JAIAA **RESEARCH ARTICLE**
Journal of AI Analytics and Applications **Vol. 3, No. 2 (2025), pp. 28-49**
ISSN 3067-7386 | Open Access | Peer-Reviewed **DOI: 10.63646/jaiaa.2025.030202**

Domain-Specialized Agentic AI Analytics for Secure Multi-Context Enterprise Decision System

Nur Aisyah Rahman[1]; Mohd Farid Hassan[2]; Siti Mariam Abdullah[3] *
[1] School of Computing, Universiti Utara Malaysia, Sintok, Malaysia
[2] Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Malaysia
[3] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Malaysia
* Corresponding author: siti.mariam@uitm.edu.my

ARTICLE INFO
Received
January 08, 2025
Revised
March 18, 2025
Accepted
May 12, 2025
Available Online
June 30, 2025
DOI
10.63646/jaiaa.2025.030202

Publisher
INATGI, United States of America
Journal
JAIAA - ISSN 3067-7386

Abstract
Agentic artificial intelligence is increasingly used to connect enterprise data, automate workflows, and support decisions across business domains. However, enterprise decisions are rarely contained within a single data source or a single policy context. They span finance, supply chain, customer operations, compliance, cybersecurity, and legacy platforms with different rules and risk thresholds. This article develops a domain-specialized framework for secure multi-context enterprise decision systems. Building on the core ideas of mesh-based multi-context coordination, the paper argues that safe agentic analytics requires domain nodes that combine local knowledge, validated tools, data-governance policies, and auditable decision rights. The framework is presented as a layered architecture and a six-stage decision cycle: request parsing, policy evaluation, local action, remote delegation, result aggregation, and assurance. A scenario-based evaluation compares three architectures--centralized agent, hub-style multi-context access, and domain-specialized mesh analytics--across decision accuracy, privacy control, fault isolation, legacy compatibility, and auditability. The results suggest that domain specialization improves privacy, resilience, and accountability even when it increases initial design effort. The paper contributes a practical roadmap for incremental adoption and a research agenda for evaluating agentic AI decision systems in regulated enterprise environments.

Keywords: Agentic AI; domain-specialized analytics; multi-context systems; enterprise decision systems; data governance; secure AI; privacy-preserving analytics

I. INTRODUCTION

Autonomous enterprise decision support is moving from dashboards and workflow automation toward agentic artificial intelligence systems that can interpret requests, select tools, consult multiple data contexts, and propose or execute actions. This transition is not only a technical shift in language model capability; it is an architectural shift in how organisations distribute decision authority across business domains. Large language models provide a flexible reasoning interface, yet enterprise decisions are rarely made within one unified data source. They are distributed across finance systems, customer records, production platforms, supply chain portals, compliance archives, and security logs. A generic agent that can call tools across all these domains may appear efficient, but it also creates a significant risk: the agent may reason with incomplete context, apply the wrong domain rule, or execute an action outside its appropriate authority boundary. The uploaded source paper identifies this problem through the idea of mesh-based multi-context coordination, where each domain operates as a specialised node rather than as a passive data source. The present article develops that idea into a broader analytical framework for domain-specialized agentic AI analytics in secure multi-context enterprise decision systems.

JAIAA **RESEARCH ARTICLE**
Journal of AI Analytics and Applications **Vol. 3, No. 2 (2025), pp. 28-49**
ISSN 3067-7386 | Open Access | Peer-Reviewed **DOI: 10.63646/jaiaa.2025.030202**

The argument of this article is that enterprise agentic AI should not be organised around one all-knowing central agent. It should be organised around domain-specialized analytical agents that understand the business rules, data constraints, operational vocabulary, and approval requirements of their own domains. A central agent may still coordinate, but it should not absorb every decision right. Instead, a secure multi-context decision system should delegate reasoning to domain nodes, evaluate whether data can be shared, request summaries when raw data is restricted, and aggregate evidence through auditable decision logic. This architecture is consistent with the recent development of large language models, retrieval-augmented generation, tool use, and multi-agent simulation, but it also responds to the documented risks of opaque reasoning, uncontrolled action, and weak accountability in foundation-model systems (Brown et al., 2020; Vaswani et al., 2017; Devlin et al., 2019; Ouyang et al., 2022; Wei et al., 2022; Yao et al., 2023; Schick et al., 2023; Lewis et al., 2020; Park et al., 2023; Liang et al., 2022; Bommasani et al., 2021; Bender et al., 2021).

The need for domain specialization arises because enterprise decision systems are socio-technical systems rather than neutral software services. In a credit-risk workflow, a recommendation must respect financial regulation and explainability expectations. In a hospital procurement workflow, it must protect patient-sensitive data even if the user request is operational rather than clinical. In a supply-chain disruption workflow, it must combine demand forecasts, supplier capacity, contract obligations, and geopolitical constraints. Treating these contexts as equivalent tool endpoints invites errors of scope and interpretation. A domain-specialized architecture instead embeds governance at the location where context is richest: the domain itself. The architectural insight is therefore not merely that agents need access to more contexts, but that context access must be mediated by local policy, local expertise, and local auditability.

This paper makes four contributions. First, it defines the concept of domain-specialized agentic AI analytics as an architecture that combines domain-specific reasoning, secure tool invocation, policy-aware data sharing, and cross-domain decision aggregation. Second, it develops a lifecycle framework for building secure multi-context decision systems across enterprise domains. Third, it presents a scenario-based analytical evaluation comparing centralized agentic AI, hub-style multi-context access, and a domain-specialized mesh architecture across decision accuracy, privacy control, fault isolation, legacy fit, and auditability. Fourth, it identifies implementation principles for organisations seeking incremental adoption without replacing existing systems or compromising data sovereignty. Figure 1 summarises the proposed architecture and introduces the five layers used throughout the article.

The difference between a secure enterprise decision system and an ordinary agentic application is the treatment of consequences. A consumer assistant may provide an imperfect suggestion without changing organisational state. An enterprise agent may alter a procurement route, approve an exception, classify a customer as high risk, or recommend a cyber-containment action. These decisions create legal, financial, and operational consequences. For that reason, the architecture must preserve the ability to determine why a recommendation was produced, which data context supported it, whether policy boundaries were respected, and which actor accepted responsibility. The contribution of a domain-specialized approach is that it converts enterprise context into an operational design principle. It recognises that decision quality depends not only on language-model competence but also on the alignment among data ownership, workflow authority, and local domain semantics.

A second motivation is the economic reality of enterprise transformation. Many organisations cannot replace legacy platforms simply to deploy agentic AI. Financial ledgers, enterprise-resource-planning systems, manufacturing execution systems, customer-relationship platforms, and compliance repositories are often deeply embedded in daily operations. A mesh-oriented decision system can wrap these systems through controlled functions and progressively expose capabilities. This approach lets the organisation gain analytical value while avoiding the risks of a central data migration. It also allows different domains to advance at different speeds: a compliance domain may begin with read-only policy retrieval, whereas a customer operations domain may pilot semi-automated routing. The architecture therefore supports differentiated maturity, which is more realistic than assuming uniform enterprise readiness.

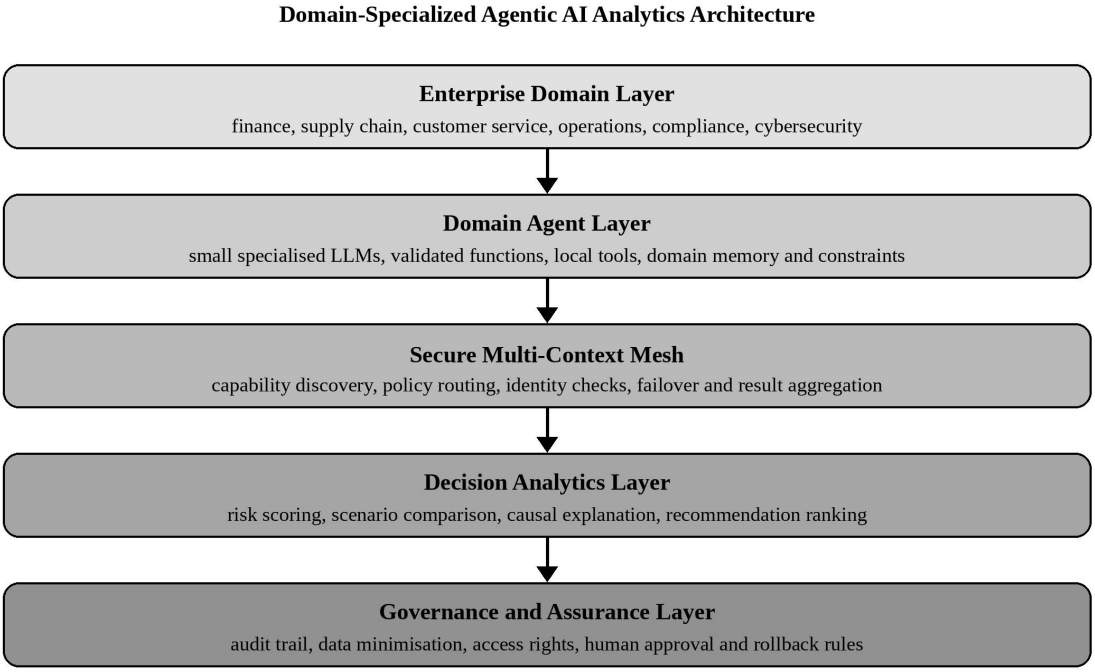

Figure 1. Domain-specialized architecture for secure multi-context enterprise decision analytics.

II. CONCEPTUAL BACKGROUND: FROM GENERAL AGENTS TO DOMAIN-SPECIALIZED ANALYTICS

The concept of an artificial agent has a longer history than contemporary language models. Classical agent research framed agents as software entities capable of perceiving, reasoning, communicating, and acting within environments. The shift brought by current language models is that natural language becomes the interface for task specification, planning, tool selection, and justification. A language-based agent can break down a request, ask a database for evidence, call a planning function, and generate an explanation. Yet this generality also weakens reliability when the agent faces a domain whose rules are not fully expressed in the prompt. The foundational literature on agents, multi-agent systems, automation, and human-AI interaction suggests that autonomy must be bounded by role assignment, coordination rules, and human oversight rather

than left to open-ended inference (Weidinger et al., 2021; Shinn et al., 2023; Mialon et al., 2023; Amershi et al., 2019; Mitchell et al., 2019; Gebru et al., 2021; Raji et al., 2020; Ribeiro et al., 2016; Lundberg and Lee, 2017; Arrieta et al., 2020; Guidotti et al., 2018; Doshi-Velez and Kim, 2017; Parasuraman et al., 2000).

Domain specialization therefore means more than fine-tuning a model on local documents. It is a system property. A specialised domain node contains four elements: a local knowledge base, a set of validated tools, a policy layer specifying what data and actions are allowed, and an explanation interface that translates recommendations into domain-relevant evidence. The model may be small or large, local or hosted, but it must operate within a clearly defined decision scope. This distinction is important because many enterprise failures arise not from poor language fluency but from wrong scope. An agent may generate a plausible answer while ignoring a compliance threshold, a data retention rule, or a required approval chain. Domain specialization turns implicit organisational knowledge into explicit control surfaces.

The proposed system also differs from conventional data integration. Data integration often tries to create a unified repository or a canonical data model. A secure agentic decision system may not need full data consolidation. In regulated or commercially sensitive environments, full consolidation may be impossible or undesirable. The system should instead support controlled context exchange: a finance node may share a risk score rather than account-level details; a human-resources node may share eligibility categories rather than employee records; a security node may share incident severity rather than raw logs. Differential privacy, secure aggregation, federated learning, and privacy-preserving computation provide technical foundations for this model, while data governance literature supplies the organisational principles needed to define ownership, quality, accountability, and permissible use (Wooldridge and Jennings, 1995; Shoham, 1993; Stone and Veloso, 2000; Gentry, 2009; Yao, 1982; Goldreich et al., 1987; Dwork, 2006; Abadi et al., 2016; Bonawitz et al., 2017; Shokri and Shmatikov, 2015; McMahan et al., 2017; Kairouz et al., 2021; Li et al., 2020; Yang et al., 2019; Nguyen et al., 2021).

Table I positions the proposed domain-specialized agentic analytics model against two more common architectures. A centralized agent has the advantage of simplicity but creates broad failure exposure. A hub-style multi-context architecture improves connectivity but does not necessarily provide local governance or failure containment. A domain-specialized mesh analytics architecture distributes reasoning and policy enforcement across domains, allowing the enterprise to coordinate without collapsing decision authority into a single point.

Domain specialization also responds to the problem of semantic compression. When all enterprise data is translated into a single generic prompt, local meanings are compressed into language that may lose operational nuance. A term such as priority, exception, exposure, or closure has different meanings in finance, cybersecurity, customer service, and legal compliance. If a central agent treats the same word as equivalent across contexts, it may make a superficially coherent but organisationally invalid recommendation. Domain-specialized analytics preserves semantic locality by allowing the domain to define its own vocabulary, evidence types, thresholds, and permissible actions. The result is not a collection of disconnected agents, but a federation of accountable analytical services that communicate through standardised decision contracts.

The conceptual foundation also requires distinguishing between information access and decision authority. Modern retrieval systems can access large amounts of information, but access alone does not establish

authority. A sales agent can retrieve a contract clause, but it may not have authority to interpret regulatory exposure. An operations agent can identify inventory scarcity, but it may not have authority to override financial controls. In the proposed framework, agentic AI becomes trustworthy only when access, inference, and action are separated into explicit layers. This separation makes it possible to permit broad analytical search while still limiting consequential action to authorised contexts. It also prevents the common error of treating answer generation as equivalent to accountable decision making.

A further conceptual distinction concerns analytics depth. Some agentic applications merely retrieve information and rephrase it. Others transform information into decision-relevant evidence. The proposed system emphasises the latter. Domain-specialized analytics should estimate uncertainty, identify assumptions, compare alternative actions, and show which rule or evidence supports the recommendation. For example, an inventory answer that says a product is low in stock is less valuable than an agentic analysis that states the expected shortage window, supplier dependency, financial exposure, and policy-approved mitigation options. The analytics layer therefore moves beyond conversational convenience toward structured decision support. This is why decision contracts, evidence classes, and assurance logs are central to the framework.

Table I. Comparison of enterprise agent architectures.

Architecture	Main Strength	Principal Risk	Governance Location	Best Use Case
Centralised general agent	Fast proof-of-concept and simple coordination	Wide failure exposure and weak local context	Platform or central team	Low-risk internal assistance
Hub-style multi-context access	Connects many repositories and tools through one interface	Connectivity without domain-level accountability	Hub policy plus user permissions	Search, retrieval, and summarisation
Domain-specialised mesh analytics	Local decision rules, constrained tools, fault isolation, and auditable routing	Higher initial design and contract effort	Domain node plus federated standards	Regulated, multi-domain decisions

III. FRAMEWORK DESIGN: SECURE MULTI-CONTEXT ENTERPRISE DECISION SYSTEM

The proposed framework is organised around a layered decision cycle rather than a single prompt-response interface. At the lowest layer, enterprise domains retain their own data assets, business rules, and operational tools. At the next layer, domain agents expose a limited set of callable capabilities. These capabilities may include retrieval from a local knowledge base, risk scoring, simulation, classification, exception detection, or workflow initiation. A secure multi-context mesh then manages routing among domains. It does not automatically expose raw data; it decides whether the requested task can be completed locally, whether another domain must be consulted, and whether the answer should be returned as raw output, structured evidence, or a privacy-preserving summary.

A key design principle is that every domain node should behave as both an analytical service and a governance boundary. The finance node, for example, should not merely answer financial questions. It should decide whether a requesting agent has authority to ask that question, whether the requested evidence can be shared, whether a human approver is required, and whether the result should be logged for future audit. Similarly, a cybersecurity node should not simply provide threat scores. It should protect indicators that could

expose vulnerabilities, limit operational actions to authorised users, and return risk categories when raw incident details are not necessary. This is why the framework is described as analytics for a decision system rather than as a general chatbot architecture.

Figure 2 presents the decision cycle. A user or system event generates a request. The request is parsed and sent through a policy gate. If a domain can answer locally, it executes a validated function. If cross-domain evidence is required, the domain agent delegates a bounded task to another agent. The receiving node evaluates its own policy before returning either direct output, transformed output, or a summary. Results are aggregated into a recommendation that includes confidence, assumptions, alternatives, and approval requirements. The decision is then logged, monitored, and subject to fallback or rollback procedures when later evidence contradicts the recommendation.

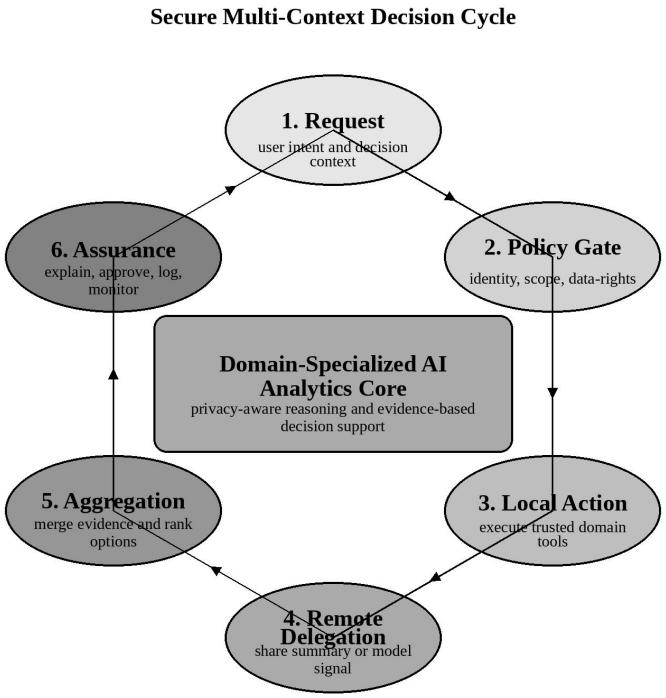

Figure 2. Six-stage secure multi-context decision cycle for domain-specialized agentic AI analytics.

The design borrows from tool-using language models, explainable AI, model reporting, and human-centred AI guidelines, yet it adds an enterprise architectural principle: every action must be traceable to a domain authority and a permissible context. This matters because enterprise decisions often combine prediction, judgment, and accountability. A model card may describe a model, and a datasheet may describe a dataset, but a domain-specialized agentic decision system must also document which domain accepted responsibility for an output, which policy authorised the data exchange, and which human role can override the final recommendation (Papernot et al., 2018; Shokri et al., 2017; Biggio et al., 2013; Goodfellow et al., 2015; Madry et al., 2018; Bagdasaryan et al., 2020; Yin et al., 2018; Blanchard et al., 2017; Bharadwaj et al., 2013; Vial, 2019; Verhoef et al., 2021; Trkman, 2010; van der Aalst, 2016; Jamshidi et al., 2018; Soldani et al.,

2018).

The architecture can be implemented using either local models, hosted models, or a hybrid arrangement. A domain that handles highly sensitive data may choose a small local language model combined with deterministic business rules. Another domain may use a retrieval-augmented external model for general summarisation while keeping action functions local. A third domain may use a specialised classifier rather than a generative model at all. The mesh layer should not enforce a single modelling strategy. Its role is to normalise capability descriptions, route requests safely, and record decision traces. This model heterogeneity is important for enterprise adoption because different domains face different combinations of latency, privacy, cost, and accuracy requirements.

A useful design practice is to define every domain capability as a contract. The contract specifies input type, data sensitivity, model type, output type, evidence requirement, approval threshold, and rollback option. For example, a fraud-risk capability may accept a transaction identifier and return a probability band, contributing features, and a recommended review priority. It should not return raw customer records unless the user has sufficient authority. A supplier-risk capability may return supplier category, disruption likelihood, and alternative sourcing recommendations, but the final supplier substitution should require a human procurement role. These contracts make agentic orchestration safer because the coordinator interacts with documented capabilities rather than with unrestricted tools.

The framework can also be interpreted as a response to the problem of context collapse. In social and organisational systems, context collapse occurs when different audiences, meanings, or norms are forced into one interpretive space. A central enterprise agent can create a similar technical problem: finance, security, customer service, and legal compliance become flattened into one prompt environment. A secure multi-context architecture avoids this by preserving separate contexts and allowing them to communicate through defined interfaces. This is not simply a privacy measure. It is also a way of preserving the epistemic conditions under which high-quality decisions are made. Domain nodes know what counts as reliable evidence in their own settings, while the mesh layer supports interoperability without erasing that knowledge.

IV. DOMAIN ANALYTICS PRIMITIVES AND DECISION FUNCTIONS

A secure multi-context enterprise decision system should not start by building a universal agent. It should start by identifying decision functions. In the proposed framework, a decision function is a reusable analytical operation that transforms domain context into evidence for a decision. Examples include supplier risk classification, customer churn explanation, credit exposure scoring, fraud anomaly detection, inventory disruption forecasting, compliance exception identification, and incident severity ranking. The agentic component becomes valuable when it can select and sequence these decision functions across domains without violating the policy constraints that govern them.

Table II identifies representative enterprise domains and the analytical primitives that can be exposed by domain-specialized agents. The purpose of the table is not to prescribe a complete enterprise architecture. Rather, it shows that secure agentic analytics requires a structured vocabulary of permissible functions. A domain node should advertise what it can do, what data it requires, what type of output it returns, and what approval level is needed. This capability description makes orchestration safer because the coordinator is not

guessing what a domain can provide. It is selecting from documented capability contracts.

Table II. Enterprise domains and analytical primitives exposed by specialised agents.

Domain	Decision Function	Local Evidence	Permissible Shared Output	Approval Boundary
Finance	Credit exposure and margin risk scoring	Ledgers, payment history, contract terms	Risk class, exposure band, explanation summary	Finance manager or credit committee
Supply chain	Supplier disruption and substitute recommendation	Inventory status, lead time, supplier events	Supply risk level and substitution options	Procurement owner
Customer operations	Churn, complaint and service-priority analytics	CRM cases, support transcripts, service logs	Segment-level priority and recommended action	Service operations lead
Cybersecurity	Incident triage and control recommendation	Alerts, logs, vulnerabilities, identity events	Incident severity and containment recommendation	Security incident commander
Compliance	Policy exception and audit-readiness screening	Policies, regulatory mapping, prior cases	Exception category and evidence checklist	Compliance officer

The table also highlights the difference between domain data and domain decision rights. A supply-chain node may contain supplier lead times and inventory records, but the right to approve a supplier substitution may belong to procurement leadership. A finance node may calculate margin exposure, but the right to approve pricing changes may belong to a commercial authority. Agentic AI analytics must therefore separate evidence generation from action authorization. This separation aligns with research on digital transformation and process management: automation becomes valuable when it redesigns decision flows, but it becomes risky when it erases accountability boundaries (Balalaie et al., 2016; Dragoni et al., 2017; Khatri and Brown, 2010; Abraham et al., 2019; Klievink et al., 2017; Floridi and Cowls, 2019; Jobin et al., 2019; Vinuesa et al., 2020; Brynjolfsson et al., 2023; Eloundou et al., 2023).

The most important advantage of the domain-function approach is that it enables incremental adoption. Organisations can begin with low-risk domains, such as internal knowledge retrieval, help-desk triage, or report summarisation. They can then expand toward higher-impact domains, such as customer-risk scoring, operational planning, or financial exception management. The source paper's emphasis on evolutionary implementation is especially relevant here: enterprise legacy systems rarely disappear quickly, and a decision system must wrap existing systems through reliable interfaces before it can coordinate complex actions. Microservice and process-mining research reinforces this point by showing that modularity, event traces, and service boundaries are key conditions for scalable digital operations (Raisch and Krakowski, 2021; Jarrahi, 2018; Kellogg et al., 2020; Lu, 2019; Zhang and Lu, 2021; Lu, 2025; Lu, 2017a).

In many enterprises, the most difficult step is not creating domain agents but deciding what counts as a domain. A domain may be organised around business ownership, data ownership, regulatory responsibility, or operational process. These definitions do not always align. For instance, customer data may be owned by the marketing function, processed by customer service, audited by compliance, and affected by finance rules. The framework recommends starting with decision domains rather than organisational charts. A decision domain is defined by who owns the decision, what evidence is required, what risk is created, and what action can be taken. This definition is more useful for agentic analytics because it focuses on accountable decision rights.

Another important primitive is the explanation boundary. Not every explanation should expose every data point. A cybersecurity explanation may need to state that the recommendation is based on unusual authentication events, known vulnerability exposure, and anomalous outbound traffic, but it may not need to reveal precise detection rules. A human-resources explanation may need to cite policy categories without exposing personal details. A finance explanation may need to display risk drivers without disclosing confidential contract clauses. Domain-specialized agents can generate explanations at the appropriate granularity because they know the disclosure rules of their domain. This is a major advantage over generic explanation interfaces that assume transparency always means more information.

The proposed domain primitives may be implemented as deterministic functions, machine-learning models, retrieval workflows, or human-in-the-loop procedures. The important requirement is that each primitive has a defined input and output boundary. In many cases, deterministic functions should be preferred to generative reasoning. A tax calculation, eligibility check, or role-permission test should not be generated probabilistically. Generative models are better suited for synthesis, explanation, triage, and scenario comparison. By combining deterministic controls with generative reasoning, the system can benefit from flexibility while limiting unpredictable behaviour. This hybrid design is especially important in enterprise decision systems because reliability matters as much as creativity.

V. SECURITY, PRIVACY, AND GOVERNANCE MODEL

Security in domain-specialized agentic AI analytics is not limited to preventing unauthorised login. It includes preventing unauthorised reasoning. A user may be authorised to ask a sales question but not authorised to trigger a pricing action. A model may be authorised to read a policy summary but not raw personnel data. A cross-domain workflow may be authorised to combine supplier risk categories and inventory forecasts but not contract-level price clauses. These distinctions require a layered governance model that includes identity, scope, data minimisation, model evaluation, action controls, and audit trails.

The proposed governance model uses five controls. First, identity and role controls define who can ask for what type of decision support. Second, data-policy controls determine whether a request may access raw data, aggregated data, synthetic data, or no data. Third, model-policy controls specify which model may be used for a task, including whether a small local model is sufficient or a stronger external model is allowed. Fourth, action controls determine whether an output can remain advisory or initiate a workflow. Fifth, audit controls record the request, domain routing, data-sharing decision, model output, explanation, and approval status. These controls should be embedded within domain nodes rather than only applied after final output.

The threat model includes prompt injection, malicious tool invocation, model extraction, membership inference, data reconstruction, cross-domain inference, and silent action drift. A domain-specialized mesh architecture cannot eliminate these threats, but it can reduce their blast radius. If one domain node is compromised, the attacker should not automatically gain access to all contexts. If a model behaves unpredictably, its action scope should be constrained by domain policy. If a cross-domain workflow fails, the system should degrade gracefully rather than executing irreversible operations. The security and privacy literature suggests that such controls must be combined rather than treated as alternatives: differential privacy, secure aggregation, adversarial robustness, interpretability, and auditability each address different failure

modes (Lu, 2017b; Xu et al., 2021; Chen et al., 2024; Lu, 2022; Zheng and Lu, 2022; Lu and Ning, 2020; Lu and Zheng, 2020; Lu et al., 2024).

Table III presents a scenario-based risk matrix. The numbers are not claims from live enterprise deployments; they are analytical scores used to compare architectures under common enterprise concerns. The scores are normalised on a 0-100 scale where higher indicates stronger performance on the specified criterion. The comparison suggests that centralized agents perform adequately on decision speed but poorly on privacy control and fault isolation. Hub-style multi-context systems improve data access but remain limited when domain policies are weak. The domain-specialized mesh approach scores highest on privacy, fault containment, and auditability, even though it may involve higher initial design effort.

The governance model should also define negative capabilities: what the agent is not allowed to do. Negative capability lists are often more important than positive tool lists. A finance agent may not approve write-offs. A compliance agent may not interpret a regulation beyond a stated jurisdiction. A procurement agent may not contact external suppliers without approval. A security agent may not disable a system outside an emergency protocol. These negative capabilities should be machine-readable and tested as part of system evaluation. Without them, the agent may infer from broad language instructions that it can perform actions that the organisation never intended to delegate.

Privacy protection has to be considered across inference chains rather than at single access points. A raw dataset may never leave a domain, yet repeated summaries from that domain can still reveal sensitive information. Cross-domain agents may infer protected attributes through correlation. Aggregated outputs may disclose competitive strategy when combined with external knowledge. Therefore, secure multi-context analytics should include query-rate monitoring, output sensitivity scoring, summary redaction, and cumulative disclosure audits. These controls are especially important when the same user or coordinating agent repeatedly requests related evidence from multiple domains.

The policy layer should be tested through adversarial scenarios. Testers should attempt to trick the agent into bypassing domain scope, leaking restricted information, escalating action rights, or combining harmless summaries into sensitive inference. These tests should become part of routine model governance rather than one-time red-team exercises. In a domain-specialized architecture, tests can be localised: the finance node can be tested for financial-policy leakage; the cybersecurity node can be tested for attack-surface disclosure; the human-resources node can be tested for protected-attribute inference. Localised testing improves specificity and reduces the burden of evaluating an entire enterprise agent as one opaque system.

Table III. Scenario-based risk matrix for secure multi-context agentic decision systems.

Criterion	Central Agent	Hub Multi-Context	Domain Mesh	Interpretive Meaning
Decision accuracy	70	78	84	Domain evidence improves contextual correctness
Privacy control	41	55	82	Local policy gates reduce unnecessary raw-data exposure
Fault isolation	35	50	86	Node boundaries reduce cascading failure risk
Legacy fit	48	61	79	Adapter-based domain functions support incremental integration
Auditability	45	58	88	Domain logs and capability contracts reconstruct

JAIAA **RESEARCH ARTICLE**
Journal of AI Analytics and Applications **Vol. 3, No. 2 (2025), pp. 28-49**
ISSN 3067-7386 | Open Access | Peer-Reviewed **DOI: 10.63646/jaiaa.2025.030202**

decision paths

VI. SCENARIO-BASED ANALYTICAL EVALUATION

To add an analytical layer to the conceptual framework, this study constructs a scenario-based evaluation across five enterprise decision criteria: decision accuracy, privacy control, fault isolation, legacy fit, and auditability. The evaluation compares three architectures. The first is a centralized agent that accesses enterprise tools through a single orchestration layer. The second is a hub-based multi-context model that provides connections to many systems but keeps governance primarily at the platform level. The third is the proposed domain-specialized mesh analytics model. Each architecture is scored against five enterprise scenarios: supplier disruption response, customer-credit exception handling, cyber-incident triage, regulatory reporting, and cross-domain operational planning.

The scoring logic is intentionally transparent. Decision accuracy refers to whether the system can combine relevant evidence without losing domain meaning. Privacy control refers to whether raw domain data can remain local when cross-domain reasoning is needed. Fault isolation refers to whether failure in one node can be contained. Legacy fit refers to whether existing enterprise systems can be wrapped without wholesale replacement. Auditability refers to whether the decision path can be reconstructed after output. Figure 3 displays the comparative results. The domain-specialized mesh approach does not win because it uses a more powerful model; it wins because the architecture aligns data authority, tool access, and decision accountability with domain boundaries.

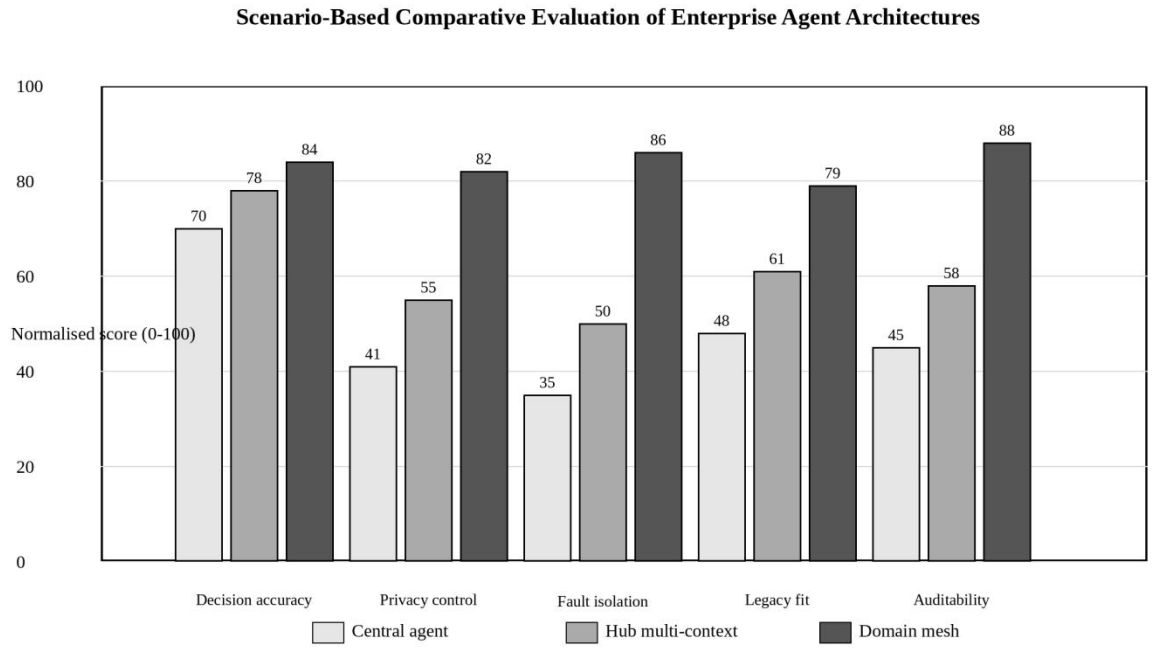

Figure 3. Normalised scenario scores comparing central, hub-style, and domain-specialized mesh architectures.

The results also clarify an implementation trade-off. A centralized architecture may be attractive for quick pilots because it requires fewer policy contracts. However, its risk grows with the number of connected tools. A hub-based multi-context architecture improves connectivity and may be suitable for information retrieval, but it is insufficient for high-stakes decision workflows unless governance is domain-aware. The domain-specialized mesh approach demands more design effort at the beginning because it requires capability descriptions, policy definitions, and audit templates for each node. Once these contracts exist, however, the architecture becomes more scalable because additional domains can be added without redesigning the whole system.

The analytical evaluation further suggests that smaller, domain-specific models can be strategically valuable. General foundation models are strong at broad language tasks, but enterprise decision systems often need constrained reasoning over local vocabulary, regulated processes, and stable action boundaries. A small model hosted inside a domain node may be safer for policy interpretation and tool selection than a larger external model that lacks access to local procedure. This conclusion is consistent with work on retrieval, tool use, human oversight, and Industry 4.0 information systems, where value depends on grounding general intelligence in operational context rather than on model scale alone (Brown et al., 2020; Schick et al., 2023; Lewis et al., 2020; Lu, 2019; Lu, 2025; Lu, 2017a).

The scenario scores are best interpreted as architectural tendencies rather than universal measurements. A central agent can score higher if it is carefully constrained and applied to a narrow process. A domain mesh can score lower if its capability contracts are poorly written or if domains refuse to cooperate. Nevertheless, the comparison captures a structural difference: centralised systems reduce coordination cost at the expense of risk concentration, while domain-specialised systems increase coordination effort in exchange for resilience and accountable control. This trade-off is similar to the broader enterprise choice between speed and governability. For low-stakes tasks, centralisation may be rational. For high-stakes multi-context decisions, governance-by-design becomes more important than initial simplicity.

The data-analysis component can be expanded in future work by using real workflow logs. A live evaluation would measure how often agents request remote domain support, how often policy gates deny requests, how much evidence is transformed into summaries, how frequently humans override recommendations, and how long decision cycles take. These metrics would create a measurable profile of enterprise agentic maturity. For example, a high denial rate may indicate unclear user roles or overly broad requests. A high override rate may indicate weak explanations or poor domain alignment. A high latency rate may indicate excessive routing. These operational indicators are more informative than task accuracy alone because they reveal whether the system is usable, governable, and trusted.

A useful extension of the evaluation method is cost-sensitive scoring. A wrong recommendation in a low-value procurement task is different from a wrong recommendation in a cyber-incident response workflow. Future evaluations should assign consequence weights to decision errors, privacy exposures, and action delays. A domain mesh may appear slower than a central agent in simple tasks, but its expected loss may be lower in high-consequence scenarios because it prevents unauthorised data sharing and contains failure propagation. This suggests that enterprise AI evaluation should incorporate expected-risk reduction, not only accuracy or speed. Such an approach would better reflect the real value of secure multi-context decision

systems.

The evaluation also implies that no single metric should dominate enterprise assessment. Accuracy can be misleading when a system reaches a correct answer through an unauthorised path. Latency can be misleading when a fast output bypasses an approval boundary. Privacy can be misleading when raw data is protected but repeated summaries reveal sensitive patterns. A mature evaluation should combine technical metrics with governance metrics, including denied request rate, policy exception frequency, human override rate, domain escalation count, and post-decision correction rate. These indicators transform the system from a black-box assistant into a measurable organisational control mechanism.

VII. IMPLEMENTATION ROADMAP FOR ENTERPRISE ADOPTION

A practical roadmap should treat domain-specialized agentic AI analytics as a staged transformation rather than as a one-time platform purchase. Stage 1 is a single-domain pilot. The organisation selects a bounded process, such as service-desk triage or compliance knowledge retrieval, and defines the domain's authorised tools, data rules, and audit requirements. Stage 2 is a two-domain workflow. For example, a procurement agent may request finance approval for a supplier-risk exception, but finance returns a decision category rather than raw financial records. Stage 3 is an enterprise mesh. Multiple domains expose capability contracts and share standard policy metadata. Stage 4 is a federated ecosystem where external partners, regulators, or subsidiaries participate through secure summaries or model-level signals.

Figure 4 illustrates this roadmap. The stages deliberately avoid a big-bang migration. Many organisations possess fragmented but mission-critical legacy systems. A realistic architecture must wrap these systems through adapters and function interfaces, while retaining local ownership of data products. The roadmap also requires human roles. Domain stewards define capability descriptions. Risk officers define policy thresholds. System architects design routing and failover. Data owners approve data-sharing categories. Operators evaluate whether recommendations are understandable and useful. The system succeeds when these roles are treated as part of the architecture rather than as afterthoughts.

JAIAA **RESEARCH ARTICLE**
Journal of AI Analytics and Applications **Vol. 3, No. 2 (2025), pp. 28-49**
ISSN 3067-7386 | Open Access | Peer-Reviewed **DOI: 10.63646/jaiaa.2025.030202**

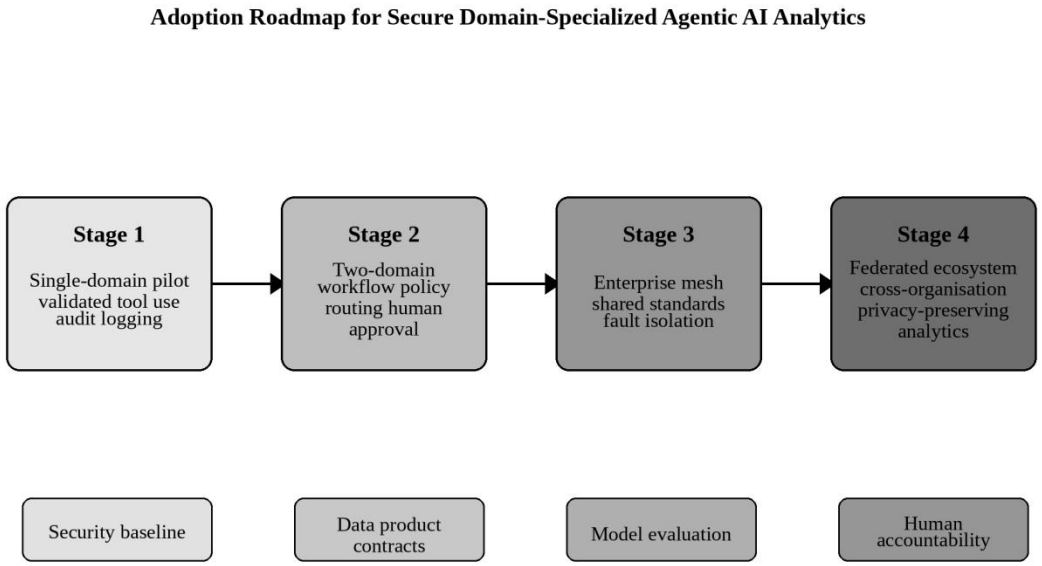

Figure 4. Four-stage adoption roadmap for secure domain-specialized agentic AI analytics.

Table IV converts the roadmap into governance actions. It lists the adoption stage, technical priority, decision-control requirement, evidence needed for assurance, and failure mode to monitor. The table emphasises that implementation maturity is not identical to model sophistication. A powerful model without auditability is immature. A simple local agent with strong policy controls, high-quality domain data, and documented approval boundaries may be more mature for enterprise use. This view is consistent with data governance, digital transformation, and management analytics research, which argue that organisational value is created when technology is embedded in accountable processes rather than simply deployed as a tool (Bharadwaj et al., 2013; Vial, 2019; Verhoef et al., 2021; Khatri and Brown, 2010; Abraham et al., 2019; Klievink et al., 2017; Lu et al., 2024).

Table IV. Implementation roadmap, assurance evidence, and monitored failure modes.

Stage	Technical Priority	Decision-Control Requirement	Assurance Evidence	Failure Mode to Monitor
1. Single-domain pilot	Validated tools and local retrieval	Read-only or advisory outputs	Prompt log, tool log, domain review	Hallucinated recommendation or unsupported evidence
2. Two-domain workflow	Policy routing and summary exchange	Human approval before action	Access decision record and explanation	Improper sharing or misunderstood domain rule
3. Enterprise mesh	Common capability contracts and audit schema	Role-based execution permissions	Cross-domain traceability report	Cascading tool invocation or routing loop
4. Federated ecosystem	Privacy-preserving analytics and external	Legal and governance sign-off	Data minimisation proof and external audit	Partner-level leakage or regulatory conflict

contracts

The roadmap also implies that procurement should change. Instead of asking only whether a vendor provides a strong model, enterprises should ask whether the system supports local policy enforcement, domain-specific capability contracts, auditable tool calls, model version records, data-minimisation options, and fallback execution. A platform that cannot answer these questions may be suitable for low-risk summarisation but not for secure multi-context decision support. Conversely, a system that supports these features can help organisations build a defensible pathway from experimentation to operational use.

Institutional support is another practical requirement. A domain-specialized decision system cannot be built by an AI team alone. It requires domain stewards, system integrators, security officers, legal advisors, and operational managers. The governance board should define common metadata standards, but it should not centralise every policy decision. A federated operating model is needed: central standards for interoperability and risk classification, local control for data exposure and action permissions. This arrangement mirrors the structure of the proposed technical architecture and reduces the organisational resistance that often accompanies centralised transformation projects.

A phased roadmap also helps organisations avoid the common trap of using pilots as demonstrations rather than as learning systems. Each pilot should produce reusable artefacts: a domain capability catalogue, a tested policy template, a model evaluation card, an audit schema, a user feedback report, and a failure-mode register. These artefacts become the foundation for the next stage. If pilots produce only demos, the organisation repeatedly starts from zero. If pilots produce reusable governance artefacts, every new domain becomes cheaper and safer to integrate. The adoption roadmap should therefore measure not only user satisfaction or automation rate but also the accumulation of reusable decision-system infrastructure.

The roadmap should also include sunset and rollback planning. Domain agents will evolve as business processes change, models are updated, and regulations shift. Without version control, an organisation may lose the ability to explain why a past recommendation was made. Every capability contract should therefore include a version identifier, model version, data-source version, and policy version. When a domain changes its rules, dependent workflows should be notified. When a model is replaced, past decision logs should remain interpretable. These practices may appear administrative, but they are essential for accountability in complex decision systems.

A final adoption concern is user education. Employees should not be trained merely to prompt the system; they should understand when an agent is giving a retrieval answer, a probabilistic inference, a policy-constrained recommendation, or an executable workflow proposal. Clear user literacy reduces over-trust and helps staff challenge recommendations when the system fails to recognise a local exception.

VIII. DISCUSSION

The framework proposed in this article reframes agentic AI from a model-centric technology into an enterprise decision architecture. This shift has theoretical and practical implications. Theoretically, it connects agentic AI research with data governance, process management, human-AI collaboration, privacy engineering, and enterprise architecture. Practically, it suggests that successful deployment depends on domain capability

contracts, not on model fluency alone. The uploaded source paper focuses on mesh multi-context protocol logic. This article extends that logic by treating domain-specialized analytics as the basis for secure enterprise decisions.

Several limitations should be acknowledged. The scenario-based evaluation is analytical rather than empirical. It clarifies architecture-level trade-offs but does not substitute for live deployment evidence. Future research should test the framework in real enterprise settings, especially in domains with strong regulatory constraints such as finance, healthcare, public administration, and industrial operations. Researchers should measure not only task accuracy but also decision latency, privacy leakage risk, operator trust, explanation quality, human override rates, incident recovery time, and total governance cost.

A second limitation concerns the practical difficulty of defining domain boundaries. Real enterprises are messy. Processes overlap, data ownership is contested, and exceptions often matter more than rules. Domain-specialized architecture may be difficult to implement when business units disagree about data authority or when legacy systems encode outdated processes. Nevertheless, this difficulty is precisely why domain specialization is necessary. Without explicit boundaries, a central agent simply inherits organisational ambiguity and may translate it into unsafe action.

A third issue concerns future model capability. As language models become stronger, it may be tempting to return to centralised architectures. However, stronger reasoning does not remove the need for governance. A model that can reason more powerfully can also make more consequential mistakes if its action space is unconstrained. The safest future architecture is likely to combine powerful general models for broad reasoning with small specialised models and deterministic functions for local execution. This hybrid approach preserves flexibility while maintaining control. The selected literature on AI, cybersecurity, federated learning, blockchain, IoT, and Industry 4.0 supports this direction by showing that security, interoperability, and contextual integration are recurring requirements across digital enterprise systems (Lu, 2017b; Xu et al., 2021; Chen et al., 2024; Lu, 2022; Zheng and Lu, 2022; Lu and Ning, 2020; Lu and Zheng, 2020).

The framework also raises questions about responsibility in distributed AI systems. If a recommendation is produced through five domain nodes, responsibility cannot be assigned only to the model that generated the final text. Responsibility must be distributed across the domain that supplied evidence, the policy that authorised disclosure, the coordinator that aggregated results, the human role that approved action, and the organisation that designed the process. This does not mean responsibility becomes vague. On the contrary, it means that responsibility must be explicitly represented in system logs and governance documents. The domain-specialized approach is valuable precisely because it creates places where responsibility can be recorded.

A further research challenge is balancing standardisation with local adaptation. Too little standardisation makes domain nodes difficult to connect. Too much standardisation erases the local context that makes domain specialization valuable. Future work should examine which elements must be standardised--such as identity, capability descriptions, policy tags, audit fields, and evidence formats--and which elements should remain local--such as scoring models, risk thresholds, explanation templates, and approval rules. This balance will determine whether secure multi-context agentic analytics can scale beyond isolated enterprise pilots.

Finally, the framework suggests a new research agenda around agentic decision observability. Observability in software engineering usually refers to metrics, logs, and traces that explain system behaviour. Agentic decision observability should include additional fields: intent interpretation, evidence retrieval path, policy evaluation result, domain delegation sequence, confidence shift across nodes, explanation generated for the user, and approval or override status. These traces would enable researchers to analyse how enterprise agents actually reason across contexts. They would also help managers identify bottlenecks, unsafe patterns, and domains where additional training or governance is needed.

IX. CONCLUSION

This article has developed a framework for domain-specialized agentic AI analytics in secure multi-context enterprise decision systems. The central claim is that enterprise agents should not be designed only for access to multiple contexts. They should be designed for governed participation in multiple contexts. Domain nodes should retain authority over their data, rules, tools, and action boundaries, while a secure mesh should coordinate evidence, summaries, and recommendations across the enterprise. This design reduces privacy risk, improves fault isolation, and supports incremental integration with legacy systems.

The proposed framework contributes an architecture, decision cycle, domain-function model, governance design, scenario-based evaluation, and implementation roadmap. Its practical value lies in showing how enterprises can adopt agentic AI without giving uncontrolled autonomy to a central model. Its theoretical value lies in connecting language agents with data governance and secure enterprise analytics. The next phase of research should move from architectural reasoning to empirical validation, testing the framework in real organisational workflows and measuring whether domain-specialized governance improves decision quality, resilience, and accountability over time.

For practitioners, the main recommendation is to treat domain-specialized agentic AI analytics as an enterprise capability rather than as a model deployment. The capability includes model selection, data governance, tool validation, policy routing, explanation design, human oversight, and post-decision monitoring. Organisations that invest only in model access will create impressive demonstrations but weak operational systems. Organisations that invest in domain contracts and assurance infrastructure will be better positioned to convert agentic AI from an experimental interface into a reliable decision-support capability.

In summary, the central design lesson is that secure agentic AI is less about asking a model to be careful and more about building an environment in which careful behaviour is structurally enforced. Domain nodes, capability contracts, policy gates, audit trails, and human approval points make safe reasoning more likely because they constrain the space of possible actions before the model chooses. This principle is especially important in enterprises where data is fragmented, compliance obligations are uneven, and the cost of failure can be high. The proposed approach is not a final solution to every risk of agentic AI, but it offers a disciplined foundation for converting multi-context access into accountable multi-context decision support.

The article also suggests that future enterprise AI maturity will be measured by the quality of governance infrastructure rather than only by model sophistication. Organisations that can describe their domain capabilities, explain their policy gates, audit their agentic workflows, and contain domain failures will be better prepared to use increasingly autonomous systems. As agentic AI becomes more capable, the enterprise

challenge will be to preserve meaningful human and organisational control. Domain-specialized analytics provides one pathway toward that goal by making autonomy local, evidence-based, and accountable.

The practical implication is direct: before connecting more tools, organisations should define more boundaries. Boundaries create the conditions under which autonomy can become useful rather than dangerous.

Such boundaries also make future empirical testing clearer and more reproducible across organisations.

REFERENCES

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33. https://doi.org/10.48550/arXiv.2005.14165

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.03762

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT, 4171-4186. https://doi.org/10.18653/v1/N19-1423

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., et al. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35. https://doi.org/10.48550/arXiv.2203.02155

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35. https://doi.org/10.48550/arXiv.2201.11903

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. Proceedings of ICLR. https://doi.org/10.48550/arXiv.2210.03629

Schick, T., Dwivedi-Yu, J., Dessi, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., et al. (2023). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36. https://doi.org/10.48550/arXiv.2302.04761

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33. https://doi.org/10.48550/arXiv.2005.11401

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. Proceedings of the ACM UIST, 1-22. https://doi.org/10.1145/3586183.3606763

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., et al. (2022). Holistic evaluation of language models. arXiv. https://doi.org/10.48550/arXiv.2211.09110

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., et al. (2021). On the opportunities and risks of foundation models. arXiv. https://doi.org/10.48550/arXiv.2108.07258

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of ACM FAccT, 610-623. https://doi.org/10.1145/3442188.3445922

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P. S., et al. (2021). Ethical and social risks of harm from language models. arXiv. https://doi.org/10.48550/arXiv.2112.04359

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2303.11366

Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., et al. (2023). Augmented language models: A survey. arXiv. https://doi.org/10.48550/arXiv.2302.07842

Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., et al. (2019). Guidelines for human-AI interaction. Proceedings of CHI, 1-13. https://doi.org/10.1145/3290605.3300233

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., et al. (2019). Model cards for model reporting.

Proceedings of ACM FAccT, 220-229. https://doi.org/10.1145/3287560.3287596

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé, H., & Crawford, K. (2021). Datasheets for datasets. Communications of the ACM, 64(12), 86-92. https://doi.org/10.1145/3458723

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., et al. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. Proceedings of ACM FAccT, 33-44. https://doi.org/10.1145/3351095.3372873

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. Proceedings of KDD, 1135-1144. https://doi.org/10.1145/2939672.2939778

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1705.07874

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges. Information Fusion, 58, 82-115. https://doi.org/10.1016/j.inffus.2019.12.012

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), 1-42. https://doi.org/10.1145/3236009

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv. https://doi.org/10.48550/arXiv.1702.08608

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics - Part A, 30(3), 286-297. https://doi.org/10.1109/3468.844354

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. Knowledge Engineering Review, 10(2), 115-152. https://doi.org/10.1017/S0269888900008122

Shoham, Y. (1993). Agent-oriented programming. Artificial Intelligence, 60(1), 51-92. https://doi.org/10.1016/0004-3702(93)90034-9

Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. Autonomous Robots, 8, 345-383. https://doi.org/10.1109/5254.875390

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. Proceedings of ACM STOC, 169-178. https://doi.org/10.1145/1536414.1536440

Yao, A. C. (1982). Protocols for secure computations. Proceedings of IEEE FOCS, 160-164. https://doi.org/10.1145/800070.802141

Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play any mental game. Proceedings of ACM STOC, 218-229. https://doi.org/10.1145/28395.28420

Dwork, C. (2006). Differential privacy. In Automata, Languages and Programming, ICALP 2006, 1-12. https://doi.org/10.1007/11787006_1

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. Proceedings of ACM CCS, 308-318. https://doi.org/10.1145/2976749.2978318

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., et al. (2017). Practical secure aggregation for privacy-preserving machine learning. Proceedings of ACM CCS, 1175-1191. https://doi.org/10.1145/3133956.3133982

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. Proceedings of ACM CCS, 1310-1321. https://doi.org/10.1145/2810103.2813687

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of AISTATS. https://doi.org/10.48550/arXiv.1602.05629

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., et al. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1-2), 1-210. https://doi.org/10.1561/2200000083

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60. https://doi.org/10.1109/MSP.2020.2975749

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 10(2), 1-19. https://doi.org/10.1145/3298981

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23(3), 1622-1658. https://doi.org/10.1109/COMST.2021.3075439

Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2018). SoK: Security and privacy in machine learning. IEEE European Symposium on Security and Privacy, 399-414. https://doi.org/10.1109/EuroSP.2018.00035

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. IEEE Symposium on Security and Privacy, 3-18. https://doi.org/10.1109/SP.2017.41

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., et al. (2013). Evasion attacks against machine learning at test time. ECML PKDD, 387-402. https://doi.org/10.1007/978-3-642-40994-3_25

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1412.6572

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1706.06083

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. Proceedings of AISTATS. https://doi.org/10.48550/arXiv.1807.00459

Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. arXiv. https://doi.org/10.48550/arXiv.1803.01498

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. arXiv. https://doi.org/10.48550/arXiv.1703.02757

Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., & Venkatraman, N. (2013). Digital business strategy: Toward a next generation of insights. MIS Quarterly, 37(2), 471-482. https://doi.org/10.25300/MISQ/2013/37:2.3

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. Journal of Strategic Information Systems, 28(2), 118-144. https://doi.org/10.1016/j.jsis.2019.01.003

Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Dong, J. Q., Fabian, N., & Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. Journal of Business Research, 122, 889-901. https://doi.org/10.1016/j.jbusres.2019.09.022

Trkman, P. (2010). The critical success factors of business process management. International Journal of Information Management, 30(2), 125-134. https://doi.org/10.1016/j.ijinfomgt.2009.07.003

van der Aalst, W. M. P. (2016). Process Mining: Data Science in Action. Springer. https://doi.org/10.1007/978-3-662-49851-4

Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The journey so far and challenges ahead. IEEE Software, 35(3), 24-31. https://doi.org/10.1109/MS.2018.2141039

Soldani, J., Tamburri, D. A., & van den Heuvel, W. J. (2018). The pains and gains of microservices: A systematic grey literature review. Journal of Systems and Software, 146, 215-232. https://doi.org/10.1016/j.jss.2018.09.082

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables DevOps. IEEE Software, 33(3), 42-52. https://doi.org/10.1109/MS.2016.64

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, today, and tomorrow. In Present and Ulterior Software Engineering, 195-216. https://doi.org/10.1007/978-3-319-67425-4_12

Khatri, V., & Brown, C. V. (2010). Designing data governance. Communications of the ACM, 53(1), 148-152.

https://doi.org/10.1145/1629175.1629210

Abraham, R., Schneider, J., & vom Brocke, J. (2019). Data governance: A conceptual framework, structured review, and research agenda. International Journal of Information Management, 49, 424-438. https://doi.org/10.1016/j.ijinfomgt.2019.07.008

Klievink, B., Romijn, B. J., Cunningham, S., & de Bruijn, H. (2017). Big data and the public sector: Uncertainties and readiness. Government Information Quarterly, 34(2), 267-283. https://doi.org/10.1016/j.giq.2016.08.001

Floridi, L., & Cowls, J. (2019). A unified framework of five principles for AI in society. Harvard Data Science Review, 1(1). https://doi.org/10.1162/99608f92.8cd550d1

Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1, 389-399. https://doi.org/10.1038/s42256-019-0088-2

Vinuesa, R., Azizpour, H., Leite, I., Balaam, M., Dignum, V., Domisch, S., et al. (2020). The role of artificial intelligence in achieving the Sustainable Development Goals. Nature Communications, 11, 233. https://doi.org/10.1038/s41467-019-14108-y

Brynjolfsson, E., Li, D., & Raymond, L. R. (2023). Generative AI at work. NBER Working Paper 31161. https://doi.org/10.3386/w31161

Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). GPTs are GPTs: An early look at the labor market impact potential of large language models. arXiv. https://doi.org/10.48550/arXiv.2303.10130

Raisch, S., & Krakowski, S. (2021). Artificial intelligence and management: The automation-augmentation paradox. Academy of Management Annals, 15(1), 192-210. https://doi.org/10.5465/annals.2018.0072

Jarrahi, M. H. (2018). Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. Business Horizons, 61(4), 577-586. https://doi.org/10.1016/j.bushor.2018.05.001

Kellogg, K. C., Valentine, M. A., & Christin, A. (2020). Algorithms at work: The new contested terrain of control. Academy of Management Annals, 14(1), 366-410. https://doi.org/10.5465/annals.2018.0174

Lu, Y. (2019). Artificial intelligence: A survey on evolution, models, applications and future trends. Journal of Management Analytics, 6(1), 1-29. https://doi.org/10.1080/23270012.2019.1570365

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224. https://doi.org/10.1016/j.jii.2021.100224

Lu, Y. (2025). The current status and developing trends of Industry 4.0: A review. Information Systems Frontiers, 27(1), 215-234. https://doi.org/10.1007/s10796-021-10221-w

Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration, 6, 1-10. https://doi.org/10.1016/j.jii.2017.04.005

Lu, Y. (2017). Cyber physical system (CPS)-based Industry 4.0: A survey. Journal of Industrial Integration and Management, 2(3), 1750014. https://doi.org/10.1142/S2424862217500142

Xu, L. D., Lu, Y., & Li, L. (2021). Embedding blockchain technology into IoT for security: A survey. IEEE Internet of Things Journal, 8(13), 10452-10473. https://doi.org/10.1109/JIOT.2021.3060508

Chen, Y., Lu, Y., Bulysheva, L., & Kataev, M. Y. (2024). Applications of blockchain in Industry 4.0: A review. Information Systems Frontiers, 26(5), 1715-1729. https://doi.org/10.1007/s10796-022-10248-7

Lu, Y. (2022). Implementing blockchain in information systems: A review. Enterprise Information Systems, 16(12), 1876-1907. https://doi.org/10.1080/17517575.2021.2008513

Zheng, X. R., & Lu, Y. (2022). Blockchain technology: Recent research and future trend. Enterprise Information Systems, 16(12), 1939895. https://doi.org/10.1080/17517575.2021.1939895

Lu, Y., & Ning, X. (2020). A vision of 6G-5G's successor. Journal of Management Analytics, 7(3), 301-320. https://doi.org/10.1080/23270012.2020.1802622

Lu, Y., & Zheng, X. (2020). 6G: A survey on technologies, scenarios, challenges, and the related issues. Journal of Industrial Information Integration, 19, 100158. https://doi.org/10.1016/j.jii.2020.100158

JAIAA **RESEARCH ARTICLE**
Journal of AI Analytics and Applications **Vol. 3, No. 2 (2025), pp. 28-49**
ISSN 3067-7386 | Open Access | Peer-Reviewed **DOI: 10.63646/jaiaa.2025.030202**

Lu, Y., Ivanov, L. A., Wang, F., Pisarenko, Z. V., & Ye, C. (2024). Management analytics: A bibliometric analysis. Nanotechnologies in Construction, 16(3), 257-266. https://doi.org/10.15828/2075-8545-2024-16-3-257-266