World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

# Runtime assurance for enterprise agentic AI systems: A policy-gated control model with quantitative autonomy-risk scoring

Kwan Hong TAN *

*School of Business, Singapore University of Social Sciences, Singapore.*

World Journal of Advanced Research and Reviews, 2026, 31(01), 512–522

Publication history: Received on 01 June 2026; revised on 07 July 2026; accepted on 09 July 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.31.1.1872

## Abstract

Enterprise adoption of agentic artificial intelligence (AI) is moving from passive text generation toward autonomous planning, tool use and cross-system workflow execution. This transition creates a control gap: conventional model governance evaluates outputs or development processes, while agentic systems create risk through sequential actions, delegated authority and changing operating context. This paper develops a runtime assurance architecture (RAA) for enterprise agentic AI and formalizes a quantitative Autonomy-Risk Exposure (ARE) score for deciding when an agent may execute, must be sandboxed, requires human approval or must be blocked. A design-science method was used to synthesize requirements from AI risk-management standards, generative AI security guidance and agentic AI threat literature. The model was then evaluated through a reproducible scenario simulation of 2,000 enterprise agent episodes across knowledge assistance, data retrieval, internal workflow and external transaction tasks. Results show that the full RAA configuration reduced mean ARE from 35.7 to 24.4 points (31.5% reduction), decreased invalid or policy-conflicting actions from 9.8% to 4.6%, eliminated unsupervised pass-through of high-risk invalid actions in the simulated environment, and improved mean audit evidence coverage from 0.61 to 0.89. The control benefit was achieved with a mean latency overhead of 95 ms and human approval for 12.6% of episodes. The paper contributes a practical reference architecture, a risk-scoring equation, a policy decision algorithm and implementation guidance for organizations deploying agentic AI in regulated or high-consequence workflows.

**Keywords:** Agentic AI; Runtime Assurance; AI Governance; Risk Scoring; Human Oversight; Enterprise Automation

## 1. Introduction

The practical frontier of artificial intelligence is shifting from systems that generate recommendations to systems that plan, call tools, retrieve data, trigger workflow actions and coordinate across multiple software environments. This development is often described as agentic AI. In contrast to conventional predictive models, an agentic system may decompose a goal into subtasks, select intermediate actions, use external APIs, update memory and execute operations over time. These capabilities create new productivity opportunities, but they also widen the risk surface because harm may arise not only from model output but also from action sequencing, authority delegation, tool misuse, hidden state and inadequate evidence trails.

Existing governance frameworks provide essential foundations but do not fully specify a runtime control mechanism for enterprise agents. The NIST AI Risk Management Framework emphasizes the governance, mapping, measurement and management of AI risks [1], while the NIST Generative AI Profile extends those concerns to risks such as confabulation, information integrity, data privacy and value-chain exposure [2]. The EU AI Act establishes a risk-based regulatory approach to AI systems [3], and ISO/IEC 42001:2023 provides a certifiable management-system approach

* Corresponding author: Kwan Hong TAN

for responsible AI development and use [4]. These frameworks are valuable at organizational and lifecycle levels. However, an enterprise agent must also be controlled at the moment it attempts to act.

Security guidance has become more concrete as large language model (LLM) applications and agentic systems have matured. OWASP identifies prompt injection, insecure output handling, sensitive information disclosure, model denial of service, excessive agency and supply-chain vulnerabilities as important categories for LLM applications [5]. Recent OWASP work on agentic AI further highlights threats such as goal hijacking, tool misuse, privilege abuse, unexpected code execution and agentic supply-chain compromise [6]. Academic reviews similarly emphasize that LLM-based agents create distinctive security and privacy risks because they combine model reasoning with memory, tools, multi-step planning and external resources [7-9].

The present paper addresses this operational gap by asking a technical design question: How can an enterprise deploy agentic AI with measurable runtime control rather than relying only on static governance policies or post-hoc audits? The paper proposes a runtime assurance architecture (RAA) that separates goal generation from execution authority. The architecture allows an agent to propose actions, but every consequential action must pass through a policy gate, an autonomy-risk scorer, an identity-aware tool proxy and an evidence logger before execution.

This study builds on five strands of prior work. First, AI risk-management literature shows that trustworthiness requires explicit measurement, accountability and risk controls rather than abstract ethical principles alone [1,2,10,11]. Second, human-centered AI research argues for systems that preserve human agency and meaningful oversight [12,13]. Third, documentation approaches such as model cards and datasheets show how AI systems can be made more auditable through structured evidence [14,15]. Fourth, recent work on AI-form organizations, AI as stakeholder, dynamic equilibrium ethics, technological displacement and digital productivity paradoxes suggests that AI value creation depends on the fit between automation capability, institutional adaptation and governance capacity [16-20]. Fifth, agentic AI security literature shows that tool-using autonomy creates risks that conventional model-level evaluations do not capture [6-9].

The paper makes four contributions. It defines an Autonomy-Risk Exposure (ARE) score that combines task sensitivity, tool criticality, autonomy tier, uncertainty, novelty, externality, irreversibility and regulatory salience. It specifies a policy-gated decision algorithm that routes tasks to execution, sandboxing, human approval or blocking. It presents a reference architecture for enterprise deployment with audit evidence, least privilege and rollback mechanisms. Finally, it evaluates the proposed architecture through a deterministic simulation of enterprise agent episodes, allowing the control trade-offs between risk reduction, oversight load and latency to be examined.

## 2. Materials and methods

### 2.1. Design-science research approach

The study uses a design-science research approach because the objective is to construct and evaluate an artifact for a practical class of information-systems problems: controlling enterprise agentic AI at runtime. The artifact consists of a formal risk score, a policy-gated decision algorithm, a runtime assurance architecture and a simulation-based evaluation protocol. The design problem is not whether AI agents can generate useful outputs, but whether their autonomy can be bounded by auditable controls during action execution.

The design requirements were synthesized from five source categories: AI risk-management frameworks [1,2], regulatory and management-system requirements [3,4], LLM and agentic security threat taxonomies [5-9], human-centered oversight principles [12,13], and AI documentation/accountability literature [14,15]. These sources were translated into operational requirements: identity isolation, least-privilege tool access, continuous risk scoring, context classification, policy matching, human approval for high-consequence actions, sandboxing for uncertain or irreversible operations, output validation, rollback mechanisms and evidence logging.

### 2.2. System model

An enterprise agent episode is represented as a tuple $E = \langle g, c, m, a, t, p, e \rangle$, where $g$ is the user or workflow goal, $c$ is the operating context, $m$ is the model or agent policy, $a$ is the proposed action, $t$ is the tool or resource to be invoked, $p$ is the applicable policy set and $e$ is the evidence record. The architecture assumes that the agent may reason and propose actions, but it does not directly hold execution authority over high-consequence tools. Execution authority is delegated to a tool proxy that evaluates policy, identity, scope and risk before allowing the action to proceed.

The architecture distinguishes four task classes. Knowledge assistance involves generating or summarizing information without direct tool execution. Data retrieval involves accessing documents, databases or internal knowledge stores. Internal workflow tasks modify enterprise systems, such as creating tickets or updating records. External transaction tasks interact with third parties, financial systems, public communication channels or regulated domains. The expected control intensity increases from knowledge assistance to external transactions because the potential harm from error, privacy leakage, unauthorized action and irreversible execution increases.

$E = < g, c, m, a, t, p, e >$

### 2.3. Autonomy-Risk Exposure score

The core measurement artifact is the Autonomy-Risk Exposure (ARE) score. ARE is designed as a runtime score from 0 to 100 that estimates how much operational risk is present when an agent attempts to execute a specific action in a specific context. It is not a model-accuracy score and does not replace domain-specific validation. Instead, it acts as an execution-control index that determines whether the proposed action may proceed autonomously or must be routed to additional controls.

Eight normalized variables are used. $S$ is data sensitivity, $C$ is tool criticality, $A$ is autonomy tier, $U$ is epistemic uncertainty, $N$ is task novelty, $E$ is externality of action, $I$ is irreversibility and $R$ is regulatory salience. Each variable is scaled from 0 to 1. Data sensitivity is high when personal, confidential, financial or health-related information is involved. Tool criticality is high when the tool can modify records, transfer value, publish content or trigger downstream effects. Autonomy tier represents how much discretion the agent has in planning and execution. Uncertainty captures low confidence, conflicting evidence or weak retrieval support. Novelty measures deviation from known approved task patterns. Externality captures effects outside the organization. Irreversibility captures actions that cannot easily be undone. Regulatory salience captures whether the action falls under high-risk, compliance-sensitive or formally documented process areas.

The score is calculated as a weighted additive model. The weights used in the simulation reflect the relative importance of data sensitivity, tool criticality and autonomy for enterprise risk. Organizations may recalibrate the weights according to domain, risk appetite and regulatory obligations.

$$ARE = 100 \times (0.18S + 0.18C + 0.15A + 0.12U + 0.10N + 0.10E + 0.10I + 0.07R)$$

### 2.4. Policy-gated decision algorithm

The policy gate receives the agent-proposed action and calculates ARE before execution. If ARE is below the autonomous execution threshold and no hard policy rule is violated, the action may proceed through least-privilege tool access. If the action involves sensitive data, external systems or moderate uncertainty, the gate may restrict the tool scope, force retrieval validation, mask data or route the action through a sandbox. If ARE exceeds the human-review threshold, the action requires explicit approval. If the action violates a prohibited policy, creates unacceptable regulatory exposure or combines high irreversibility with high uncertainty, it is blocked and recorded as an exception.

The gate therefore behaves as a runtime safety case. It requires the system to produce evidence that a proposed action is permitted, proportionate, logged and reversible enough for the intended context. This differs from conventional access control because it evaluates not only who or what is calling a tool, but also why the tool is being called, what data are involved, how uncertain the system is, and whether the resulting action can be reversed.

#### *2.4.1. Algorithm 1. Runtime policy-gated execution*

Input: goal $g$, context $c$, proposed action $a$, tool $t$, policy set $p$, model confidence $q$, evidence state $e$

- Classify task class and retrieve applicable policy rules.
- Estimate $S$, $C$, $A$, $U$, $N$, $E$, $I$ and $R$ for the proposed action.
- Calculate ARE using the weighted scoring equation.
- If a hard prohibition is matched, block the action and create an exception record.
- Else if ARE $\geq 60$ or regulatory salience is high, route to human approval.
- Else if tool criticality, uncertainty or irreversibility is moderate, sandbox or restrict the action.
- Else execute through least-privilege tool proxy.
- Log prompt, context, retrieved sources, tool call, policy decision, model version, approval record and residual risk.

- Update thresholds and exception rules using monitored outcomes.

### 2.5. Runtime assurance architecture

Figure 1 presents the proposed RAA. The architecture places the agent planner inside a bounded control environment. The agent may interpret goals, create plans and propose tool calls, but execution authority sits outside the planner. The policy gate calculates ARE, matches policy rules and selects a control route. The runtime assurance layer monitors telemetry, applies least privilege, validates outputs, manages sandbox execution and writes evidence to a ledger. Human approval is treated as an explicit control path rather than an informal escalation practice.

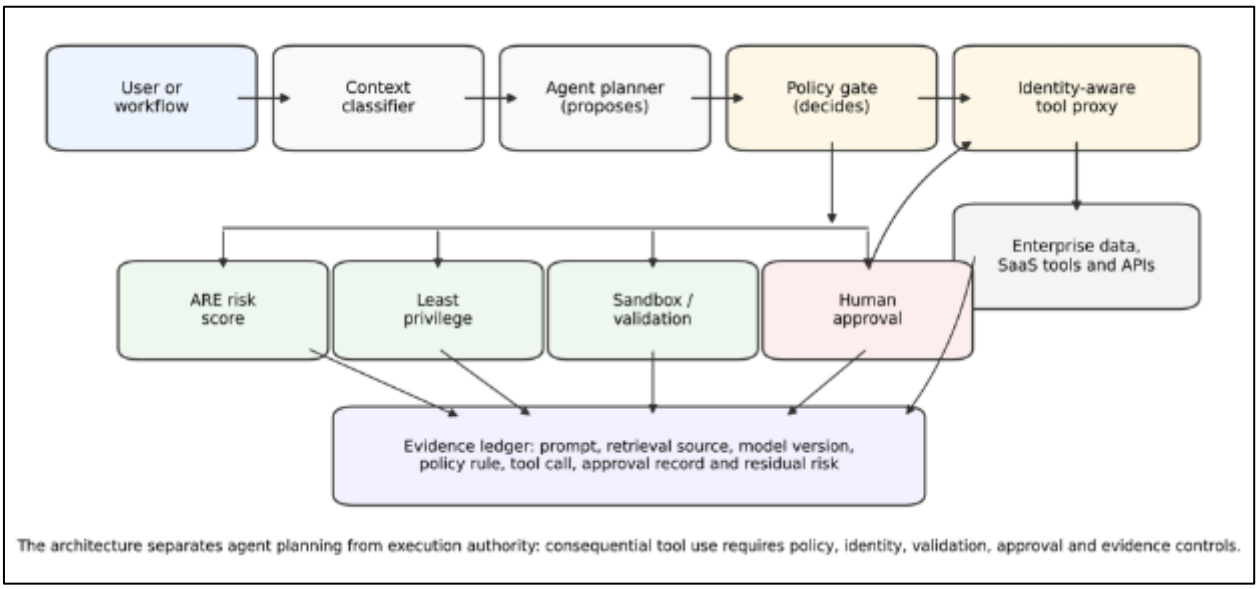


**Figure 1** Runtime assurance architecture for enterprise agentic AI

### 2.6. Simulation protocol

To evaluate the architecture without using proprietary enterprise records or human participant data, a deterministic scenario simulation was conducted with a fixed random seed. The simulation generated 2,000 agent episodes distributed across four task classes: knowledge assistance, data retrieval, internal workflow and external transaction. For each episode, the eight risk variables were sampled from class-conditional distributions. External transactions were assigned higher expected tool criticality, externality, irreversibility and regulatory salience than knowledge assistance tasks. This reflects a conservative enterprise assumption that tool-mediated external actions should receive stronger controls than low-consequence information tasks.

Three configurations were evaluated. The baseline configuration allowed agent actions without runtime risk controls beyond ordinary authentication. The policy-gate configuration applied ARE thresholds but not the full runtime assurance layer. The full RAA configuration applied least privilege, sandboxing, retrieval validation, human approval, blocking and evidence logging. The primary outcome was mean residual ARE. Secondary outcomes included invalid or policy-conflicting action rate, high-risk unsupervised pass-through, audit evidence coverage, mean latency overhead, human approval load, blocked-action rate, false-positive escalation rate and task-completion rate.

The evaluation is intentionally presented as a technical simulation rather than empirical field evidence. Its purpose is to test internal consistency, compare control configurations and make trade-offs visible. Field deployment would require organization-specific calibration, real incident review, red-team testing and monitoring over time.

### 2.7. Statistical analysis

Descriptive statistics were calculated across all simulated episodes and by task class. Risk reduction was computed as one minus the ratio of residual ARE to baseline ARE. Invalid-action suppression was computed as one minus the ratio of invalid or policy-conflicting actions after control to those before control. Human approval load, block rate and false-positive escalation rate were calculated as proportions of total episodes. Because the simulation is deterministic and

not based on a sample from a real operational population, inferential statistics are not used to claim external generalizability. The results should be interpreted as scenario-based performance estimates under the stated assumptions.

### 2.8. Risk variables and control routes

**Table 1** Risk variables used in the Autonomy-Risk Exposure score

| Variable | Operational meaning | Examples of high-score conditions | Primary control implication |
|---|---|---|---|
| S: Data sensitivity | Sensitivity of data touched by the action | Personal, financial, health, confidential or regulated data | Masking, retrieval limits, approval |
| C: Tool criticality | Potential consequence of the tool call | Write access, payments, public posting, record deletion | Least privilege, sandbox, transaction limits |
| A: Autonomy tier | Degree of agent discretion | Multi-step planning, self-selection of tools, unattended execution | Human approval, narrower scope |
| U: Uncertainty | Weak confidence or contradictory evidence | Low retrieval support, conflicting sources, ambiguous user intent | Validation and clarification |
| N: Novelty | Deviation from known approved patterns | New workflow, unseen tool combination, unusual destination | Sandbox, additional review |
| E: Externality | Impact beyond internal environment | Customer communication, vendor action, regulator-facing output | Approval and audit trail |
| I: Irreversibility | Difficulty of undoing the action | Deletion, payment, binding commitment, credential change | Block, approval or rollback plan |
| R: Regulatory salience | Exposure to formal compliance requirements | High-risk AI use, employment, finance, health or safety context | Documentation and formal review |

## 3. Results and discussion

### 3.1. Reference architecture behavior

The proposed architecture operationalizes a simple principle: the agent may generate a plan, but the enterprise control layer decides whether the plan may be executed. This separation matters because LLM-based agents can appear fluent while still misinterpreting goals, accepting malicious instructions, overusing tools or acting beyond appropriate authority. By routing all meaningful tool calls through a policy gate, the enterprise can enforce risk appetite at the action level rather than treating the agent as a trusted extension of the user.

Figure 2 summarizes the control loop. The system first classifies the task context, estimates risk variables, calculates ARE and checks applicable policies. The action is then approved, sandboxed, escalated or blocked. Every route produces an evidence record. This means that even rejected actions contribute to governance learning because exceptions reveal where prompts, policies, retrieval sources or tool permissions should be improved.

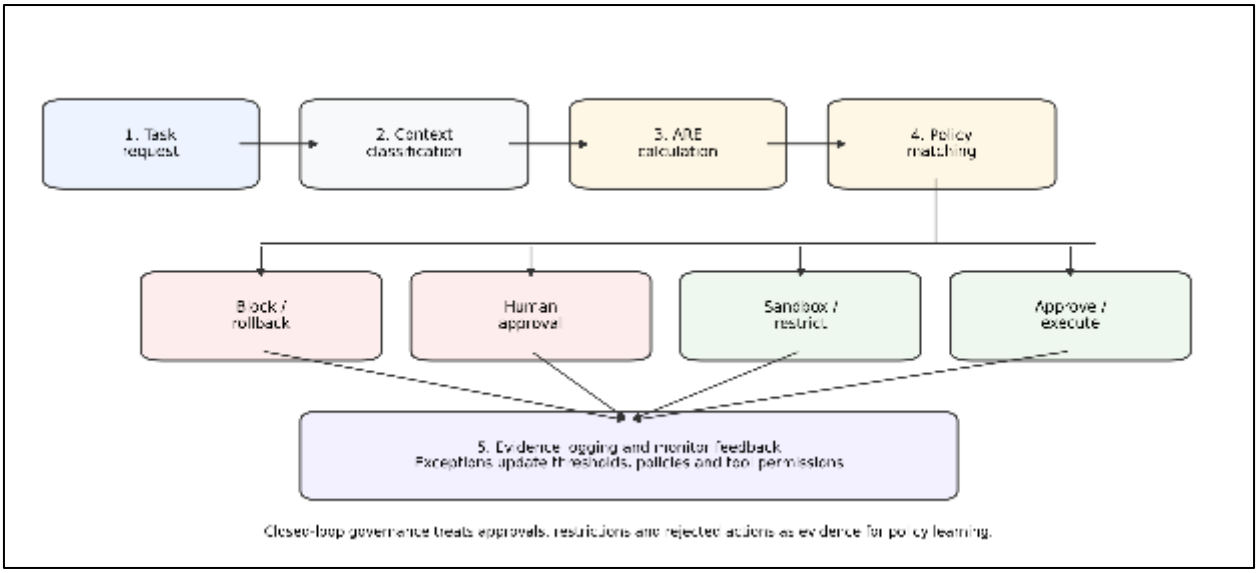


**Figure 2** Policy-gated control loop for runtime assurance

### 3.2. Simulation outcomes

Across 2,000 simulated episodes, the baseline mean ARE was 35.7. The full RAA configuration reduced mean residual ARE to 24.4, representing a 31.5% reduction in autonomy-risk exposure. The invalid or policy-conflicting action rate decreased from 9.8% before control to 4.6% after control, corresponding to 53.6% suppression. In the simulated environment, high-risk invalid actions did not pass through without human approval or blocking. Mean audit evidence coverage increased from 0.61 to 0.89, while mean latency increased from 217 ms to 312 ms. The resulting average overhead was 95 ms per episode.

The system required human approval in 12.6% of episodes and blocked 0.6% of episodes. This indicates that the proposed control model did not reduce risk by escalating every action to a human reviewer. Instead, most low- and moderate-risk actions remained executable under least-privilege or validation controls. The autonomous execution rate after controls was 70.1%, while the task-completion rate remained 99.4% because only a small number of extreme policy-violating or high-irreversibility episodes were blocked.

**Table 2** Scenario simulation results by enterprise task class (n = 2,000 episodes)

| Task | n | Base ARE | Residual ARE | Invalid before % | Invalid after % | Approval % | Block % | Latency (ms) |
|---|---|---|---|---|---|---|---|---|
| K-assist | 645 | 27.6 | 22.6 | 6.5 | 4.7 | 3.4 | 0.0 | 49.0 |
| Retrieval | 550 | 33.4 | 24.3 | 8.4 | 4.4 | 8.0 | 0.0 | 78.0 |
| Workflow | 492 | 41.2 | 26.0 | 10.6 | 3.5 | 18.3 | 0.6 | 125.0 |
| External | 313 | 47.3 | 26.1 | 17.9 | 6.4 | 30.7 | 2.9 | 173.0 |

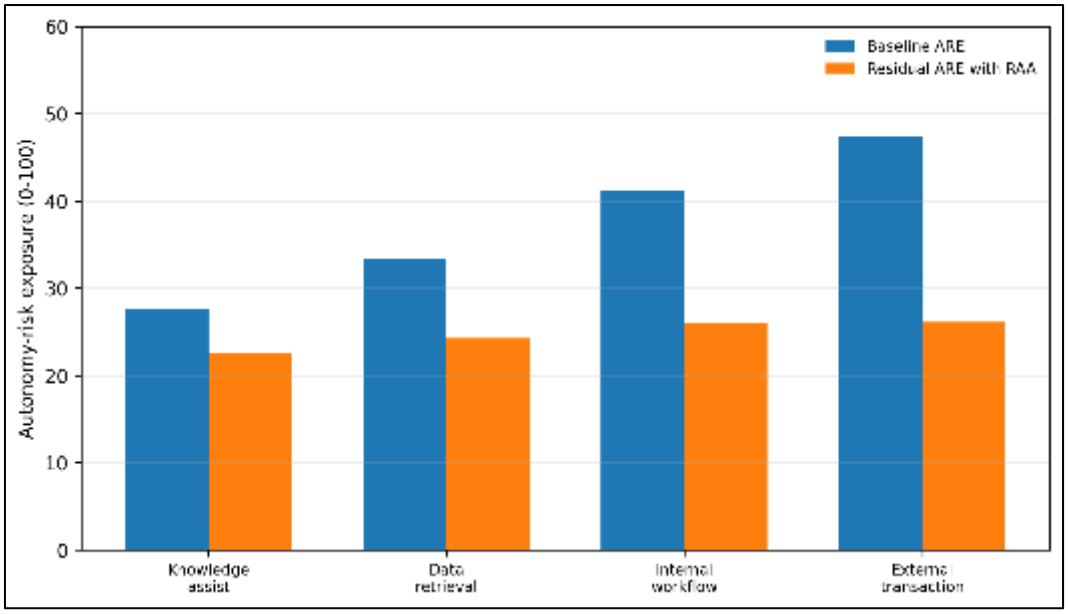


**Figure 3** Reduction in Autonomy-Risk Exposure by task class

### 3.3. Control trade-offs

The results show that risk reduction is uneven across task classes. Knowledge assistance had the lowest baseline ARE and therefore received the least intervention. External transactions had the highest baseline ARE and the strongest control intensity, with 30.7% routed to human approval and 2.9% blocked. This pattern is desirable because it aligns control burden with consequence. A flat governance model would either over-control low-risk knowledge work or under-control high-risk transactions. The ARE-based model provides a middle path by increasing friction only when risk variables justify it.

Latency overhead is also uneven. Knowledge assistance incurred an average overhead of 49 ms, while external transactions incurred 173 ms. In most enterprise settings, this trade-off is likely acceptable because high-consequence transactions already require additional verification. However, the design should not assume that all workflows can tolerate the same latency. Real-time cyber defense, trading, industrial control and emergency response environments would require stricter latency budgets and possibly pre-approved bounded autonomy patterns.

The false-positive escalation rate was 4.0% overall. False positives matter because excessive escalation can undermine user trust and create review bottlenecks. The architecture therefore requires continuous calibration. If too many low-risk tasks are escalated, thresholds can be adjusted, task patterns can be whitelisted or evidence requirements can be simplified. Conversely, if incidents occur after autonomous execution, weights and thresholds should be tightened for the affected class of action.

### 3.4. Governance interpretation

The RAA model translates governance into executable system behavior. NIST, ISO and regulatory frameworks emphasize risk management, documentation, accountability and human oversight [1-4]. The proposed model operationalizes those expectations by embedding them into the agent execution path [21]. A model card or policy statement may describe an AI system, but a runtime gate determines whether a specific action is allowed in a specific context. The distinction is important for agentic AI because risk changes as the agent moves from reasoning to action.

The architecture also responds to the principle-practice gap in responsible AI. Many organizations adopt AI principles but struggle to translate them into daily engineering controls. The proposed RAA provides a bridge between abstract principles and operational controls: fairness and safety become policy constraints; accountability becomes evidence logging; human oversight becomes approval routing; privacy becomes data-sensitivity scoring and masking; security becomes identity isolation, tool-scoping and sandboxing.

The model is also consistent with prior work on AI stakeholder recognition and dynamic equilibrium ethics. Treating AI as a consequential actor in an enterprise ecosystem does not require granting it moral priority over humans. It does

require recognizing that agentic systems can affect stakeholders through delegated actions, hidden intermediaries and automated decisions [16,17]. Runtime assurance therefore preserves human agency by keeping consequential action authority outside the unconstrained agent planner.

### 3.5. Threat-control mapping

Table 3 maps common agentic AI threat categories to the controls used in the proposed architecture. The table highlights why a single defense is insufficient. Prompt injection cannot be managed only by better prompts because the agent may still call tools. Tool misuse cannot be managed only by access control because context and purpose matter. Output validation cannot replace evidence logging because organizations need to reconstruct what happened after an incident. A layered control architecture is therefore necessary.

**Table 3** Agentic AI threat-control mapping for the runtime assurance architecture

| Threat category | Failure mode | RAA control response | Evidence generated |
|---|---|---|---|
| Goal or prompt hijacking | External instruction changes the agent objective | Context classifier, policy gate, tool-call confirmation | Original prompt, injected content, policy decision |
| Tool misuse | Agent uses a permitted tool for an inappropriate purpose | Purpose-bound authorization and least privilege | Tool scope, action rationale, authorization token |
| Privilege abuse | Agent inherits excessive user or service permissions | Agent-specific identity and scoped credentials | Identity, role, tool permission, expiry time |
| Sensitive data leakage | Agent retrieves or discloses confidential information | Sensitivity scoring, masking, retrieval limits | Data class, masking decision, output check |
| Unexpected code execution | Agent produces executable or unsafe downstream output | Sandboxing and output validation | Sandbox result, validation status, blocked payload |
| Irreversible action | Agent deletes, pays, publishes or commits externally | Human approval, rollback plan or block | Approver, timestamp, residual risk, rollback status |
| Audit failure | Action cannot be reconstructed after incident | Evidence ledger and immutable event record | Prompt, model version, policy version, tool call |

### 3.6. Implementation guidance

Practical implementation should begin with a tool inventory. Organizations should classify every tool an agent can call by read/write authority, data class, reversibility, external effect and regulatory relevance. Tools should then be exposed to agents through a proxy rather than through direct credentials. The proxy should support scoped permissions, short-lived tokens, transaction limits, rate limits, policy checks and complete event logging. This prevents agent identity from becoming indistinguishable from user identity.

Second, organizations should define autonomy tiers. A Tier 0 system provides advice only. Tier 1 retrieves information. Tier 2 drafts actions for human execution. Tier 3 executes bounded internal actions. Tier 4 executes external or irreversible actions only after approval. Tier 5 represents high autonomy in high-consequence contexts and should be used only with formal assurance evidence, red-team testing, rollback design and continuous monitoring [22-25]. Autonomy tiering allows organizations to scale governance across many use cases without debating every deployment from first principles.

Third, policy should be machine-readable. Natural-language policy statements are useful for human governance but insufficient for runtime enforcement. The policy gate requires structured rules that identify prohibited actions, required approvals, evidence requirements, retention requirements, regulatory triggers and escalation paths. Rules should be versioned so that the evidence ledger can reconstruct which rule set was active when an action occurred.

Fourth, evidence should be designed before deployment. A useful evidence record should include the user goal, task class, model or agent version, prompt and system instructions, retrieval sources, tool identity, proposed action, ARE variables, policy decision, human approval record, execution result and output validation status. Evidence logging must

be privacy-aware; it should avoid storing unnecessary sensitive content while preserving enough information for audit, incident analysis and accountability.

Fifth, calibration should be continuous. The initial ARE weights and thresholds should be treated as governance priors, not permanent truths. They should be updated using incident reports, false-positive reviews, red-team findings, user feedback, regulatory changes and observed model behavior. This adaptive calibration is particularly important because agentic AI systems and their threat environment evolve rapidly.

### *Limitations and future research*

The study has limitations. First, the evaluation is simulation-based and therefore cannot prove that the architecture will reduce incidents in a real organization. The simulation is useful for design validation and trade-off analysis but should be followed by field testing. Second, the ARE weights are justified conceptually rather than learned from historical incident data. Future studies could use incident records, red-team traces or expert elicitation to calibrate weights empirically. Third, the model focuses on enterprise governance and may require adaptation for embedded systems, safety-critical industrial control, military systems or clinical AI. Fourth, the architecture assumes that policy rules can be made explicit enough for runtime enforcement, which may be difficult in ambiguous human contexts.

Future research should evaluate RAA in live enterprise pilots, compare different scoring functions, test adversarial prompt and tool-misuse scenarios, measure reviewer burden over time and examine how evidence ledgers interact with privacy obligations. Another important direction is multi-agent governance. When several agents delegate tasks to one another, risk may emerge from inter-agent coordination rather than from any single action. Runtime assurance must therefore evolve from single-agent tool gating to multi-agent transaction governance.

## 4. Conclusion

Agentic AI changes the enterprise risk problem because the system does not merely generate content; it may plan, retrieve, decide and act through tools. This paper proposed a runtime assurance architecture for controlling enterprise agentic AI at the point of action. The architecture separates agent planning from execution authority and applies a policy gate, an Autonomy-Risk Exposure score, least-privilege tool access, sandboxing, human approval, blocking and evidence logging.

The simulation results suggest that the proposed architecture can reduce autonomy-risk exposure while preserving a high level of task completion. In the simulated environment, the full RAA reduced mean ARE by 31.5%, suppressed invalid or policy-conflicting actions by 53.6%, eliminated unsupervised pass-through of high-risk invalid actions, and improved audit evidence coverage from 0.61 to 0.89 with a mean latency overhead of 95 ms. These results support the feasibility of policy-gated runtime control as a practical complement to AI governance frameworks, security standards and responsible AI principles.

The central implication is that enterprise agentic AI should not be governed only by model evaluation, user training or policy documents. It requires an execution architecture that makes every consequential action inspectable, scoped, interruptible and accountable. Runtime assurance is therefore not an optional add-on; it is a necessary control layer for organizations seeking to capture AI productivity while preserving human authority, regulatory defensibility and operational resilience.

## Compliance with ethical standards

### *Acknowledgments*

The author acknowledges the broader research communities working on responsible AI, AI risk management, LLM security and enterprise AI governance. No institutional funding, paid data source or human participant contribution was used for this manuscript.

### *Disclosure of conflict of interest*

The author declares that there is no conflict of interest.

### *Statement of ethical approval*

This technical research used design-science modelling and scenario simulation only. It did not involve human participants, animals, clinical samples, patient records or proprietary organizational records. Formal ethics approval was therefore not required.

### *Statement of informed consent*

Not applicable because the study did not involve human participants.

### *Funding*

This research received no external funding.

### *Data availability*

No external dataset was used. The simulation variables, equations and aggregate results are reported in the manuscript to support reproducibility of the design logic.

### *Author contributions*

Kwan Hong TAN is the sole author and is responsible for conceptualization, methodology, formal analysis, writing, review and final approval of the manuscript.

## References

[1] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). Gaithersburg: NIST; 2023. doi:10.6028/NIST.AI.100-1.

[2] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. Gaithersburg: NIST; 2024. doi:10.6028/NIST.AI.600-1.

[3] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence. Official Journal of the European Union; 2024.

[4] International Organization for Standardization. ISO/IEC 42001:2023 Information technology - Artificial intelligence - Management system. Geneva: ISO; 2023.

[5] OWASP Foundation. OWASP Top 10 for Large Language Model Applications 2025 [Internet]. OWASP; 2025 [cited 2026 Jul 6]. Available from: https://owasp.org/www-project-top-10-for-large-language-model-applications/

[6] OWASP Gen AI Security Project. Agentic AI - threats and mitigations [Internet]. OWASP; 2026 [cited 2026 Jul 6]. Available from: https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

[7] Raza S, Sapkota R, Karkee M, Emmanouilidis C. TRiSM for Agentic AI: A review of trust, risk, and security management in LLM-based agentic multi-agent systems. arXiv preprint arXiv:2506.04133. 2025.

[8] Gan Y, Yang Y, Ma Z, He P, Zeng R, Wang Y, et al. Navigating the risks: A survey of security, privacy, and ethics threats in LLM-based agents. arXiv preprint arXiv:2411.09523. 2024.

[9] He F, Zhu T, Ye D, Liu B, Zhou W, Yu PS. The emerged security and privacy of LLM agent: A survey with case studies. arXiv preprint arXiv:2407.19354. 2024.

[10] Amodei D, Olah C, Steinhardt J, Christiano P, Schulman J, Mané D. Concrete problems in AI safety. arXiv preprint arXiv:1606.06565. 2016.

[11] Weidinger L, Mellor J, Rauh M, Griffin C, Uesato J, Huang PS, et al. Ethical and social risks of harm from language models. arXiv preprint arXiv:2112.04359. 2021.

[12] Shneiderman B. Human-centered AI. Oxford: Oxford University Press; 2022.

[13] Russell S. Human compatible: Artificial intelligence and the problem of control. New York: Viking; 2019.

[14] Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, et al. Model cards for model reporting. Proceedings of the Conference on Fairness, Accountability, and Transparency; 2019. p. 220-229.

[15] Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, Daumé H III, et al. Datasheets for datasets. Communications of the ACM. 2021;64(12):86-92.

[16] Tan KH. Artificial Intelligence as Stakeholder: A Novel Framework for Ethical Recognition in Value-Creation Ecosystems [Internet]. ResearchGate; 2025 [cited 2026 Jul 6]. Available from: https://www.researchgate.net/publication/398103183_Artificial_Intelligence_as_Stakeholder_A_Novel_Framework_for_Ethical_Recognition_in_Value-Creation_Ecosystems

[17] Tan KH. Dynamic Equilibrium Theory for Ethical AI: Balancing Epistemic Uncertainty, Human Autonomy and Social Equity in High-Stakes Fluctuational Decision Systems [Internet]. ResearchGate; 2025 [cited 2026 Jul 6]. Available from: https://www.researchgate.net/publication/396616876_Dynamic_Equilibrium_Theory_for_Ethical_AI_Balancing_Epistemic_Uncertainty_Human_Autonomy_and_Social_Equity_in_High-Stakes_Fluctuational_Decision_Systems

[18] Tan KH. The Digital Productivity Paradox: A Novel Framework for Understanding the Asymmetric Distribution of Technological Benefits [Internet]. ResearchGate; 2025 [cited 2026 Jul 6]. Available from: https://www.researchgate.net/publication/399079534_The_Digital_Productivity_Paradox_A_Novel_Framework_for_Understanding_the_Asymmetric_Distribution_of_Technological_Benefits

[19] Tan KH. The Paradox of Technological Displacement: A Novel Framework for Understanding AI and Automation's Impact on Human Capital Development and Labor Market Transformation [Internet]. ResearchGate; 2025 [cited 2026 Jul 6]. Available from: https://www.researchgate.net/publication/392940107_The_Paradox_of_Technological_Displacement_A_Novel_Framework_for_Understanding_AI_and_Automation%27s_Impact_on_Human_Capital_Development_and_Labor_Market_Transformation

[20] Tan KH. Beyond automation: Sensemaking, ontological insecurity, and the emergence of the AI-form organisation in the digital era. British Research Review. 2026;1(1). doi:10.67177/gywqbn07.

[21] Raji ID, Smart A, White RN, Mitchell M, Gebru T, Hutchinson B, et al. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; 2020. p. 33-44.

[22] Leveson N. Engineering a safer world: Systems thinking applied to safety. Cambridge: MIT Press; 2011.

[23] Sha L. Using simplicity to control complexity. IEEE Software. 2001;18(4):20-28.

[24] Schierman JD, DeVore MD, Richards N, Gandhi N, Cooper S, Horneman KR. Runtime assurance framework development for highly adaptive flight control systems. AIAA Guidance, Navigation, and Control Conference; 2015.

[25] Koopman P, Wagner M. Challenges in autonomous vehicle testing and validation. SAE International Journal of Transportation Safety. 2016;4(1):15-24.