IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

Volume 14, Issue 1, January 2025

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.514

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

Agentic AI in Enterprise Resource Planning: Design Patterns for Autonomous Process Orchestration across Finance, Logistics, and Trade Compliance Workflows

Srinivas Kakarla

SAP Architect, USA

ABSTRACT: As enterprise resource planning (ERP) landscapes mature beyond dashboards and predictive scoring, a new architectural unit is emerging inside them: the autonomous agent. Unlike a conventional machine learning model that returns a score or a classification, an agentic AI system plans a sequence of steps, calls tools and APIs, evaluates intermediate outcomes, and - within defined permission boundaries - takes action directly against ERP systems of record. This article examines design patterns for embedding such agentic systems into ERP-centric business processes, with particular attention to finance, logistics, and trade compliance workflows, where the cost of an unsupervised error is highest. We present a hub-and-spoke multi-agent reference architecture, contrast centralized and decentralized orchestration topologies, walk through a swimlane-level trade compliance screening scenario, and propose a concentric governance model for constraining autonomous agent action. The article closes with an implementation roadmap, an illustrative risk profile, and a discussion of open questions as they stood at the start of 2025.

KEYWORDS: Agentic AI, Autonomous Agents, Enterprise Resource Planning, Multi-Agent Orchestration, Finance Automation, Logistics, Trade Compliance, Tool-Calling, Retrieval-Augmented Generation, AI Governance

I. INTRODUCTION: FROM PREDICTIVE AI TO AUTONOMOUS AGENTS

The previous wave of AI investment inside enterprise resource planning (ERP) landscapes was largely about prediction: forecasting demand, scoring invoices, flagging anomalies. A human remained the actor - the model informed a decision, but a person clicked the button. Beginning in 2023 and accelerating through 2024, a distinct architectural pattern moved from research demonstrations into early production deployments: the autonomous agent, built around a large language model (LLM) as a reasoning engine, capable of planning a multi-step task, invoking tools and APIs to gather information or take action, and iterating based on the results - with a human positioned as an approver or exception handler rather than the executor of every step.

This shift matters most in ERP-centric processes precisely because those processes are already highly structured, API-addressable, and rule-governed - the conditions under which tool-calling agents perform most reliably. A finance agent that can read an open invoice, check a vendor's payment history via an ERP API, apply a matching rule, and post the transaction is a fundamentally different system from a model that merely predicts whether the invoice looks anomalous. The former takes action; the latter recommends one.

The question enterprises are now asking is no longer whether an AI system can produce a useful answer, but how much autonomy to grant it before a human must be in the loop.

This article focuses specifically on three ERP domains where agentic design patterns were seeing the earliest production traction as of early 2025: finance (accounts payable, close, cash management), logistics (order-to-delivery orchestration, exception handling), and trade compliance (denied-party screening, classification, licensing). These domains share a common trait relevant to agent design: they combine well-defined, rule-governed sub-tasks with a high volume of edge cases that historically required a skilled human to adjudicate - precisely the mix that plays to the strengths, and exposes the risks, of agentic automation.

The remainder of the article proceeds as follows. Section 2 defines what distinguishes an "agentic" system from earlier ERP AI capabilities. Section 3 presents a hub-and-spoke reference architecture for multi-agent ERP orchestration. Section 4 compares centralized and decentralized orchestration topologies. Section 5 walks through a concrete design

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

pattern for autonomous trade compliance screening. Section 6 examines each of the three focal domains in more depth. Section 7 proposes a governance model for constraining agent action. Section 8 offers an implementation roadmap. Section 9 discusses risk and failure modes, Section 10 raises open questions, and Section 11 concludes.

II. WHAT MAKES AN ERP AGENT "AGENTIC"

The term "agentic AI" is used inconsistently across vendor marketing and technical literature. For the purposes of this article, we adopt a narrower, more operational definition grounded in four characteristics that, taken together, distinguish an agentic system from a conventional predictive model or a single-turn generative AI assistant.

Table.1. Distinguishing Agentic AI from Conventional Predictive AI and Single-Turn Generative Assistants

Characteristic	Conventional Predictive AI	Single-Turn GenAI Assistant	Agentic AI System
Planning	None - single inference call	None - one response per prompt	Decomposes a goal into an ordered sequence of steps
Tool use	N/A	Optional, typically read-only retrieval	Calls APIs/tools to read and write ERP data
Iteration	N/A	N/A	Evaluates intermediate results and re-plans
Autonomy boundary	Human acts on model output	Human acts on generated text	Agent acts directly, within defined permissions

2.1 The Agentic Reasoning Loop

Across the implementations reviewed for this article, agentic ERP systems converge on a recognizable four-stage reasoning loop, illustrated in Figure 2: the agent perceives the current state (reading ERP records, recent events, or attached documents), plans a sequence of tool calls needed to achieve its assigned goal, acts by invoking those tools (which may include ERP System APIs, sub-agents, or external data services), and reflects on the outcome - deciding whether the goal has been met, whether to retry with adjusted parameters, or whether to escalate to a human reviewer. A human-in-the-loop gate, shown at the center of the loop, is not an optional add-on in mature implementations; it is where permission boundaries (Section 7) are enforced.

Figure 2. The Agentic Reasoning Loop: Perceive - Plan - Act - Reflect

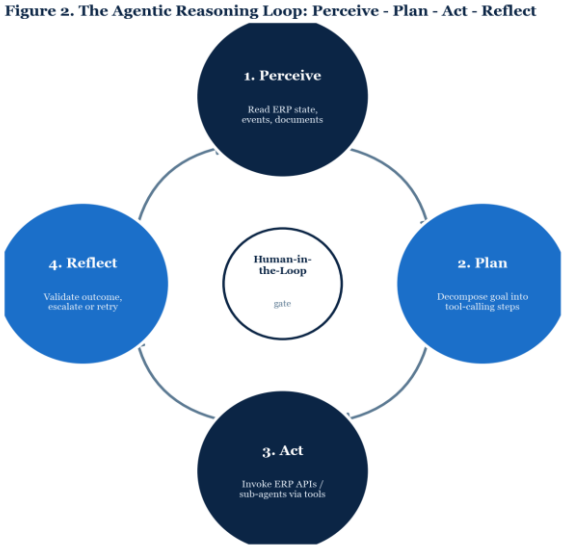


Figure.2. The agentic reasoning loop - Perceive, Plan, Act, Reflect - with a human-in-the-loop gate positioned at the center of the cycle.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

2.2 Tool-Calling as the Integration Surface

The mechanism by which an agent acts on an ERP system is a tool call - a structured invocation of a predefined function or API endpoint, typically described to the underlying LLM via a schema the model uses to decide when and how to call it. This is, in effect, the same API surface that a cloud integration platform would expose to any other consumer, which means many of the API governance disciplines that predate agentic AI (contract versioning, rate limiting, access scoping) remain directly applicable - but must now be applied with the assumption that the calling party is a model whose calling behavior is probabilistic rather than hand-coded.

2.3 Single-Agent vs. Multi-Agent Systems

Early agentic ERP deployments were often single-agent: one LLM-driven agent with a broad tool set spanning a whole domain (for example, "the finance agent"). As scope expanded, a multi-agent pattern emerged in which specialized agents - each with a narrower tool set and domain vocabulary - are coordinated by an orchestrating agent or workflow. Section 3 presents a reference architecture built around this multi-agent pattern, which the case material reviewed for this article suggests scales more predictably as the number of covered business processes grows.

III. REFERENCE ARCHITECTURE: HUB-AND-SPOKE MULTI-AGENT ORCHESTRATION

Figure 1 presents a hub-and-spoke reference architecture recurring across multi-agent ERP deployments examined for this article. An orchestrator agent sits at the center, responsible for decomposing an incoming goal or event into sub-tasks, delegating each sub-task to the appropriate domain agent, and reconciling the results into a coherent outcome. Domain-specific agents - Finance, Logistics, Trade Compliance, and Procurement in the illustration - each carry a narrower tool set and specialized prompting/context tailored to their domain. A shared Tool/API layer exposes the underlying ERP System APIs, event bus, and data services that any agent may call, typically mediated by the same cloud integration platform infrastructure used for non-agentic integration (see our companion article on AI-enabled ERP integration architecture for a fuller treatment of that integration layer).

Figure 1. Hub-and-Spoke Multi-Agent Orchestration Architecture for ERP Process Automation

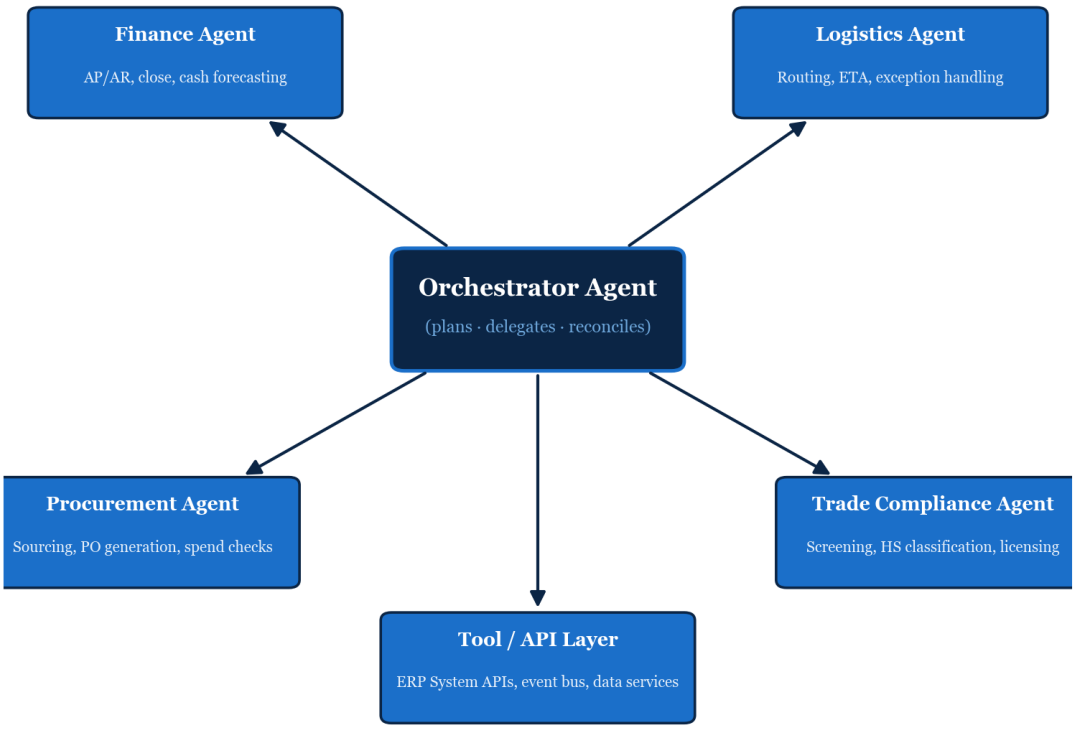


Figure.1. Hub-and-spoke multi-agent orchestration architecture: an orchestrator agent delegates to domain-specific agents, all of which call a shared Tool/API layer.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

3.1 Why the Orchestrator Is a Distinct Role

Separating orchestration from domain execution serves three purposes observed across implementations. First, it provides a single point at which cross-domain consistency is enforced - for example, ensuring a logistics exception and a finance credit hold triggered by the same root event are reconciled rather than acted on independently. Second, it isolates the most complex planning logic in one place, simplifying the prompting and tool design of domain agents. Third, and most relevant to the governance discussion in Section 7, it creates a natural chokepoint for policy enforcement and audit logging that would otherwise need to be duplicated across every domain agent.

3.2 The Shared Tool/API Layer

Domain agents in mature implementations do not typically hold direct, unmediated database credentials to ERP systems. Instead, they call a shared set of governed tools/APIs - the same kind of System API cataloged and versioned by a cloud integration platform. This has an important practical benefit: rate limiting, permission scoping, and observability instrumentation applied at this layer automatically apply to every agent that calls through it, regardless of which LLM or agent framework is driving that agent's reasoning.

IV. ORCHESTRATION TOPOLOGIES: CENTRALIZED VS. DECENTRALIZED

Not every multi-agent ERP deployment adopts the hub-and-spoke topology of Section 3. Figure 4 contrasts two topologies observed in practice: a centralized model, in which a single orchestrator mediates all cross-agent communication, and a decentralized (peer-to-peer) model, in which domain agents communicate directly with one another without a central mediator.

Figure 4. Two Multi-Agent Orchestration Topologies Observed in Enterprise Deployments

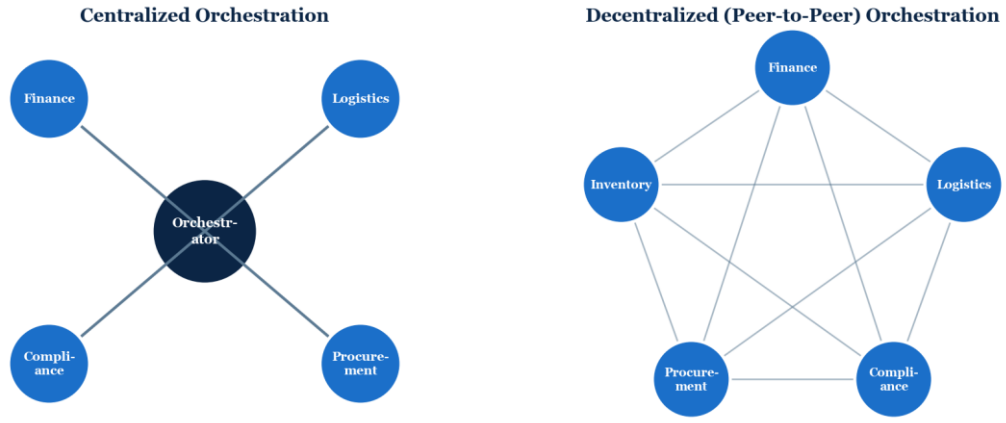


Figure.4. Two orchestration topologies: centralized (all coordination flows through one orchestrator) versus decentralized peer-to-peer (agents communicate directly).

Table.2.Trade-offs Between Centralized and Decentralized Multi-Agent Orchestration Topologies

Dimension	Centralized Orchestration	Decentralized (Peer-to-Peer)
Auditability	High - single point captures full trace	Lower - traces are distributed across agent pairs
Latency for simple tasks	Adds a hop through the orchestrator	Can be lower for direct agent-to-agent needs
Failure isolation	Orchestrator outage affects all flows	More resilient to a single agent's failure
Policy enforcement	Straightforward - one enforcement point	Requires policy logic duplicated per agent
Complexity as agents scale	Orchestrator logic grows but stays centralized	Coordination logic grows combinatorially

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

The case material reviewed for this article suggests that centralized orchestration remains the dominant pattern for ERP-centric agentic systems specifically because auditability and policy enforcement are unusually important in finance, logistics, and trade compliance workflows - domains where regulatory and audit exposure (Section 9) makes a single, traceable decision point highly valuable. Decentralized topologies appear more often in research settings or in narrower, lower-stakes automation scenarios where the coordination overhead of a central orchestrator is not justified by the risk profile of the task.

V. DESIGN PATTERN WALKTHROUGH: TRADE COMPLIANCE SCREENING

To ground the architectural concepts in a concrete scenario, Figure 3 traces a swimlane-level process map for autonomous trade compliance screening - a use case with unusually strict correctness requirements, since an incorrect release of a restricted shipment carries direct regulatory exposure.

Figure 3. Swimlane Process Map: Autonomous Trade Compliance Screening

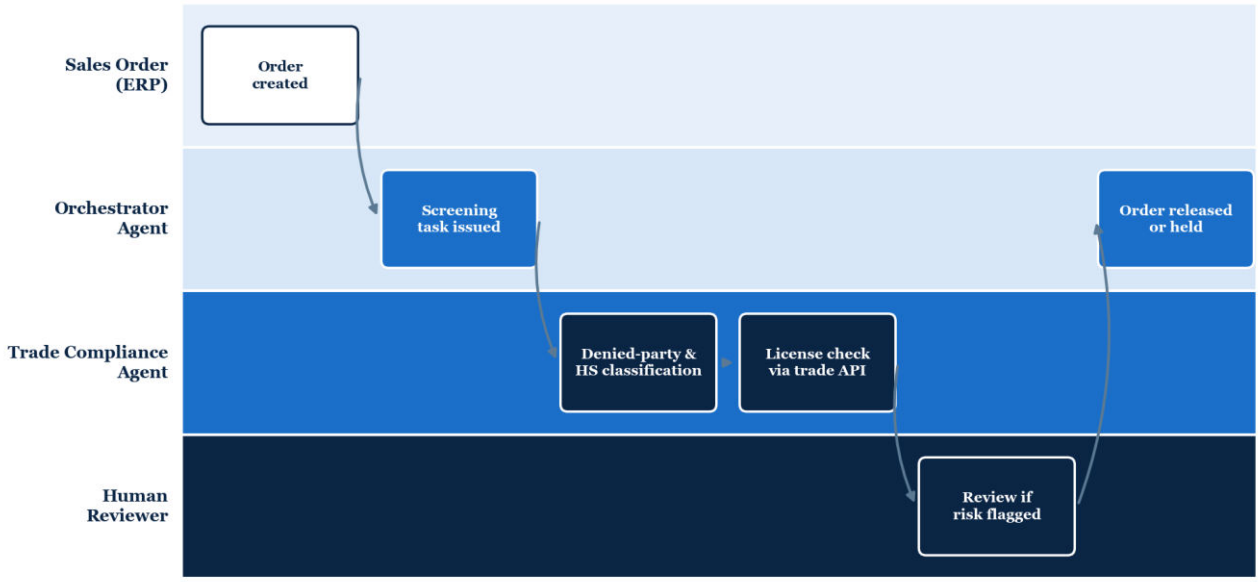


Figure.3. Swimlane process map for autonomous trade compliance screening, spanning the ERP sales order, the orchestrator agent, the trade compliance agent, and a human reviewer lane

5.1 Step-by-Step Walkthrough

1. **Order created.** A sales order is created in the ERP system, triggering an event consumed by the orchestrator agent.
2. **Screening task issued.** The orchestrator agent recognizes that the order requires trade compliance screening (based on destination country, product category, or customer classification) and delegates a screening task to the Trade Compliance Agent.
3. **Denied-party and HS classification.** The Trade Compliance Agent calls tools to check the counterparty against denied- and restricted-party lists and to classify the goods under the applicable Harmonized System (HS) code.
4. **License check via trade API.** Where the classification indicates a controlled item or destination, the agent calls a trade-compliance API to determine whether an export license is required and, if so, whether one is on file.
5. **Review if risk flagged.** If the agent's confidence is below a defined threshold, or if any check returns an ambiguous or high-risk result, the order is routed to a human reviewer rather than being autonomously released.
6. **Order released or held.** The orchestrator agent receives the outcome - clear, held, or reviewed-and-cleared - and updates the ERP order status accordingly, closing the loop.

This pattern illustrates a design principle that recurs across the higher-stakes domains examined in this article: the agent is granted autonomy over the investigatory steps (screening, classification, license lookup) but not, by default, over the final release decision when risk signals are present. Section 7 formalizes this as a governance principle rather than a one-off design choice specific to trade compliance.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

VI. DOMAIN DEEP DIVES: FINANCE, LOGISTICS, TRADE COMPLIANCE

Figure 6 presents an illustrative comparison of relative agentic-AI capability coverage across the three focal domains, scored across six dimensions relevant to agent design. The scoring reflects a qualitative synthesis of documented deployments rather than a measured benchmark, and is intended to highlight relative emphasis rather than absolute performance.

Figure 6. Relative Agentic-AI Capability Coverage by ERP Domain (Illustrative, 1-5 Scale)

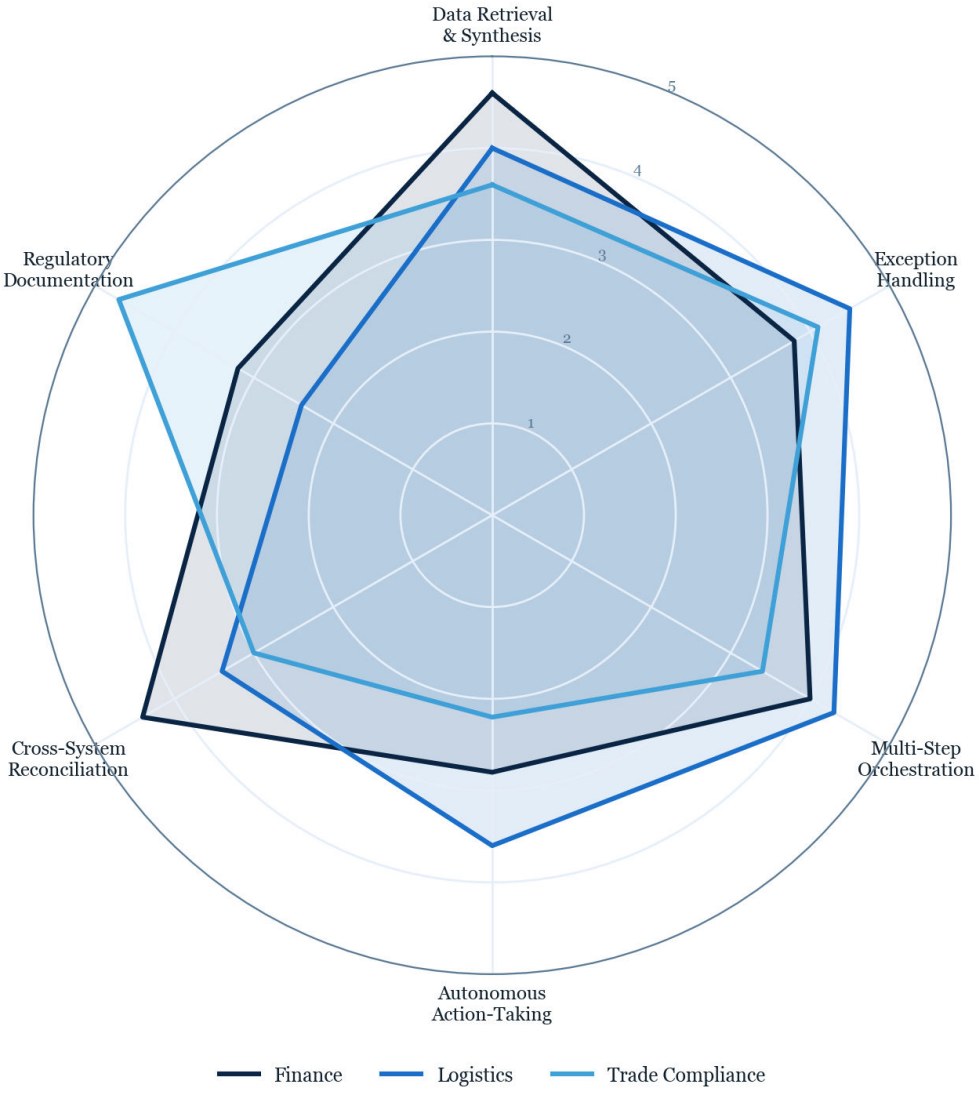


Figure.6. Illustrative radar comparison of agentic-AI capability coverage across Finance, Logistics, and Trade Compliance domains, scored on a 1–5 scale across six dimensions.

6.1 Finance

Finance shows the strongest illustrative coverage in data retrieval and synthesis and cross-system reconciliation - consistent with the domain's reliance on matching and reconciling records across sub-ledgers, bank feeds, and the general ledger. Representative agentic patterns include:

- **Autonomous invoice-to-payment matching.** An agent retrieves an incoming invoice, matches it against a purchase order and goods receipt, applies configurable tolerance rules, and either posts the transaction autonomously or escalates a discrepancy to accounts payable staff.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

- **Agent-assisted close.** During period-end close, an agent executes a checklist of reconciliation tasks across sub-ledgers, flags variances above a threshold, and drafts (but typically does not autonomously post) adjusting entries for controller review.
- **Cash positioning and forecasting narration.** An agent aggregates cash positions across entities and bank accounts, and generates a natural-language narrative of drivers behind forecast changes for treasury review.
- Logistics
  Logistics shows the strongest illustrative coverage in exception handling and multi-step orchestration - consistent with the domain's need to coordinate across carriers, warehouses, and customer commitments in response to frequent real-world disruptions. Representative patterns include:
- **Autonomous exception rerouting.** When a shipment tracking event indicates a delay or failure, an agent evaluates rerouting options, checks cost and service-level implications via carrier and ERP APIs, and executes a rebooking within pre-approved cost thresholds.
- **Order-to-delivery orchestration.** A logistics agent coordinates sequential sub-tasks - inventory allocation, carrier selection, customs documentation preparation - that previously required manual hand-offs between systems and teams.
- **Proactive customer communication.** Upon detecting a delay, an agent drafts and, in more autonomous implementations, sends a customer notification with an updated estimated delivery time, grounded in current shipment data.

6.3 Trade Compliance

Trade compliance shows the strongest illustrative coverage in regulatory documentation - reflecting the domain's dependence on producing auditable, well-documented decisions. Representative patterns include the screening workflow detailed in Section 5, as well as:

- **Automated HS classification with confidence scoring.** An agent classifies goods against the Harmonized System using product descriptions and specifications, surfacing a confidence score that determines whether human review is required.
- **License and permit lifecycle tracking.** An agent monitors license expirations and usage against quota limits, proactively flagging renewal needs before they block a shipment.
- **Audit trail generation.** Because trade compliance decisions are subject to regulatory audit, agents in this domain typically generate a structured decision record - inputs checked, rules applied, outcome - as a first-class output of every screening task, not an afterthought.

Table.3. Domain Comparison: Highest-Coverage Capability, Representative Autonomous Action, and Typical Human Gate

	Highest-Coverage Capability	Representative Autonomous Action	Typical Human Gate
Finance	Cross-system reconciliation	Autonomous invoice-to-payment posting within tolerance	Discrepancies above tolerance threshold
Logistics	Exception handling	Autonomous carrier rebooking within cost threshold	Rebooking cost above pre-approved limit
Trade Compliance	Regulatory documentation	Autonomous screening & classification	Any denied-party match or low-confidence classification

VII. GOVERNANCE: A CONCENTRIC MODEL FOR CONSTRAINING AGENT ACTION

Governing an agentic system differs from governing a predictive model in one crucial respect: the object of governance is not just an output (a score, a classification) but an action taken directly against a system of record. Figure 7 proposes a concentric governance model in which layers of control surround the point of autonomous agent action, each layer narrowing what the agent is permitted to do without triggering the next layer out.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

Figure 7. Concentric Governance Model Surrounding Autonomous Agent Actions

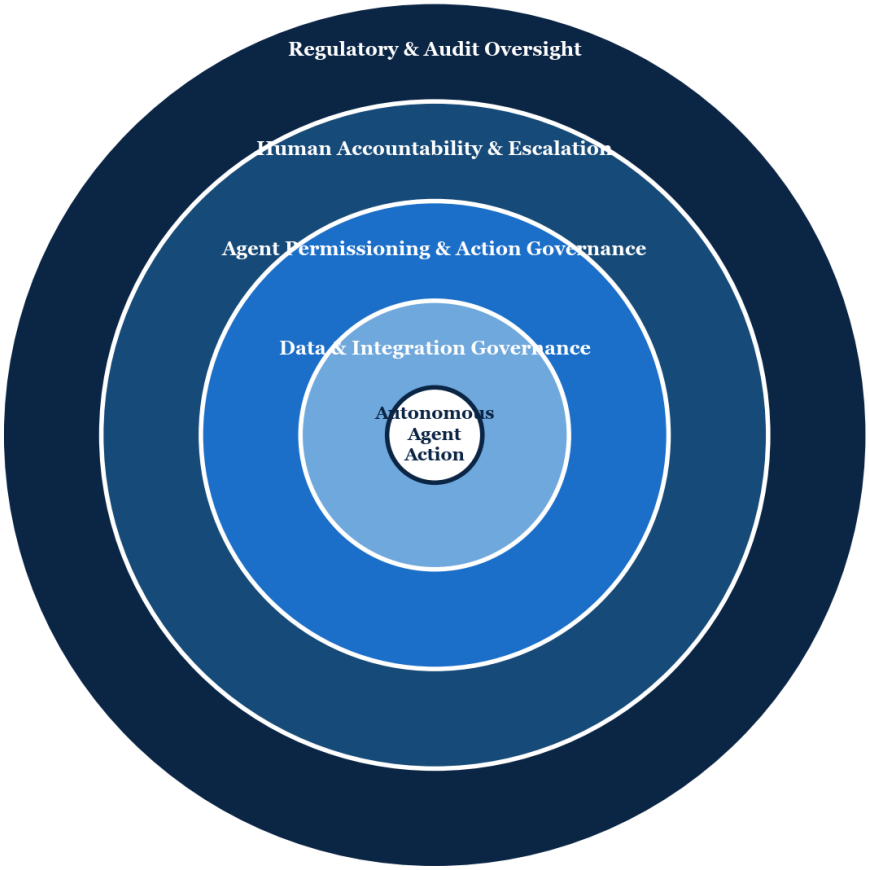


Figure.7. Concentric governance model: data and integration governance forms the innermost ring around autonomous agent action, followed by agent permissioning, human accountability, and regulatory oversight.

7.1 Data and Integration Governance (Innermost Ring)

At the innermost ring, the same data and integration governance disciplines that apply to any ERP integration remain foundational: data lineage, access control, and API contract stability. An agent that calls a governed, well-documented tool inherits the guarantees of that tool; an agent given ungoverned direct data access inherits none of them.

7.2 Agent Permissioning and Action Governance

- **Scoped tool access.** Each agent should be granted the minimum set of callable tools required for its role - a trade compliance agent has no legitimate need to call a payment-posting tool.
- **Action-level thresholds.** As illustrated in Sections 5 and 6, autonomous action is typically bounded by a threshold (a dollar amount, a confidence score, a risk flag) above which the agent must escalate rather than act.
- **Rate and blast-radius limits.** Guardrails should cap how many actions an agent (or a runaway planning loop) can take within a time window, limiting the damage from a misbehaving agent before a human notices.

7.3 Human Accountability and Escalation

Every autonomous action pathway should have a clearly designated human owner accountable for reviewing escalations and, periodically, for auditing a sample of autonomously completed actions - not only the escalated ones - to catch cases where the agent's confidence was miscalibrated rather than genuinely low.

7.4 Regulatory and Audit Oversight (Outermost Ring)

At the outermost ring, the same regulatory and audit considerations discussed in governance frameworks for predictive AI apply with added force to agentic systems, because the artifact under review is now a chain of autonomous actions

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

rather than a single model output. As of early 2025, several jurisdictions' AI-specific regulatory frameworks were still clarifying how existing internal-control and audit requirements apply to autonomous, tool-calling AI systems specifically; enterprises deploying such systems in finance and trade compliance should expect this area to continue developing.

Table.4. The Four Governance Rings, the Question Each Answers, and a Representative Control.

Governance Ring	Primary Question It Answers	Representative Control
Data & integration governance	Is the data the agent sees accurate and access-appropriate?	Data lineage, row/field-level access control
Agent permissioning	What is this agent allowed to do, and up to what threshold?	Scoped tool access, action-level thresholds
Human accountability	Who is accountable when the agent escalates, or when it doesn't?	Designated reviewer roles, sampled audits
Regulatory & audit oversight	Does the full action chain meet applicable regulatory expectations?	Structured decision records, model/agent inventories

VIII. IMPLEMENTATION ROADMAP AND ILLUSTRATIVE VALUE

Figure 8 presents an illustrative 18-month rollout roadmap synthesized from common practice patterns across the case material reviewed for this article, sequencing guardrail development ahead of, and overlapping with, use-case piloting - reflecting a recurring emphasis among practitioners on establishing permissioning and audit infrastructure before, not after, granting an agent write access to production ERP data.

Figure 8. Illustrative 18-Month Rollout Roadmap for Agentic AI Across ERP Workflows

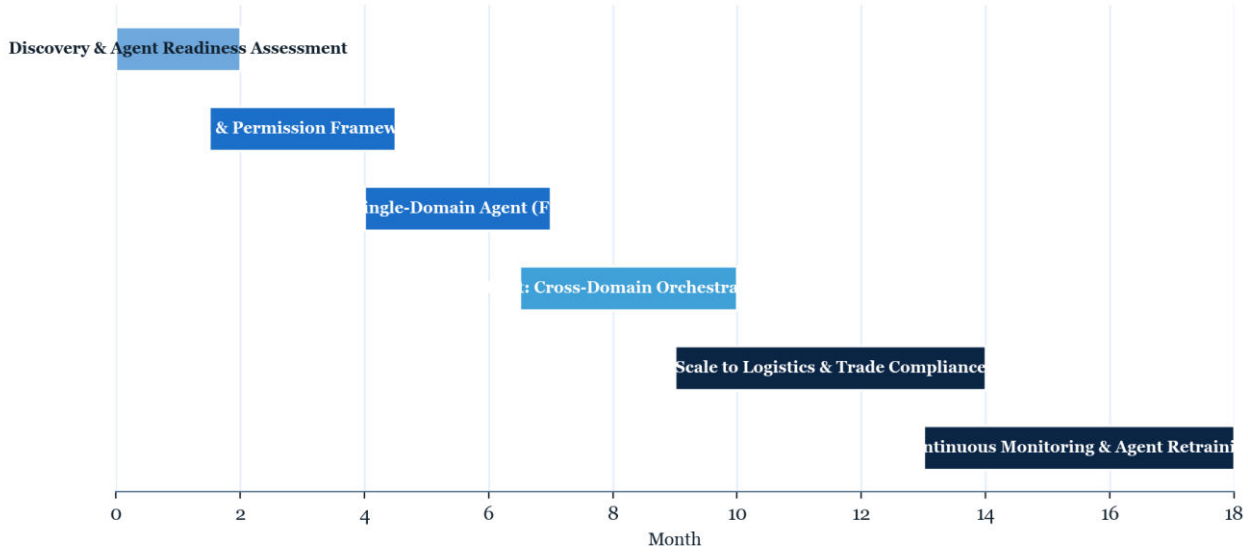


Figure 8. Illustrative 18-month rollout roadmap: guardrail development begins before and overlaps with the first single-domain pilot, ahead of cross-domain scaling.

8.1 Phase Descriptions

1. **Discovery & agent readiness assessment (Months 0–2).** Inventory candidate processes, assess which ERP capabilities are already API-addressable, and identify a first pilot domain with clear, bounded autonomy thresholds.
2. **Guardrail & permission framework build (Months 1.5–4.5).** Establish the agent permissioning model, action-level thresholds, and audit logging infrastructure described in Section 7 - begun early enough to be ready before the first pilot goes live.
3. **Pilot: single-domain agent (Months 4–7).** Deploy a single agent (commonly in finance, given its well-defined rule structure) in a narrowly scoped, heavily monitored pilot.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

4. **Pilot: cross-domain orchestration (Months 6.5–10).** Introduce a second domain agent and the orchestrator role, testing cross-domain hand-offs such as the finance/logistics interaction triggered by a shipment exception.
5. **Scale to logistics & trade compliance (Months 9–14).** Expand the agent portfolio to the remaining focal domains, applying lessons learned on threshold calibration and escalation design from the earlier pilots.
6. **Continuous monitoring & agent retraining (Months 13–18 and ongoing).** Establish an ongoing cadence for reviewing escalation rates, sampled-audit findings, and prompt/tool updates as underlying ERP processes and regulatory requirements evolve.

8.2 Illustrative Directional Gains by Domain

Figure 10 presents illustrative, directional estimates of cycle-time and exception-rate improvements by domain, compiled from patterns described anecdotally across vendor case studies and practitioner commentary. These figures are not drawn from a single controlled study and should be read as indicative of relative emphasis - trade compliance deployments, for instance, are typically justified more by risk and error reduction than by raw cycle-time compression, while logistics deployments show the reverse emphasis.

Figure 10. Illustrative Directional Gains from Agentic AI, by ERP Domain

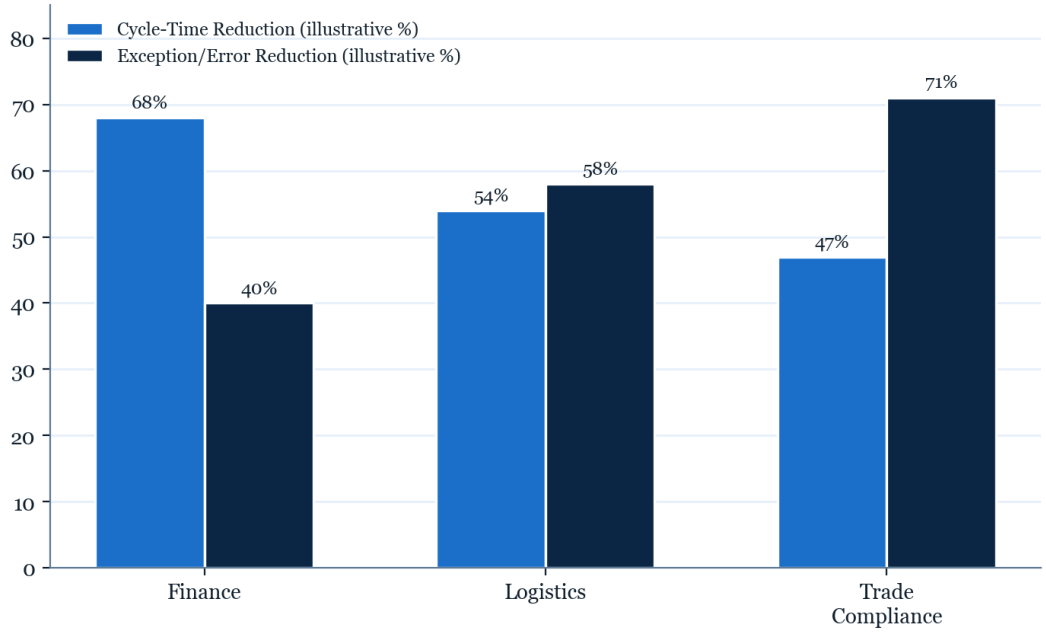


Figure.10. Illustrative directional gains from agentic AI deployment, by domain: cycle-time reduction versus exception/error reduction (both illustrative percentages).

IX. RISK PROFILE AND FAILURE MODES

Figure 9 presents an illustrative risk profile scoring six risk categories specific to agentic systems (as distinct from the broader AI governance risks already discussed in prior literature on predictive and generative AI in ERP contexts) along impact and likelihood.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

Figure 9. Illustrative Risk Profile for Agentic AI in ERP: Impact vs. Likelihood

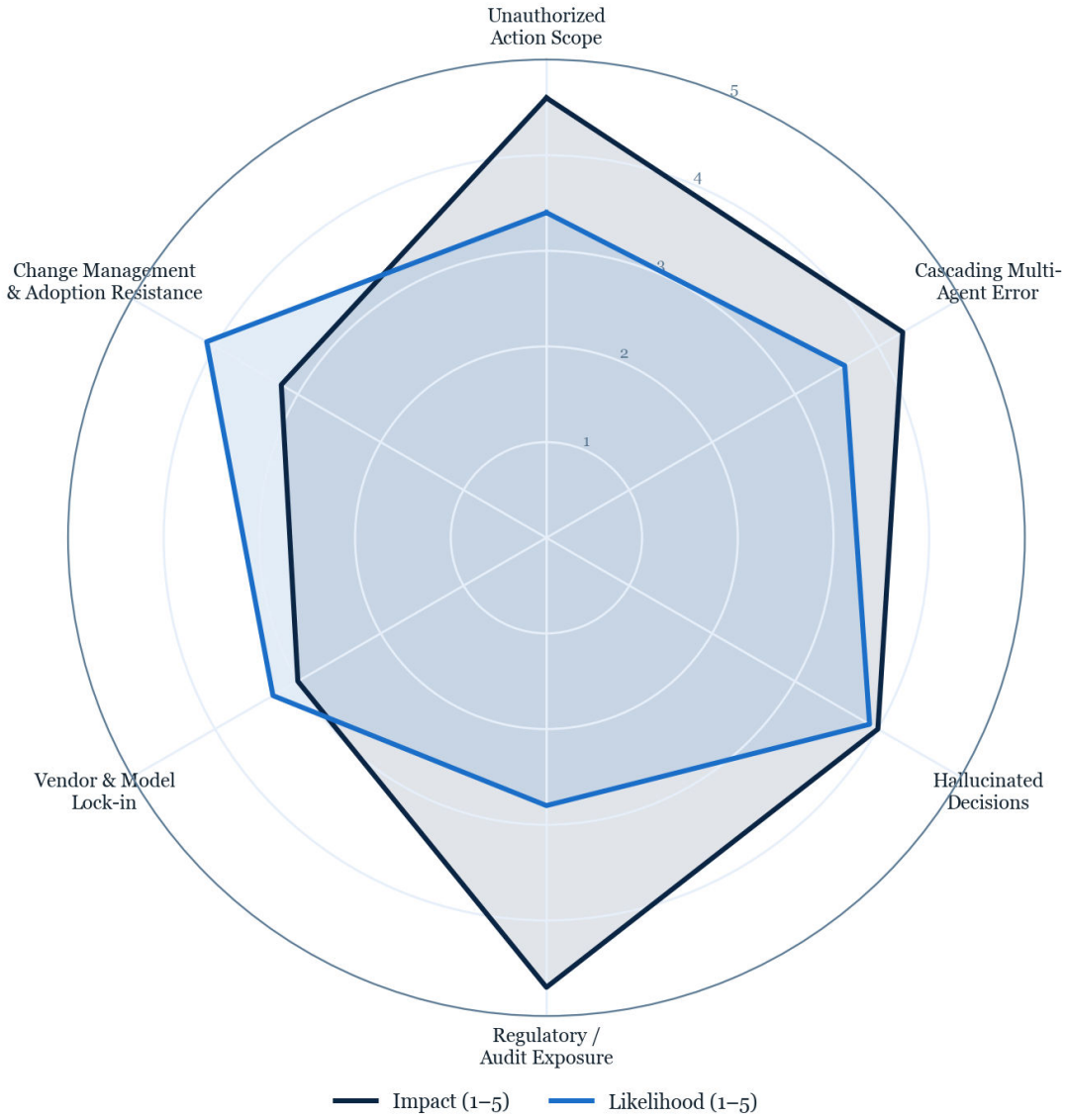


Figure 9 Illustrative risk profile for agentic AI in ERP, scoring six risk categories on Impact and Likelihood (1–5 scale each)

9.1 Unauthorized Action Scope

The highest-impact risk category specific to agentic systems is an agent taking an action outside its intended scope - for example, a finance agent instructed to "resolve this discrepancy" interpreting that instruction as license to issue a credit memo when only investigation was intended. This risk is mitigated primarily through the tool-scoping and threshold controls described in Section 7.2, not through prompt engineering alone.

9.2 Cascading Multi-Agent Error

In multi-agent systems, an error or hallucinated fact produced by one agent can be passed to a second agent as if it were verified ground truth, compounding rather than being caught. This is particularly relevant to the hub-and-spoke architecture of Section 3: the orchestrator's reconciliation step is the primary defense against this failure mode, and its design should explicitly include cross-checking agent outputs against the ERP system of record rather than trusting agent-to-agent claims at face value.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

9.3 Hallucinated Decisions

LLM-driven agents can produce plausible-sounding but factually incorrect justifications for an action, particularly under ambiguous or incomplete input data. Structured decision records (Section 7.4) that log the specific data an agent retrieved and the rule or threshold it applied - rather than only its natural-language justification - make this failure mode detectable during audit.

9.4 Change Management and Adoption Resistance

Scored as the highest-likelihood risk in Figure 9, resistance from staff whose judgment calls are being partially automated is a organizational, not technical, risk - but one that materially affects rollout success. Case material reviewed for this article suggests that framing agent deployment around escalation-handling capacity (freeing staff to focus on the genuinely ambiguous cases the agent defers to them) rather than headcount reduction correlates with smoother adoption, though this observation is anecdotal rather than rigorously measured.

X. OPEN QUESTIONS AND OUTLOOK FOR 2025

Several questions were actively unresolved in practitioner and research discussion as this article was being written in early January 2025:

- How should audit standards evolve to evaluate an autonomous decision chain rather than a single model's output - and will existing internal-control frameworks require formal extension, or will they be interpreted as already covering agentic systems?
- What is the right default orchestration topology (Section 4) as the number of domain agents in a single enterprise landscape grows past the four or five illustrated in this article - does centralized orchestration remain tractable at that scale, or does it become a bottleneck that pushes enterprises toward hybrid or hierarchical topologies?
- How will ERP vendors' own first-party agentic features (several of which were announced or previewed through late 2024) interact with independently built, cross-vendor multi-agent systems of the kind described in this article?
- What standardized format for agent decision records and audit trails, if any, will emerge as a de facto or formal standard, given the current diversity of ad hoc logging approaches across vendors and implementations?

Looking ahead, three trajectories appear likely to shape agentic AI in ERP contexts over the following one to two years: continued convergence of agent orchestration frameworks toward common tool-calling and permissioning standards; deeper first-party agent capability shipped directly by ERP vendors, narrowing the scope of what independent integration-layer agents need to cover; and growing regulatory specificity - building on the AI-specific frameworks that began taking effect in 2024 - around what constitutes adequate human oversight for autonomous financial and trade-related decisions.

XI. CONCLUSION

Agentic AI represents a genuine architectural departure from earlier generations of AI embedded in ERP landscapes, not merely an incremental capability upgrade. The shift from a system that recommends to a system that acts changes what governance must accomplish: it is no longer sufficient to validate that a model's output is accurate, because the object requiring oversight is now a chain of autonomous actions taken directly against systems of record. The hub-and-spoke architecture, orchestration topology comparison, and concentric governance model presented in this article are offered as a shared vocabulary for reasoning about that shift, grounded in design patterns already appearing in finance, logistics, and trade compliance deployments as of early 2025. Organizations that treat agent permissioning and audit infrastructure as prerequisites for piloting - rather than as concerns to be addressed after autonomy has already been granted - appear, based on the case material examined here, best positioned to capture the efficiency benefits of agentic automation without absorbing its most consequential risks.

REFERENCES

[1] Accenture. "Technology Vision 2024: AI-Native Enterprise Architecture." Accenture Research, 2024.
[2] AWS (Amazon Web Services). "Agents for Amazon Bedrock: Building Autonomous Applications." AWS Documentation, 2024.
[3] AWS (Amazon Web Services). "Best Practices for Integrating Machine Learning with Enterprise Applications." AWS Prescriptive Guidance, 2024.
[4] Anthropic. "Building Effective Agents." Anthropic Engineering Blog, 2024.
[5] Deloitte Insights. "Tech Trends 2024: AI and the Modern Enterprise Core." Deloitte Development LLC, 2023.
[6] Forrester Research, Inc. "Predictions 2025: Enterprise Software and Autonomous AI Agents." Forrester Research, 2024.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 14, Issue 1, January 2025||

|DOI:10.15662/IJAREEIE.2025.1401023|

[7] Gartner, Inc. "Predicts 2025: The Rise of Agentic AI in the Enterprise." Gartner Research, 2024.
[8] Gartner, Inc. "Magic Quadrant for Enterprise Integration Platform as a Service." Gartner Research, 2023.
[9] IBM Corporation. "What Are AI Agents? A Guide to Autonomous Systems." IBM Think Topics, 2024.
[10] IDC. "Worldwide AI and Automation in ERP Software Forecast, 2023–2027." IDC Market Perspective, 2023.
[11] McKinsey & Company. "The State of AI in Early 2024: Gen AI Adoption Spikes." McKinsey Global Survey, 2024.
[12] Microsoft Corporation. "Autonomous Agents in Copilot Studio: Architecture Overview." Microsoft Learn, 2024.
[13] Microsoft Corporation. "Copilot in Dynamics 365: Overview and Architecture." Microsoft Learn, 2024.
[14] MIT Sloan Management Review. "Winning with AI: 2024 Global Executive Study." MIT Sloan Management Review and Boston Consulting Group, 2024.
[15] MuleSoft, LLC (Salesforce). "Connectivity Benchmark Report." MuleSoft Research, 2024.
[16] National Institute of Standards and Technology (NIST). "AI Risk Management Framework (AI RMF 1.0)." U.S. Department of Commerce, 2023.
[17] OpenAI. "Function Calling and Tool Use in Large Language Models." OpenAI Platform Documentation, 2024.
[18] Oracle Corporation. "Oracle Fusion Cloud ERP: AI Agents and Autonomous Process Automation." Oracle Documentation, 2024.
[19] SAP SE. "SAP Joule Agents: Multi-Agent Orchestration Overview." SAP News Center, 2024.
[20] SAP SE. "SAP Business Technology Platform: Integration Suite Overview." SAP Help Portal, 2024.
[21] Salesforce, Inc. "Agentforce: Building Autonomous Agents for the Enterprise." Salesforce Engineering Blog, 2024.
[22] U.S. Department of Commerce, Bureau of Industry and Security. "Export Administration Regulations: Denied Persons List Guidance." Federal Register, 2023.
[23] Workday, Inc. "Workday AI: Architecture and Governance Overview." Workday Documentation, 2024.
[24] World Economic Forum. "Chief Data Officer Insight Series: Governance for AI in the Enterprise." WEF Whitepaper, 2023.
[25] European Commission. "Regulation (EU) 2024/1689 Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)." Official Journal of the European Union, 2024.
[26] LangChain, Inc. "Multi-Agent Orchestration Patterns." LangChain Documentation, 2024.
[27] Google Cloud. "Application Integration and AI: Reference Architectures." Google Cloud Architecture Center, 2024.
[28] Boston Consulting Group. "From Potential to Profit: Closing the AI Impact Gap." BCG Publications, 2024.

Appendix A: Glossary of Terms

Agentic AI - An AI system, typically built around a large language model, that plans a sequence of steps toward a goal, invokes tools or APIs to gather information or take action, and iterates based on intermediate results, as distinct from a single-inference predictive model.

Denied-Party Screening - The process of checking a transaction counterparty against government lists of individuals and entities restricted from certain trade activities.

ERP (Enterprise Resource Planning) - Integrated software managing core business processes such as finance, procurement, supply chain, and human resources within a single system of record.

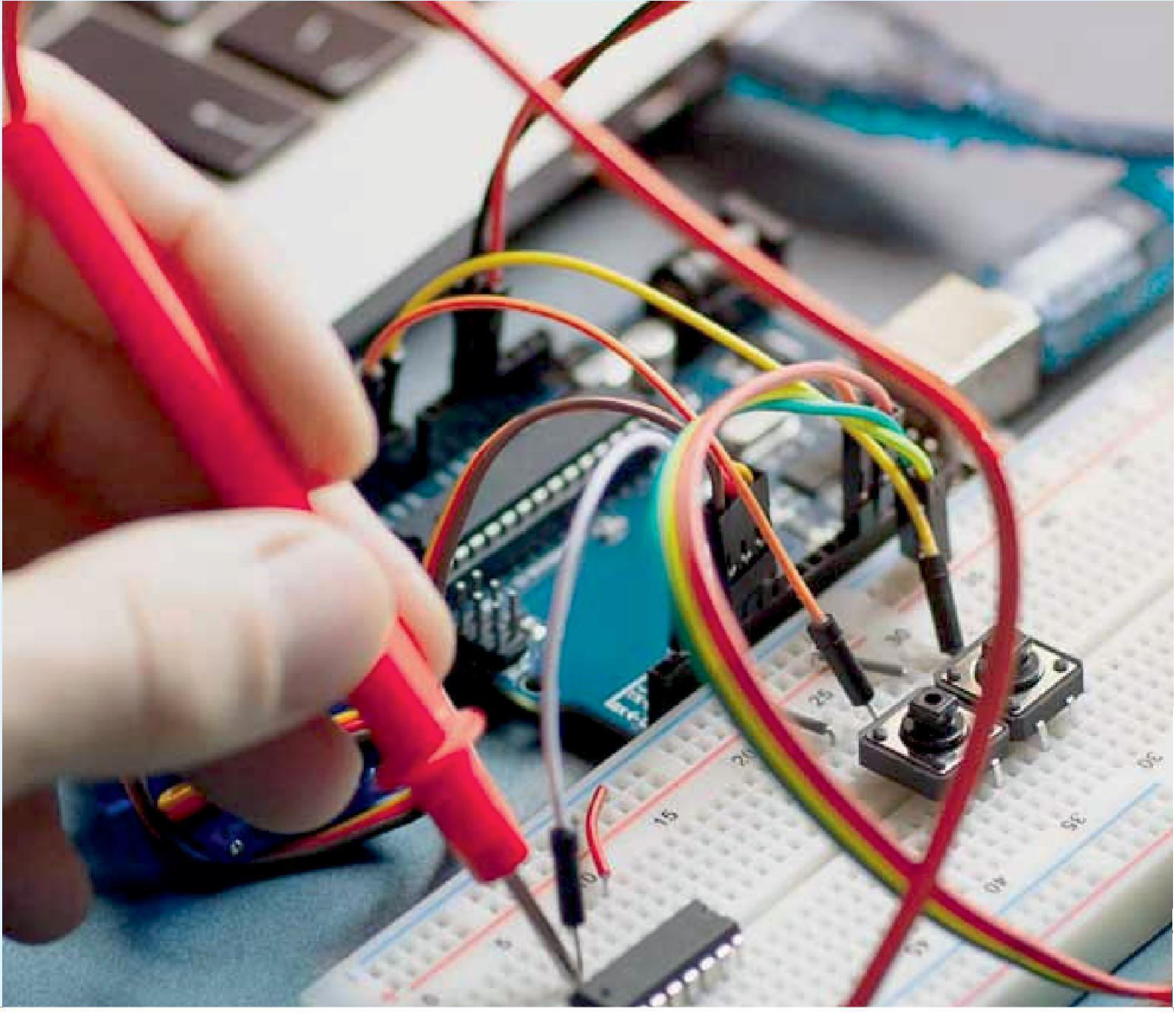
Harmonized System (HS) Classification - An internationally standardized system of names and numbers for classifying traded products, used to determine duties, tariffs, and licensing requirements.

Human-in-the-Loop - A design pattern in which a human reviewer must approve, or is notified of, an AI system's action before or immediately after it takes effect, particularly for high-risk decisions.

Multi-Agent System - An AI architecture composed of multiple distinct agents, each often specialized to a domain or task, coordinated by an orchestrator or by direct peer-to-peer communication.

Orchestrator Agent - In a hub-and-spoke multi-agent architecture, the agent responsible for decomposing an incoming goal into sub-tasks, delegating them to domain agents, and reconciling the results.

Tool-Calling - The mechanism by which an LLM-driven agent invokes a predefined function or API endpoint, described to the model via a schema, to read or write data outside its own context.



INNO SPACE
SJIF Scientific Journal Impact Factor

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

9940 572 462 6381 907 438 ijareeie@gmail.com

Scan to save the contact details