Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 14, Issue. 11, November 2025, pg.162 – 167

Trustworthy Agentic AI: Balancing Autonomy with Human Oversight

Saurabh Atri

satri@ieee.org, srbwin@gmail.com

DOI: https://doi.org/10.47760/ijcsmc.2025.v14i11.013

Abstract: Agentic AI systems increasingly plan, decide, and execute actions across tools and data sources with minimal supervision. To be deployable in safety- and compliance-sensitive environments, autonomy must be bounded by transparent decision making, predictable behavior, and recoverable failure modes. This paper proposes a practical pattern for trustworthy agentic AI that integrates a policy gate, typed tools and effects, human-in-the-loop (HITL) controls, layered safety monitors, and rigorous verification via evidence logs. We align these choices with the NIST AI Risk Management Framework and its Generative AI Profile, ISO/IEC 42001 for AI management systems, and the staged implementation of the EU AI Act [1–6].

Keywords: Agentic AI, Trustworthy AI, Human-in-the-loop (HITL), Policy gate, Red teaming, Policy-as-code, auditability, Typed tools / typed effects, Prompt injection, Compliance engineering, Risk management, Transparency & explainability, Predictability & recoverability

1. Introduction and Motivation

Agentic AI promises faster workflows and richer automation, but it also compounds risk: agents chain tools, interpret ambiguous or adversarial inputs, and may operate continuously. A trustworthy approach therefore demands an explicit framing of what 'trust' means in operations and how it is engineered into the system. The NIST AI RMF frames trustworthiness as a set of measurable characteristics and lifecycle practices (govern, map, measure, manage), with the Generative AI Profile offering cross-sector guidance tailored to GenAI risks [1–2]. Where governance must be systematic, ISO/IEC 42001 specifies requirements for an AI management system (AIMS) so organizations can establish, implement, maintain, and continually improve AI processes [3]. Meanwhile, the EU AI Act introduces phased obligations for general-purpose and high-risk AI; awareness of key dates and scope is crucial to avoid late-stage compliance scrambles [4–6].

2. Trust Goals and Threat Model

Concretely declaring trust goals comprising of transparency, predictability, recoverability, alongside a living threat model helps convert abstract principles into testable requirements that governance and audits can verify. Teams should maintain a small, versioned artifact (YAML/JSON) enumerating goals, threats (e.g., prompt injection, tool misuse, PII exfiltration, specification drift), mitigations, and review cadences aligned to organizational policy. This artifact becomes the seed for continuous hardening and aligns the team on what success and failure look like across the AI lifecycle [1–3].

3. Reference Architecture and the Oversight Loop

A repeatable oversight loop replaces ad-hoc 'model + prompt + tools' with an accountable flow:

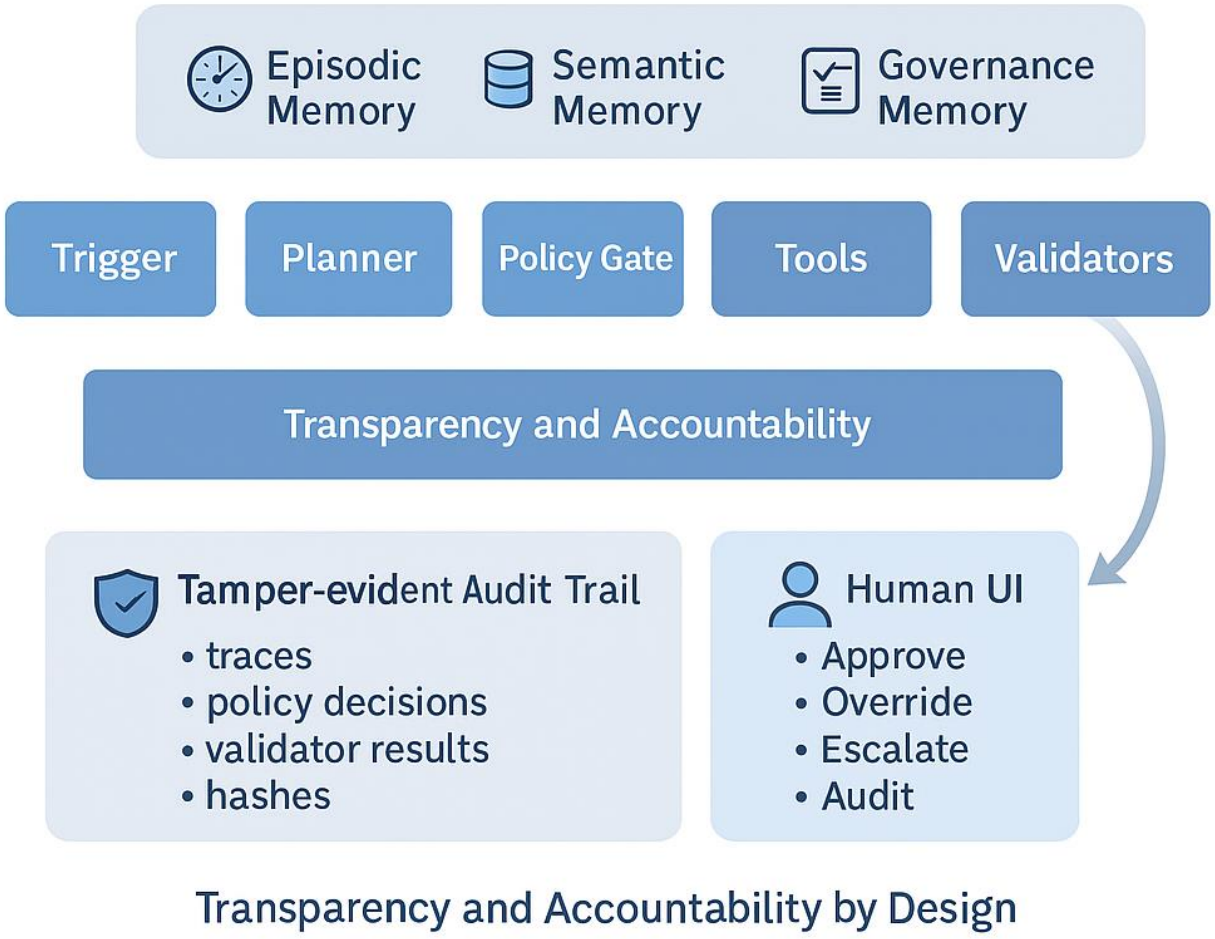

Fig 1. Design document – Transparency and accountability

Trigger → Planner → Policy Gate → Tools → Validators → Outcome, persisted in an evidence log. Before execution, the policy gate evaluates proposed actions against organizational rules; during and after execution, validators and runtime monitors check outputs and effects. The evidence log preserves prompts, tool I/O, decisions, approvals, and hashes to support audits and post-incident analysis. This pattern operationalizes the AI RMF's 'measure' and 'manage' functions by making risk controls observable and testable [1–2].

4. Interpretable-by-Construction (Not Just Post-hoc)

Interpretable-by-construction means the agent's behavior is explainable because the system requires explicit plans (steps + citations), typed schemas for arguments and effects, and uncertainty exposure that triggers human review. In high-risk operations, combining typed inputs with declared postconditions enables deterministic validation before any external system is touched supporting RMF-aligned measurement and governance outcomes [1–2].

5. Policy Gates and Typed Effects

The policy gate adjudicates proposed actions (Allow / Deny / Defer to HITL) using intent, structured arguments, and declared effects. Policy-as-code approaches like Open Policy Agent's Rego language make decisions testable and explainable; Rego is a declarative language inspired by Datalog for expressing rules over JSON-like data [7–8]. Typed effects bound side effects even when plans are permitted - for example, a 'refund' tool that hard-limits maximum liability, or an 'attachRole' tool that can only assign time-bound least-privilege roles.

6. Human-in-the-Loop Control Patterns

Human review should be risk-based, not universal: thresholds on predicted risk, uncertainty, and explicit policies route edge cases to approvers, while low-risk intents execute automatically under logging. To reduce reviewer fatigue, batch similar approvals, show diffs (requested vs. proposed scope), and surface the evidence that informed the plan. A one-flag kill switch should drop autonomy back to shadow mode if monitors detect anomalies or near-miss volume spikes [1–2].

7. Layered Safety and Runtime Monitoring

Trust cannot rely on a single control. Combine pre-execution constraints (tool allowlists, parameter ranges, rate limits), policy evaluation before execution, runtime monitors (toxicity/PII detectors, tool-output validators), and governance controls (periodic reviews, incident post-mortems, promotion gates). These layers align with management-system thinking in ISO/IEC 42001 and with the RMF's call to integrate risk controls throughout the AI lifecycle [1–3].

8. Verification, Regression, and Evidence Packs

As prompts, models, and tools evolve, regressions are inevitable. Treat agents like software-plus-policy: maintain a regression suite that exercises representative intents and adversarial cases; couple unit tests with property-based tests at the boundaries (e.g., refund amounts always within an allowed interval). Property-based testing frameworks like Hypothesis generate diverse inputs including edge cases to validate invariants and surface corner cases that example-based tests may miss [9–10]. Evidence packs bundle traces, policy decisions, validator results, and cryptographic hashes, creating auditable artifacts for internal reviews and external assessments [1–3].

9. Case Study: Access Request Agent

A typical internal access flow: a user requests access to a sensitive bucket; the planner proposes identity checks and policy evaluation; the policy gate denies 'admin' by default and suggests a time-bound read-only role unless a manager approves; tools such as IAM.getUser and IAM.attachRole execute under typed bounds; validators compare granted vs. requested scope and enforce PII redaction in notifications; and the evidence log records approvals and hashes for audit. This pattern encodes least-privilege and accountability while preserving rapid response, aligning with risk-management principles in NIST's RMF and with organizational governance under ISO/IEC 42001 [1–3].

10. Evaluation, Red Teaming, and Promotion Gates

Before expanding autonomy, run shadow mode with production-like traffic, track near-misses, and score performance along capability, safety, and UX dimensions with explicit thresholds. Jailbreak, prompt-injection, and exfiltration suites should be continuous, not one-off. Promotion gates should require clean regressions, acceptable risk metrics, and stable reviewer latency; otherwise, autonomy remains restricted to low-risk intents with HITL in the loop [1–2].

11. A high level Rollout Plan

1. Scoping & policy: define 10–20 intents, write initial policy rules, and stand up the evidence log.
2. Validators & tests: add PII/scope-drift/cost monitors; implement a regression suite and red-team scenarios.

3. Shadowing & UX: shadow hundreds of real requests; hold near-miss reviews; finalize the approval UI and escalation paths.
4. Controlled autonomy: graduate a narrow slice of intents to 5–10% auto-approve under monitors, with a one-flag kill switch and daily scorecards [1–2].

12. Standards Alignment and Regulatory Awareness

The controls above align with recognized guidance: the NIST AI RMF and its Generative AI Profile (risk functions, trustworthiness characteristics), ISO/IEC 42001 (AIMS requirements), and the EU AI Act's risk-based obligations and staged timelines for general-purpose and high-risk AI systems [1–6]. Designing for auditability (evidence logs), explainability (plans and typed effects), and human oversight (risk-based routing) lays the groundwork for compliance without sacrificing iteration speed.

Conclusion

Trustworthy agentic AI is achievable today by baking accountability into the control loop: plan visibly, gate actions with policy-as-code, constrain tools with typed effects, route risk to humans, monitor continuously, and preserve evidence. Aligning these practices with NIST's RMF, ISO/IEC 42001, and the EU AI Act's timeline reduces incident risk, accelerates stakeholder acceptance, and makes autonomy a reversible, auditable choice rather than a leap of faith [1–6].

References

[1]. National Institute of Standards and Technology (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf
[2]. National Institute of Standards and Technology (2024). AI RMF: Generative Artificial Intelligence Profile (NIST.AI.600-1). https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf
[3]. ISO/IEC 42001:2023. Artificial intelligence — Management system (AIMS) requirements. https://www.iso.org/standard/42001
[4]. Associated Press (2024). Europe's world‑first AI rules get final approval; what happens next. https://apnews.com/article/155157e2be2e42d0f1acca33983d8c82
[5]. Reuters (2025). EU to proceed on schedule with AI Act implementation. https://www.reuters.com/world/europe/artificial-intelligence-rules-go-ahead-no-pause-eu-commission-says-2025-07-04/
[6]. Le Monde (2025). First measures of the European AI Act regulation take effect. https://www.lemonde.fr/en/pixels/article/2025/02/02/artificial-intelligence-the-first-measures-of-the-european-ai-act-regulation-take-effect_6737691_13.html
[7]. Open Policy Agent. Policy Language (Rego) documentation. https://openpolicyagent.org/docs/policy-language
[8]. Open Policy Agent. Policy Reference (Rego syntax). https://openpolicyagent.org/docs/policy-reference
[9]. Hypothesis. Property-based testing for Python (documentation). https://hypothesis.readthedocs.io/
[10]. Hypothesis. What is property-based testing? https://hypothesis.works/articles/what-is-property-based-testing/

