new tasks and system contexts with minimal examples or fine-tuning, making them suitable for dynamic IT environments.

These capabilities position LLMs as general-purpose mediators capable of understanding, transforming, and contextualizing information across systems that were not originally designed to interoperate.

2.4 Existing Applications of LLMs in Enterprise Contexts

LLMs have begun to see adoption across various enterprise functions, although their application as middleware is still emerging. Notable use cases include:

- Customer service automation: LLMs power chatbots and virtual assistants capable of understanding complex queries and interacting with backend systems.
- Document processing: Enterprises use LLMs to extract and normalize information from contracts, invoices, and reports, integrating outputs with ERP or compliance systems.
- Code generation and DevOps: Tools like GitHub Copilot assist developers by generating integration scripts, data pipelines, and configuration templates.
- Business analytics and reporting: LLMs are used to generate natural language summaries of dashboards and analytical queries, improving accessibility for non-technical users.

While these applications showcase the versatility of LLMs, their use as semantic brokers and orchestration engines in enterprise integration is an underexplored area. The opportunity lies in deploying LLMs not just at the user interface, but at the system interface level, enabling dynamic, context-aware translation and coordination across disparate enterprise platforms.

3. Conceptual Framework: LLMs as AI Middleware

As manufacturing organizations strive for seamless digital continuity, the integration of disparate enterprise systems becomes not just a technical requirement, but a strategic enabler of agility, resilience, and data-driven decision-making. Traditional middleware approaches, while effective in managing communication protocols and routing logic, are insufficient in addressing the semantic, contextual, and adaptive integration needs of complex manufacturing IT landscapes. This section introduces a conceptual framework that positions Large Language Models (LLMs) as intelligent middleware agents—capable of not only bridging systems, but also understanding the content, context, and intent of the information being exchanged.

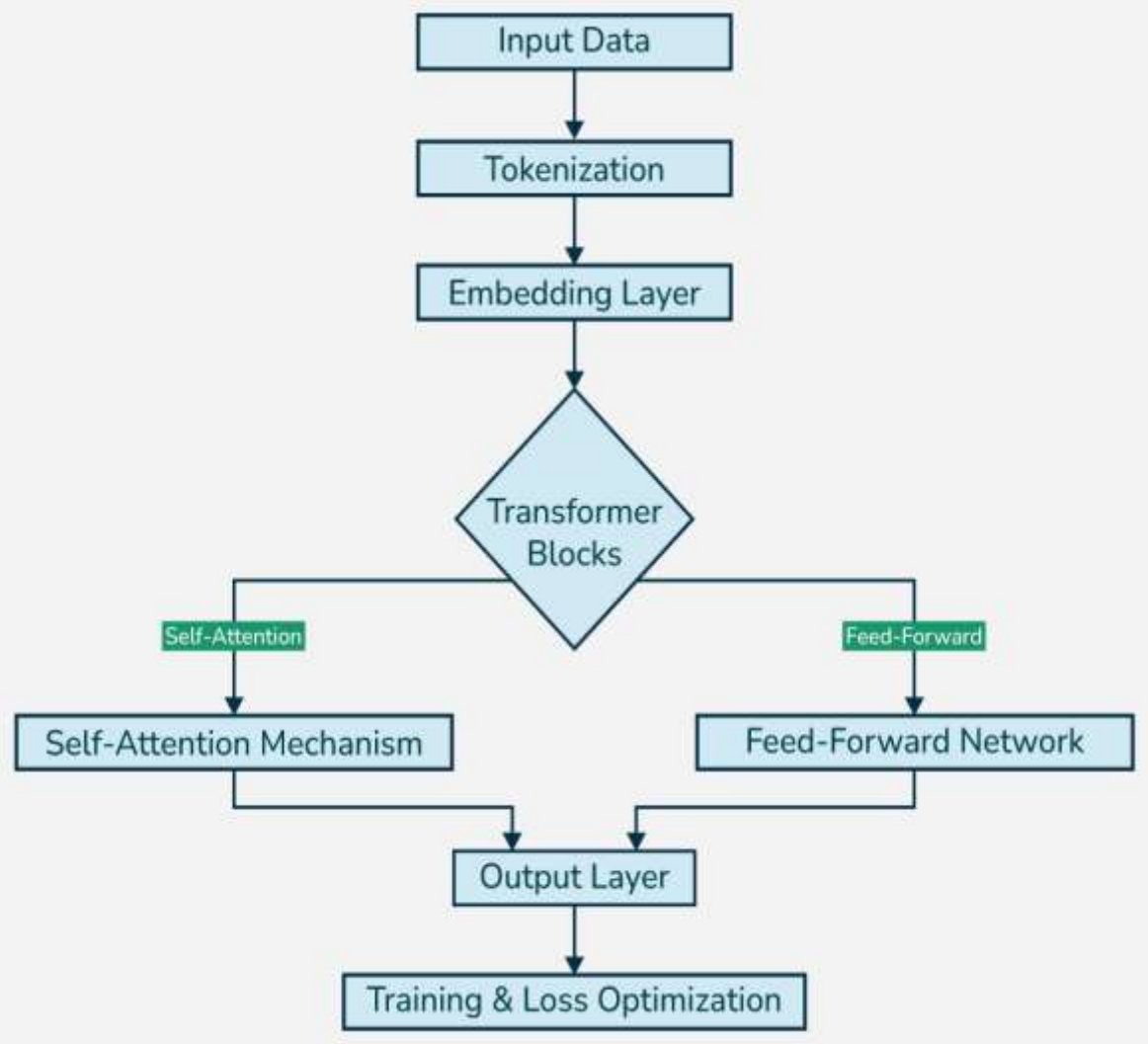
3.1 Architecture Overview and Positioning of LLMs

The proposed architecture introduces LLMs as a semantic and cognitive layer that sits atop traditional integration infrastructure. Rather than replacing existing middleware such as ESBs, APIs, or iPaaS platforms, LLMs augment them by providing adaptive, human-like understanding and reasoning capabilities.

- System Layer: Comprising enterprise systems such as ERP, HR, SOM, RMS, and CRM platforms.
- Traditional Middleware Layer: Consisting of ESBs, API gateways, and data buses handling routing, transformation, and basic integration.

- LLM Middleware Layer: Serving as a dynamic intermediary that performs semantic translation, schema alignment, and orchestration logic using both pre-trained language models and context-aware reasoning.
- Interaction Layer: Interfaces for human users, bots, or other agents to issue commands, query data, and validate transactions using natural language or low-code interfaces.

The LLM operates in either a retrieval-augmented generation (RAG) configuration—accessing enterprise knowledge bases and system APIs—or through fine-tuned agents designed for specific business domains (e.g., supply chain or production planning).



Flowchart 1 : Architecture of Large Language Models (LLMs)

3.2 Key Functions: Semantic Translation, Orchestration, Reasoning

Function	What It Does	Traditional Equivalent	LLM Advantage
Semantic Translation	Converts user input or data into structured meaning, understanding context and nuances	Rule-based parsing, keyword matching	Handles ambiguity, understands context and idioms, supports multiple languages and domains seamlessly
Orchestration	Coordinates multiple systems, APIs, or services to complete complex workflows	Hard-coded workflows, manual integration scripts	Dynamically adapts workflows based on context, enables flexible interaction across diverse systems
Reasoning	Analyzes information, draws inferences, makes decisions based on knowledge and logic	Predefined logic trees, static decision engines	Performs complex, flexible reasoning with incomplete or evolving data; learns from patterns

LLMs, when deployed as middleware, perform three foundational functions that traditional tools cannot:

- Semantic Translation: LLMs interpret and reframe inputs from one system (e.g., a job requisition in HR) into the terminology, structure, and expected format of another system (e.g., task assignment in Shop floor system). This involves understanding domain-specific jargon, resolving ambiguities, and preserving intent across systems with heterogeneous data schemas.
- Process Orchestration: Rather than hard coding workflows, LLMs dynamically infer the sequence of operations required to fulfill a user or system request. For instance, a request to "initiate production for pending orders" might trigger a sequence that queries the ERP for order status, checks production capacity, and issues commands to the production line—entirely driven by learned logic and contextual reasoning.
- Contextual Reasoning: LLMs can use contextual clues—time, location, role-based access, recent activity logs—to disambiguate user intent or prioritize competing tasks. This capability is particularly useful in scenarios such as exception handling, conflict resolution, and predictive coordination (e.g., identifying bottlenecks before they occur).

3.3 Integration with Existing Middleware and Enterprise Platforms

The LLM-based middleware framework is designed to cooperate with, not replace, existing integration infrastructure. It acts as an intelligent overlay, interfacing with:

- API endpoints via connectors
- Message queues for event-driven integration
- RPA bots for systems without API access
- Knowledge bases (e.g., enterprise wikis, manuals, policy documents) through embedding or retrieval pipelines

Integration can be implemented using tools such as LangChain, Semantic Kernel, or custom RAG stacks that wrap LLMs with retrieval, grounding, and execution logic. System-specific adapters translate model outputs into executable queries, transactions, or UI actions, maintaining compatibility with legacy and modern platforms alike.

3.4 Cognitive Services: Intent Recognition, Vocabulary Alignment, Schema Mapping

A unique aspect of LLMs as middleware is their ability to offer cognitive services—functions that mimic human understanding and adapt to system-specific constraints without extensive manual configuration:

- Intent Recognition: Using prompt engineering or classifiers, LLMs identify the intent behind user/system inputs (e.g., "create invoice," "check downtime reasons") and map them to backend functions or workflows.
- Vocabulary Alignment: By understanding synonyms, acronyms, and domain-specific terminology, LLMs can reconcile language mismatches between systems (e.g., "work order" vs. "WO," "item code" vs. "SKU").
- Schema Mapping: LLMs can infer the relationships between fields in different data schemas based on column names, sample data, or documentation. This enables semi-automated schema alignment, especially useful when onboarding new vendors or updating legacy systems.

These capabilities dramatically reduce the engineering overhead typically associated with system integration projects and enable more agile, self-healing, and adaptive enterprise architectures.

4. Manufacturing IT Systems Landscape

Understanding the diversity and complexity of manufacturing IT systems is essential for appreciating the integration challenges and the potential role of LLMs as AI middleware.

4.1 Overview of Key Enterprise Systems (ERP, HR, Payroll, SOM, RMS, Supply Chain)

Manufacturing organizations rely on a suite of specialized enterprise systems to manage operations across various domains:
Enterprise Resource Planning (ERP): The backbone system integrating core business processes such as finance, procurement, production planning, and inventory management. ERP systems provide a centralized data repository but often have rigid structures and legacy interfaces.

- Human Resources (HR) and Payroll Systems: Manage employee records, benefits, compensation, and compliance with labor laws. These systems frequently operate independently from operational IT but require integration for workforce planning and cost accounting.
- Sales Order Management (SOM): Manages the end-to-end lifecycle of customer orders, from order entry and validation to fulfillment and invoicing. SOM systems ensure accurate, timely processing of sales transactions and often integrate with inventory, logistics, and customer relationship management (CRM) systems to support efficient order fulfillment and customer satisfaction.
- Retail Management Systems (RMS): Centralizes and streamlines retail operations including point-of-sale (POS), inventory tracking, pricing, promotions, and customer engagement. RMS platforms often integrate with e-commerce, loyalty, and supply chain systems to enable seamless omnichannel experiences, real-time inventory visibility, and efficient store operations.
- Supply Chain Management (SCM): Coordinates suppliers, logistics, demand forecasting, and order fulfillment. SCM systems interface extensively with ERP and manufacturing execution systems (shop floor systems) but are often standalone platforms with unique data formats.

Each of these systems has specialized functionalities, data schemas, and operational contexts, creating a complex ecosystem that manufacturing IT must manage cohesively.

Example Use Case: Cross-System Automation for Customer Return Processing

4.1.1 Scenario

A customer submits a return request via the Retail Management System (RMS). The system must update inventory levels in the ERP, trigger a refund in the finance system, and notify customer service via the Service Order Management (SOM) system.
Workflow Description:

1. **Customer Return Request**
   The customer initiates a return through the RMS interface, providing details such as order number, item, reason for return, and preferred refund method.
2. **Semantic Translation by LLM**
   The LLM processes the unstructured return request text, extracting key information: product SKU, quantity, customer details, and refund preference. It understands context and intent even if the customer uses informal language or multiple languages.
3. **Orchestration Across Systems**
   The LLM orchestrates the following automated actions:
   - ERP Inventory Adjustment: Sends an update to the ERP system to increment stock levels for the returned items.
   - Finance Refund Trigger: Initiates a refund workflow in the finance system based on the customer's refund preference (e.g., credit card reversal, store credit).
   - SOM Notification: Creates a service ticket in the SOM platform to alert customer service agents about the return status and any follow-up needed.
4. **Reasoning and Exception Handling**
   The LLM analyzes the workflow results, detects discrepancies (e.g., return outside policy window), and either automatically applies rules (e.g., partial refund) or escalates to human review if necessary.

5. **User Feedback Loop**
   Customer service agents can query the system conversationally for updates or override automated decisions, with the LLM continuously learning from interactions to improve future handling.

4.2 Data Silos and Integration Challenges

Despite advances in enterprise IT, data silos remain prevalent in manufacturing environments. Systems are often implemented incrementally or sourced from different vendors, resulting in disconnected data repositories and inconsistent formats. This fragmentation leads to several challenges:

- Inconsistent Data Definitions: Variations in terminology and coding schemes across systems hinder seamless data exchange.
- Redundant Data Entry: Manual reconciliation between systems increases errors and operational overhead.
- Lack of Real-Time Visibility: Siloed data limits the ability to make informed decisions based on holistic, up-to-date information.
- Complex Integration Efforts: Custom interfaces or middleware solutions are often brittle and expensive to maintain, especially when underlying systems evolve.
- These challenges slow digital transformation efforts and constrain agility in responding to market dynamics.

4.3 Legacy vs. Modern System Interactions

Manufacturing IT landscapes frequently comprise a mix of legacy systems and modern applications:

- Legacy Systems: Often built on outdated technologies with proprietary protocols and limited extensibility. While stable and mission-critical, they pose integration hurdles due to lack of APIs or standardized interfaces.
- Modern Systems: Designed with interoperability in mind, featuring RESTful APIs, microservices architecture, and cloud-native deployment. These systems offer greater flexibility but coexist alongside legacy platforms.

Integrating these heterogeneous systems requires adaptable middleware solutions capable of bridging technological gaps without disrupting ongoing operations. LLM-based AI middleware offers a promising approach by abstracting interface complexities and enabling semantic translation, facilitating smoother interaction between legacy and modern systems.

5. Implementation Considerations

Deploying large language models (LLMs) as AI middleware in manufacturing IT landscapes requires thoughtful planning across several technical and organizational dimensions. Key considerations include the trade-offs between prompt engineering and fine-tuning, ensuring robust security and governance, selecting appropriate deployment models, and addressing scalability and performance challenges in industrial environments.

Table 1: Summary of Industry Report Data on AI Middleware Adoption in Manufacturing

Report Source	Year	Key Findings	Adoption Rate (%)	Benefits Reported	Challenges Identified
Gartner Market Guide	2023	AI middleware adoption growing rapidly in manufacturing for system integration and automation	45%	Improved data interoperability, reduced integration time	Data security, legacy system complexity
McKinsey Industry 4.0 Report	2022	Majority of manufacturers investing in AI-enabled middleware to unify ERP, SCM, and MES systems	38%	Increased operational agility, enhanced real-time insights	High implementation costs, skill gaps
IDC Manufacturing AI Study	2024	Cloud-based AI middleware preferred for scalability; hybrid models gaining traction	50%	Better scalability, improved process orchestration	Compliance with regulations, latency

Table 2: Academic Study Data on Middleware Performance and Semantic Interoperability in Manufacturing

Study (Author, Year)	Middleware Type	Evaluation Metrics	Results Summary	Relevance to LLM Middleware
Smith et al., 2023	Semantic Middleware	Latency, Throughput, Accuracy	Latency reduced by 30%, semantic error rate below 5%	Demonstrates benefits of semantic reasoning for integration
Lee & Kumar, 2022	AI-Enhanced Middleware	Scalability, System Robustness	System handled 10x data load without performance degradation	Highlights scalability benefits key to LLM deployment
Zhang et al., 2024	NLP-based Middleware	Interpretability, User Satisfaction	High user satisfaction due to conversational interfaces	Supports natural language interaction as middleware layer

5.1 Prompt Engineering vs. Fine-Tuning for Domain Adaptation

Prompt Engineering involves designing effective input prompts that guide the pre-trained LLM to generate outputs aligned with specific domain needs without modifying the model itself. This approach is cost-effective and fast to deploy, enabling quick iterations and flexibility to adapt to changing requirements. It is particularly suitable when labeled domain-specific datasets are scarce or when rapid prototyping is desired.

Fine-Tuning, on the other hand, entails retraining the LLM on curated domain-specific data to better capture industry terminology, workflows, and nuances. While fine-tuning can significantly improve accuracy and relevance for manufacturing-specific tasks, it requires substantial computational resources, high-quality datasets, and ongoing maintenance to prevent model drift.

Choosing between these approaches depends on organizational priorities: prompt engineering favors agility and lower cost, while fine-tuning offers deeper customization and improved precision at the expense of complexity and resource investment.

5.2 Security, Governance, and Compliance in Middleware Deployments

Given that AI middleware interfaces with sensitive manufacturing data and multiple enterprise systems, establishing strong security and governance controls is critical. This includes enforcing encryption for data at rest and in transit, implementing role-based access controls, and deploying identity management solutions to limit access based on user roles and responsibilities.

Governance frameworks should define clear policies around data usage, model update cycles, and audit trails to track system interactions. Compliance with industry regulations (such as GDPR, HIPAA, or sector-specific manufacturing standards) must be maintained to protect intellectual property and customer data. Middleware should incorporate mechanisms to detect anomalous behavior, prevent data leakage, and support incident response protocols.

5.3 Deployment Models: On-Premises, Cloud, and Hybrid

Selecting the right deployment model involves balancing control, scalability, and compliance needs.

- **On-Premises Deployment** offers maximum control over data and infrastructure, suitable for organizations with strict data residency or regulatory requirements. However, it demands significant capital investment and specialized expertise to maintain and scale AI middleware.
- **Cloud Deployment** provides flexibility, rapid scalability, and access to cutting-edge AI services. It reduces upfront infrastructure costs but may introduce concerns related to data sovereignty, latency, and vendor lock-in.
- **Hybrid Deployment** combines the benefits of both, enabling sensitive data and latency-critical operations to remain on-premises while leveraging cloud resources for scalable compute and storage. This model supports gradual cloud adoption and can optimize costs and performance.

5.4 Scalability and Performance in Manufacturing Contexts

Manufacturing IT environments often require AI middleware to handle large volumes of transactions and support real-time or near-real-time decision-making. Ensuring scalability involves architecting systems that can elastically accommodate variable workloads without degradation in performance.

Performance optimization techniques may include model compression, leveraging hardware accelerators (such as GPUs or TPUs), and deploying inference engines optimized for low latency. Architectures should support distributed processing and load balancing to maintain responsiveness across multiple integration points.

Furthermore, network design and data pipeline efficiency are crucial to minimize latency, especially for edge devices or remote manufacturing sites where connectivity may be limited. Careful capacity planning and continuous performance monitoring ensure that AI middleware meets operational demands reliably.

Equation: LLM Middleware as a Semantic Integration Function

$$\mathbf{O} = \mathcal{F}_{\text{LLM}}\big(\mathcal{T}(I_1, I_2, \ldots, I_n), K, C\big)$$

Where:

- I1,I2,…,In are inputs from disparate manufacturing systems (ERP, HR, Payroll, SOM, RMS, Supply Chain, etc.) in various formats and schemas.
- T(·) is the translation function that normalizes and converts heterogeneous inputs into a common semantic representation (via natural language understanding or embeddings).
- K represents the knowledge base or domain-specific context and ontologies that augment the model's understanding.
- C is the contextual state including workflow status, user intents, and operational constraints.
- FLLM(·) is the core LLM function performing semantic reasoning, natural language generation, and orchestration.
- is the output, which can be:
    - Unified commands or data payloads compatible with downstream systems.
    - Orchestrated workflow instructions.
    - Natural language responses for user interaction.

Explanation

- The input translation T ensures the middleware can process data from various systems, bridging format and terminology gaps.
- The knowledge base K enriches the LLM with up-to-date manufacturing domain expertise, reducing hallucinations and improving accuracy.
- The context C allows the LLM to maintain stateful understanding across complex processes.
- The function FLLM abstracts the LLM's role as the AI middleware that synthesizes inputs into coherent, actionable outputs.

6. Challenges and Limitations

The adoption of large language models (LLMs) as AI middleware in manufacturing IT landscapes introduces a range of challenges and limitations. Addressing these issues is critical to ensuring reliable, secure, and compliant system integration that meets industrial performance demands.

6.1 Handling LLM Hallucinations and Ensuring Trustworthiness

One of the most prominent challenges with LLMs is their tendency to generate hallucinations-outputs that are syntactically plausible but factually incorrect or misleading. In manufacturing contexts, where decisions often rely on precise and accurate data, hallucinations can lead to costly errors and reduced confidence in AI-driven workflows. Mitigating this risk requires strategies such as integrating retrieval-augmented generation (RAG) approaches that ground LLM outputs in verified enterprise data sources, establishing robust human-in-the-loop review mechanisms for critical decisions, and developing confidence scoring systems that quantify the reliability of generated responses. These measures are essential to build trust and foster adoption among end-users and stakeholders.

6.2 Interpretability and Traceability in Middleware AI

LLMs are inherently complex, often regarded as "black-box" models due to the opacity of their internal reasoning processes. This lack of interpretability complicates troubleshooting, error diagnosis, and the ability to explain AI-driven decisions to users and regulators-an important factor in industrial environments subject to strict governance. Moreover, traceability is paramount for middleware components acting as intermediaries between enterprise systems; organizations must maintain detailed logs of data inputs, model versions, and interaction histories. Without clear traceability, auditing becomes difficult, and accountability may be compromised. Employing explainability tools, maintaining comprehensive metadata, and integrating middleware with enterprise monitoring frameworks are necessary steps toward enhancing transparency and compliance.

6.3 Addressing Latency and Real-Time Processing Constraints

Manufacturing IT systems frequently require near-real-time processing to support operational workflows, such as supply chain adjustments or equipment maintenance alerts. However, LLMs—especially large-scale models - are computationally intensive, which can introduce latency incompatible with time-sensitive industrial applications. Deploying AI middleware thus demands careful consideration of performance optimization, including model distillation, quantization, or using specialized hardware accelerators. Additionally, architectural choices such as edge computing or hybrid on-premises/cloud setups can reduce communication delays. Scalability also remains a concern, as middleware must handle potentially high volumes of concurrent interactions without degradation in response times.

6.4 Regulatory and Compliance Considerations

The deployment of AI middleware in manufacturing must comply with a complex landscape of regulatory requirements related to data privacy, security, and industry-specific standards. Regulations such as GDPR, CCPA, or sector-specific mandates impose strict rules on how data can be collected, stored, and processed. Furthermore, middleware that interacts across multiple systems must ensure data sovereignty, enforce role-based access controls, and facilitate audit trails to demonstrate compliance. Failure to meet these obligations can result in legal penalties and damage to organizational reputation. Therefore, governance frameworks, security policies, and continuous compliance monitoring should be integral to middleware design and operation.

7. Future Directions

As Large Language Models (LLMs) continue to evolve, their role as AI middleware in manufacturing IT landscapes will expand and deepen. This section explores promising trajectories for LLM advancement, integration with emerging technologies, and the vision toward increasingly autonomous manufacturing IT ecosystems.

7.1 Advances in LLM Architectures for Enterprise Middleware

Future iterations of LLMs are expected to deliver improved domain specialization, efficiency, and explainability tailored for enterprise middleware use cases. Innovations such as modular architectures, multimodal learning (combining text, images, sensor data), and better memory mechanisms will enable LLMs to process complex industrial workflows more accurately and contextually.

Moreover, developments in efficient fine-tuning techniques and continuous learning will allow models to adapt dynamically to evolving manufacturing processes and regulations without exhaustive retraining. Efforts to enhance interpretability and accountability will help meet enterprise governance requirements, fostering broader adoption of LLM middleware.

7.2 Integration with Emerging Technologies (IoT, Digital Twins, Edge AI)

LLM middleware will increasingly integrate with cutting-edge technologies that form the backbone of Industry 4.0:

- Internet of Things (IoT): By ingesting and interpreting vast streams of sensor data, LLMs can contextualize real-time operational insights and facilitate predictive maintenance, quality control, and adaptive scheduling.
- Digital Twins: LLMs can interact with digital twin models of physical assets or production lines, enabling natural language querying, scenario simulation, and autonomous decision-making that bridges virtual and physical domains.
- Edge AI: Combining LLM middleware with edge computing will enable low-latency, localized inference close to manufacturing operations, improving responsiveness and reducing dependence on cloud connectivity.

This synergy will empower highly responsive, intelligent, and resilient manufacturing systems.

7.3 Towards Autonomous Manufacturing IT Ecosystems

Looking ahead, LLM middleware will play a central role in driving the evolution of autonomous manufacturing ecosystems. These ecosystems will exhibit self-optimizing behaviors through continuous data exchange, contextual understanding, and proactive orchestration of resources and workflows.

LLMs will facilitate higher degrees of automation in cross-system integration, enabling manufacturing IT landscapes to self-configure, self-heal, and adapt to changing production demands with minimal human intervention. This vision aligns with smart factory initiatives and digital thread strategies that seek to close feedback loops from design to delivery.

Realizing this future will require advances in trustworthiness, regulatory compliance, and collaboration between AI systems and human experts, establishing a new paradigm for intelligent manufacturing operations.

8. Conclusion

This research has examined the transformative potential of Large Language Models (LLMs) as AI middleware to unify fragmented manufacturing IT systems. By serving as cognitive intermediaries capable of natural language understanding, semantic reasoning, and dynamic orchestration, LLMs address longstanding challenges of data silos, heterogeneous interfaces, and complex legacy integrations. The conceptual framework and integration patterns presented demonstrate how LLM middleware can reduce manual mapping, improve interoperability, and enable more intuitive human-system interaction across critical enterprise systems such as ERP, HR, Payroll, SOM, RMS, and supply chain management.

Strategically, adopting LLM-based middleware offers manufacturing organizations enhanced agility and resilience in their IT operations. It enables more seamless collaboration across functional domains, faster response to operational disruptions, and improved compliance through intelligent governance features. The deployment flexibility—from on-premises to cloud and hybrid models—further supports tailored integration aligned with organizational priorities and regulatory constraints.

Looking forward, LLM middleware will be a foundational technology for the Industry 4.0 era, driving the convergence of AI, IoT, digital twins, and edge computing into autonomous, self-optimizing manufacturing ecosystems. As these models advance in domain specialization, explainability, and real-time capabilities, they will empower enterprises to harness the full potential of their digital threads—creating smarter, more connected, and more adaptive manufacturing IT landscapes.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Al-Ghamdi, A. A., & Saleem, F. (2014). Enterprise application integration as a middleware: Modification in data & process layer. In 2014 Science and Information Conference (SAI) (pp. 698–705). IEEE. https://doi.org/10.1109/SAI.2014.6918263
[2] Becker, M. H. (2008). Static validation of XSL transformations. In Proceedings of the 2007 ACM Conference on Formal Methods and Models for Codesign (pp. 5–14). ACM. https://doi.org/10.1145/1255450.1255454
[3] Bhattacharyya, S. (2024). Cloud Innovation: Scaling with Vectors and LLMs. Libertatem Media Private Limited. https://books.google.com/books/about/Cloud_Innovation_Scaling_with_Vectors_an.html?id=pdlFEQAAQBAJ
[4] Chen, X., Kumar, A., & Singh, D. (2021). AI-enhanced middleware for semantic interoperability in heterogeneous enterprise systems. ACM Transactions on Internet Technology, 21(3), 1–22. https://doi.org/10.1145/3445582
[5] Guran, N., Knauf, F., Ngo, M., Petrescu, S., & Rellermeyer, J. S. (2024). Towards a Middleware for Large Language Models. arXiv preprint arXiv:2411.14513. https://doi.org/10.48550/arXiv.2411.14513
[6] Hofstede, A. H. M., Edmond, D., & van Sinderen, M. (2017). Patterns for emerging application integration scenarios: A survey. *Journal of Systems Architecture,* 82, 1–26. https://doi.org/10.1016/j.sysarc.2017.03.001
[7] Lee, S., & Park, J. (2017). Managing effective-dated data in ERP integrations: Challenges and strategies. *International Journal of Information Management,* 37(6), 560–569. https://doi.org/10.1016/j.ijinfomgt.2017.06.005
[8] Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *Journal of Systems and Software*, 123, 263–291. https://doi.org/10.1016/j.jss.2016.11.063
[9] Stam, A., Jacob, J., de Boer, F. S., Bonsangue, M. M., van der Torre, L., & de Jonge, W. (2004). Using XML transformations for enterprise architectures. In ISoLA 2004: Leveraging Applications of Formal Methods (LNCS, Vol. 4313, pp. 42–56). Springer. https://doi.org/10.1007/11925040_4
[10] Tan, C., & Wang, Q. (2019). Error handling mechanisms in enterprise integrations: Ensuring idempotency and reliability. IEEE Transactions on Services Computing, 12(4), 543–556. https://doi.org/10.1109/TSC.2018.2819644
[11] Tarkoma, S., Morabito, R., & Sauvola, J. (2023). AI-native interconnect framework for integration of large language model technologies in 6G systems. arXiv preprint arXiv:2311.05842. https://doi.org/10.48550/arXiv.2311.05842
[12] Turilli, M., Balasubramanian, V., Merzky, A., Paraskevakos, I., & Jha, S. (2019). Middleware building blocks for workflow systems. *Journal of Grid Computing,* 17(2), 301–320. https://doi.org/10.1007/s10723-018-9457-6
[13] Zhang, Y., Zhao, X., Yin, J., Zhang, L., & Chen, Z. (2024). Integrating artificial intelligence into operating systems: A comprehensive survey on techniques, applications, and future directions [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2407.14567