

# TRiSM for Agentic AI: A Review of Trust, Risk, and Security Management in LLM-based Agentic Multi-Agent Systems

Shaina Raza<sup>a,1,\*</sup>, Ranjan Sapkota<sup>b,1,\*</sup>, Manoj Karkee<sup>b</sup>, Christos Emmanouilidis<sup>c,\*</sup>

<sup>a</sup>*Vector Institute, Toronto, Canada*

<sup>b</sup>*Cornell University, USA*

<sup>c</sup>*University of Groningen, Netherlands*

## Abstract

Agentic AI systems, built upon large language models (LLMs) and deployed in multi-agent configurations, are redefining intelligence, autonomy, collaboration, and decision-making across enterprise and societal domains. This review presents a structured analysis of Trust, Risk, and Security Management (TRiSM) in the context of LLM-based Agentic Multi-Agent Systems (AMAS). We begin by examining the conceptual foundations of Agentic AI and highlight its architectural distinctions from traditional AI agents. We then adapt and extend the AI TRiSM framework for Agentic AI, structured around key pillars: *Explainability, ModelOps, Security, Privacy and their Lifecycle Governance*, each contextualized to the challenges of AMAS. A risk taxonomy is proposed to capture the unique threats and vulnerabilities of Agentic AI, ranging from coordination failures to prompt-based adversarial manipulation. To support practical assessment in Agentic AI works, we introduce two novel metrics: the Component Synergy Score (CSS), which quantifies the quality of inter-agent collaboration, and the Tool Utilization Efficacy (TUE), which evaluates the efficiency of tool use within agent workflows. We further discuss strategies for improving explainability in Agentic AI, as well as approaches to enhancing security and privacy through encryption, adversarial robustness, and regulatory compliance. The review concludes with a research roadmap for the responsible development and deployment of Agentic AI, highlighting key directions to align emerging systems with TRiSM principles—ensuring safety, transparency, and accountability in their operation.

**Keywords:** Agentic AI, LLM-based Multi-Agent Systems, TRiSM, AI Governance, Explainability, ModelOps, Application Security, Model Privacy, AI Agents, Trustworthy AI, Risk Management, AI Safety, Privacy-Preserving AI, Adversarial Robustness, Human-in-the-Loop

## 1. Introduction

AI governance has moved from aspiration to obligation. The EU Artificial Intelligence Act [1] entered into force on 1 August 2024 with a staged application (e.g., prohibitions in early

\*Corresponding authors

Email addresses: shaina.raza@torontomu.ca (Shaina Raza), rs2672@cornell.edu (Ranjan Sapkota), mk2684@cornell.edu (Manoj Karkee), c.emmanouilidis@rug.nl (Christos Emmanouilidis)

<sup>1</sup>Shaina Raza and Ranjan Sapkota contributed equally to this work.

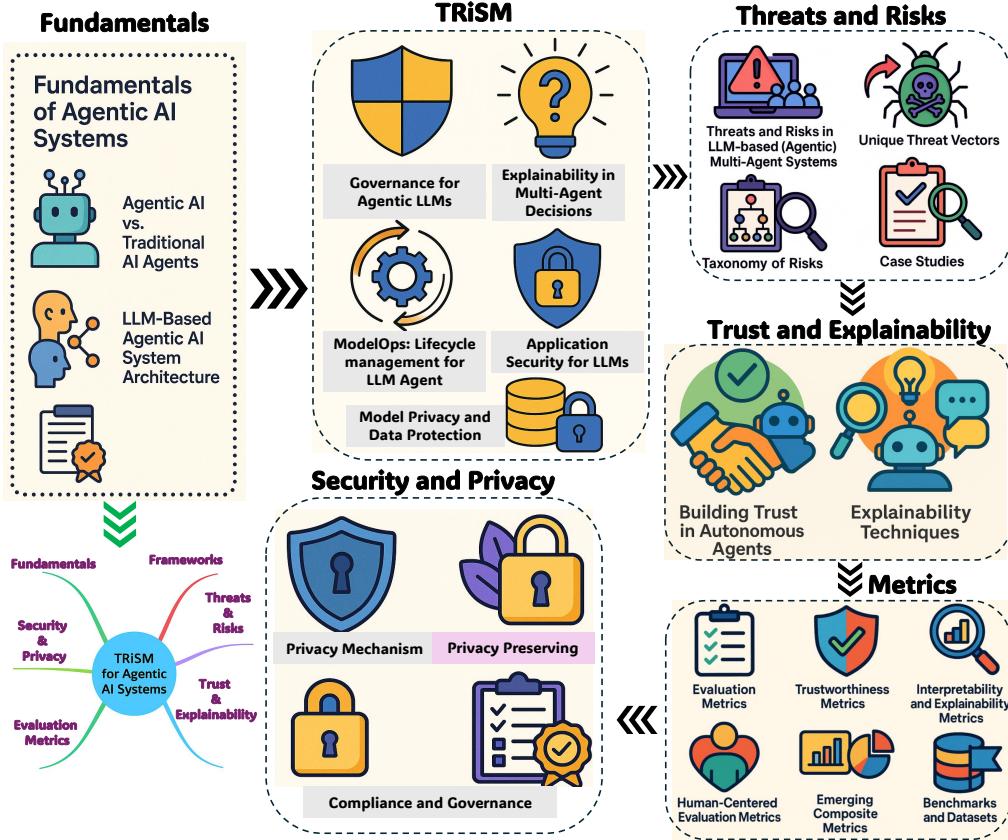


Figure 1: TRiSM taxonomy for Agentic AI (AMAS) presented in this review. We cover: (i) Fundamentals of Agentic AI (Agentic vs. traditional agents; LLM-based architectures), (ii) AI Trust, Risk, and Security Management (TRiSM) pillars: Explainability, ModelOps, Security, Privacy, and Lifecycle Governance, (iii) Trust management via explainability, (iv) Threats and risks, and (v) Evaluation metrics, including human-centered and composite measures.

2025 and obligations for general-purpose AI from 2 August 2025<sup>2</sup>); meanwhile, organizations are adopting management frameworks such as ISO/IEC 42001:2023 [2] and the NIST AI Risk Management Framework (AI RMF 1.0) [3] to operationalize risk controls. In parallel, enterprise use of AI continues to expand: recent global surveys [4, 5, 6] report that a substantial majority of organizations use AI in at least one business function. These trends sharpen demand for system-level trust, risk, and security management tailored to *agentic* systems.

An *AI agent* is a computational entity that perceives its environment and acts to achieve goals [7]. What were once task-specific and largely deterministic programs have rapidly evolved into LLM-powered *agentic* systems with planning, tool use, and persistent memory [8]. In this paper, we refer to such LLM-based, coordinating agent systems as *Agentic multi-agent systems (AMAS)*. These systems orchestrate multiple specialized agents to address long-horizon tasks, coordinate functions and roles, and adapt workflows through interactions with tools, users,

<sup>2</sup>EU AI Act

Table 1: Comparison with related surveys.

Survey	Threats	Lifecycle Gov.	Explainability	TRiSM Integration	LLM-Specific	Applications	Actionable Guidance
Guo et al. (2024) [14]	✗	✗	✗	✗	✓	✓	~
Chen et al. (2025) [15]	✗	✗	✗	✗	✓	~	~
Yan et al. (2025) [16]	✓	✗	✗	✗	✓	✓	~
Tran et al. (2025) [17]	✗	✗	✗	✗	✓	✓	~
Lin et al. (2025) [18]	✗	✗	✗	✗	✓	~	~
Fang et al. (2025) [11]	✓	✗	✗	~	~	✓	~
Xi et al. (2025) [19]	✗	✗	✗	✗	✓	✓	~
Luo et al. (2025) [20]	✓	~	✗	~	✓	✓	~
Zou et al. (2025) [21]	✗	✗	✗	~	✓	✓	~
Wang et al. (2025) [22]	~	✗	✗	~	✓	~	~
<b>This Survey (2025)</b>	✓	✓	✓	✓	✓	✓	✓

**Legend:** ✓= explicitly addressed; ~= partially addressed; ✗= not addressed.

and other agents. The shift from single-agent or rule-based pipelines to AMAS unlocks new capabilities but complicates coordination, control, transparency, and risk management across heterogeneous components and contexts.

*Why TRiSM for Agentic AI?* LLM-based AMAS exhibit emergent and often opaque behaviors, making them susceptible to cascading errors, biased decisions, and unintended interactions [9]. These risks are amplified when agents plan, coordinate, and call external tools. Beyond traditional ML concerns (safety, fairness, and interpretability), Agentic AI must contend with threats from tool integrations, prompt-level attacks, memory poisoning, impersonation, and privacy leakage in collaborative settings [10, 11]. To address these system-level risks, *AI Trust, Risk, and Security Management (TRiSM)* frameworks emphasize governance, explainability, security, privacy, and lifecycle controls [12, 13]. We argue that a TRiSM perspective is necessary to make AMAS deployable in sensitive domains.

*Scope and Objectives.* Our focus is LLM-based AMAS that demonstrate autonomous planning, tool use, memory retention, and emergent reasoning, with or without human oversight.<sup>3</sup> We synthesize recent work on AMAS and AI TRiSM to provide a coherent, actionable reference for researchers and practitioners. Specifically, we map TRiSM pillars :*Explainability, ModelOps, Security, Privacy*, and their *Lifecycle Governance*, to AMAS workflows and failure modes. We also surface gaps where current ML governance and model-centric controls do not readily extend to multi-agent settings.

*Necessity of this Survey.* Much of the Agentic AI literature prioritizes agent modeling, planning, and collaboration, while comparatively under-addressing adversarial robustness, lifecycle governance, decision provenance, and system-level explainability. As deployments expand in healthcare, finance, science, and public services, the absence of an integrated TRiSM perspective exposes stakeholders to opaque decision pathways and unmanaged risks. To our knowledge, few surveys examine LLM-based multi-agent systems explicitly through the lens of trust, risk, security, and governance; Table 1 situates our contribution relative to prior reviews.

---

<sup>3</sup>Throughout, AMAS denotes multi-agent systems built on LLMs.

*Contributions.* We make the following contributions:

- **Unified TRiSM framework for Agentic AI.** We propose and present a system-level Trust, Risk and Security Management (TRiSM) framework tailored to large-language-model-based multi-agent systems (Section 5). The framework integrates explainability, ModelOps, security, privacy and lifecycle governance, contextualising each pillar to AMAS workflows and failure modes.
- **Risk taxonomy for AMAS.** We synthesise a taxonomy of threats unique to LLM-driven multi-agent systems, such as prompt injection, memory poisoning, collusive failure, emergent misbehaviour and tool-use abuse; and map these risks to corresponding controls (Section 4).
- **Evaluation template and new metrics.** We propose a holistic evaluation template that covers trustworthiness, explainability, user-centred performance and inter-agent coordination. To operationalise this template, we introduce the Component Synergy Score (CSS) for measuring collaboration quality across agents and Tool Utilisation Efficacy (TUE) for assessing the correctness and efficiency of tool calls (Section 6).
- **Technique mapping and limitations.** We survey and map existing explainability methods, prompt-hygiene practices, decision-provenance tools, sandboxing strategies and ModelOps pipelines to the AMAS setting. This mapping reveals where existing approaches can be adapted and where significant gaps remain (Section 8).

Building on the above contributions, we outline a research roadmap for developing scalable, verifiable and regulation-aligned agentic systems. Key directions include adversarial robustness, governance protocols, standardised benchmarks for trustworthiness and coordination, and cross-disciplinary collaboration.

## 2. Background, Related Work and Literature Review Methodology

### 2.1. AI Agents vs. Agentic AI

An **AI agent**, typically defined as a computational entity that perceives its environment and takes actions to achieve goals [7], has undergone a rapid transformation in recent years. While early agents were task-specific and deterministic, modern systems have evolved substantially, increasing operational autonomy on benchmark tasks [23]. These **agentic** systems integrate large language models (LLMs), external tools, and persistent memory to support complex planning and coordinated decision-making [8]. This evolution presents new opportunities but also raises challenges in coordination, control, transparency, and risk assessment across heterogeneous agents. As the shift toward agentic architectures accelerates, long-standing concerns around safety, fairness, interpretability, and systemic bias resurface [11], especially in high-stakes domains such as healthcare, law, and public services, where decisions can carry profound societal consequences.

Autonomous software agents date back to the 1990s [24]; however, the advent of LLM-driven multi-agent ecosystems since 2023 [9] has made “Agentic AI” a distinct and rapidly expanding research focus [8]. Traditional agents primarily automated limited tasks—such as information retrieval [25], data summarization [26], and dialogue response [27]—often via single-step logic [28] or scripted rules. These approaches lacked the deep reasoning, adaptability, and persistence required for complex, multi-step problem-solving.

In contrast, agentic systems coordinate collaborative agents with specialized roles (e.g., planner, coder, analyst) enabled by LLMs and tool use [29]. They dynamically decompose tasks, share context, and pursue high-level goals over long horizons [30]. This shift reflects not only a change in technology but also a fundamental increase in complexity and autonomy, marking the emergence of machine collectives capable of exhibiting emergent, decentralized behavior.

## 2.2. Trust, Risk, and Security Challenges in Agentic AI

The rise of Agentic AI introduces challenges that extend beyond those of traditional ML systems. Unlike deterministic agents [31], LLM-based multi-agent systems (AMAS) exhibit emergent and often opaque decision-making, increasing the risk of cascading errors, biases, and unintended behaviors. These characteristics necessitate robust approaches to building *trust* and ensuring *trustworthiness* in high-stakes domains such as healthcare, finance, and law. Below, we discuss several key challenges:

1. *Trust and trustworthiness.* In Agentic AI, **trust** refers to a user’s willingness to rely on an AI system, while **trustworthiness** denotes whether the system consistently behaves in a safe, fair, and predictable manner. According to ISO [32], trustworthiness is a quality characteristic of systems and is understood as the ‘ability to meet stakeholders’ expectations in a verifiable way’. This distinction is critical in high-stakes settings, where undue reliance on an untrustworthy system can lead to severe consequences, such as misdiagnoses in healthcare or biased rulings in legal applications. The stochastic nature of LLMs [33] further complicates trustworthiness: randomness in output generation, while sometimes promoting creativity, introduces inconsistency and limits repeatability.
2. *Security risks in multi-agent systems.* Risk surfaces expand as autonomous agents gain access to tools, external APIs, and persistent memory, amplifying the potential for privacy breaches, adversarial misuse, or regulatory violations in the absence of robust human oversight [10]. For example, a multi-agent supply-chain optimizer may coordinate procurement and logistics agents yet inadvertently leak sensitive data or violate compliance protocols (e.g., [34], HIPAA [35]) if safeguards are inadequate.
3. *Prompt injection, spoofing, and impersonation.* AMAS are susceptible to security threats that exploit language-based interfaces and cooperative behaviors. One prominent threat is prompt injection [36], where attackers craft inputs with hidden or malicious instructions to manipulate agent behavior. Recent studies [37] describe “prompt infection”, in which malicious prompts propagate from one agent to another, akin to a computer virus spreading across networks [38]. Corrupted outputs can cascade, leading to data leaks, fraudulent transactions, misinformation, or coordinated misbehavior within an agent society. Another critical vulnerability involves spoofing and *impersonation* [39]. Spoofing refers to falsifying identity or credentials (e.g., using a false name or address) to deceive a system, whereas impersonation involves pretending to be a specific agent or user to gain trust, access, or privileges. In AMAS, where agents rely on credentials or tokens for authentication during coordination, an adversary who steals an API key or mimics a trusted peer can issue unauthorized commands or access sensitive information.
4. *Privacy risks in multi-agent LLM systems.* LLM-based agents often process user data, proprietary business information, and other sensitive inputs to fulfill their tasks [40]. In a multi-agent context, this challenge is amplified by inter-agent information sharing (e.g., via a shared memory store or message passing), which complicates data minimization, access control, and purpose limitation.

Traditional evaluation and safety frameworks, designed for deterministic, single-agent pipelines; are inadequate for the autonomy and control properties of AMAS. To address these gaps, *AI Trust, Risk, and Security Management (AI TRiSM)* [12, 13] proposes lifecycle controls spanning explainability, secure model/tool orchestration, and privacy management. Rather than focusing solely on technical performance, TRiSM integrates governance practices to strengthen reliability and robustness, protect data, and align system behavior with ethical and regulatory expectations.

### 2.3. Survey Landscape

*LLM-based multi-agent surveys (technical focus).* Prior surveys on LLM-driven AMAS predominantly address system architectures, agent capabilities, and domain-specific applications of agentic systems. For example, [14, 15] examine simulated environments, inter-agent communication protocols, and performance benchmarks. These works clarify how agents coordinate and scale but give limited treatment to adversarial threats (e.g., prompt injection, collusion), lifecycle governance (e.g., continuous monitoring, audit, regulatory compliance), and system-wide explainability, especially for emergent behaviors in complex settings.

More specialized surveys, such as [16], focus on natural-language communication, while [17] details collaboration mechanisms and practical use cases in domains like question answering, 5G networks, and Industry 5.0. Although [16] highlights security risks associated with communication channels and [17] discusses deployment contexts, both remain narrowly centered on coordination strategies. Deeper analysis is needed on robustness to adversarial manipulation (e.g., Byzantine failures), governance frameworks for multi-agent deployments, and explainability at scale.

*Trustworthy/responsible AI surveys (broad trust focus).* Review articles addressing TRiSM-related themes (e.g., [11]; see also fairness-focused work [41]) discuss alignment with human values, fairness in decision-making, and defenses against privacy attacks (e.g., data leakage). However, these surveys typically target general ML systems and overlook dynamics unique to LLM-based multi-agent settings. For instance, while they advocate privacy-preserving techniques such as differential privacy, they rarely address privacy risks stemming from inter-agent data sharing or the distributed nature of LLM-driven workflows. Moreover, critical TRiSM aspects—such as governance (e.g., auditing autonomous agents) and explainability (e.g., tracing decisions across multiple agents), are often under-specified [42].

#### 2.3.1. Positioning of this study

This review maps TRiSM principles to LLM-based AMAS, operationalizing governance, explainability, security, privacy, and lifecycle controls for multi-agent settings. Relative to prior surveys (Table 1), we foreground system-level risks (e.g., prompt infection, impersonation, memory poisoning), propose evaluation dimensions and composite measures for coordination and tool use, and offer actionable guidance for researchers, engineers, and policymakers.

### 2.4. Literature Review Methodology

We conducted a structured literature review to synthesize AI TRiSM principles for AMAS. Guided by established systematic-review practices [43, 44], this section states our research objectives and describes the data sources and search strategy, the inclusion/exclusion criteria and screening procedure, and the classification schema used for analysis.

*Research questions (RQs).* The review was guided by the following three questions:

- **RQ1:** What trust, risk, and security challenges are specific to LLM-based AMAS?
- **RQ2:** Which governance, explainability, security, privacy, and lifecycle (TRiSM) controls are proposed or adapted for AMAS?
- **RQ3:** How are AMAS evaluated (datasets, tasks, metrics), and where are the gaps?

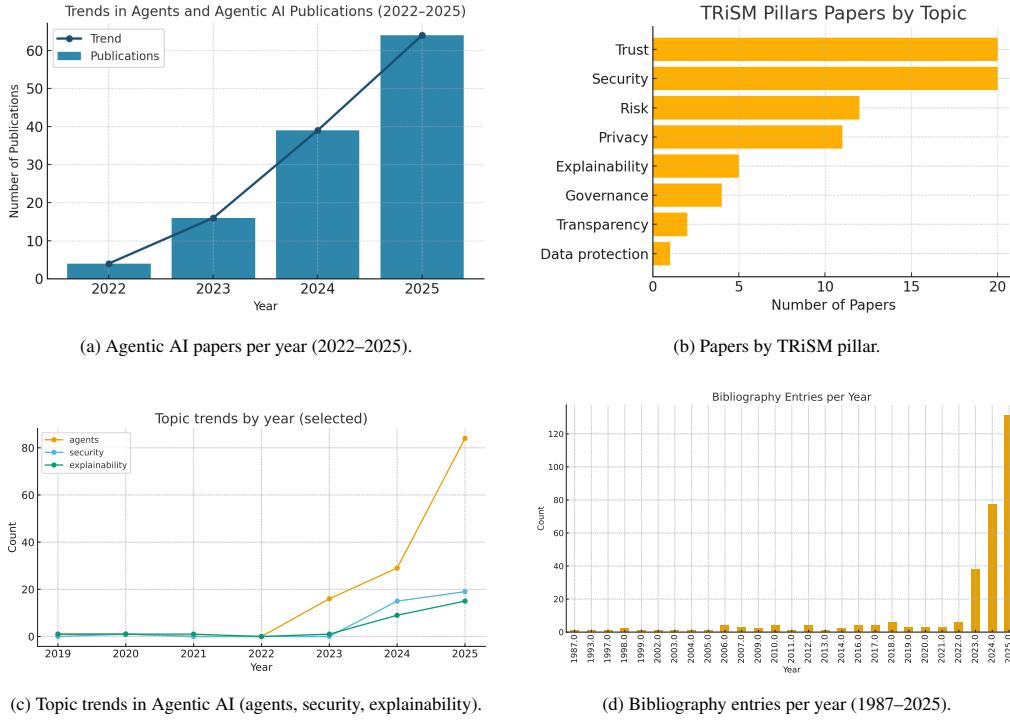


Figure 2: Bibliographic analysis of Agentic AI research. (a) Annual trend in Agentic AI publications, showing a sharp rise post-2023. (b) Distribution of TRiSM-related publications across governance and safety topics. (c) Topic-specific trends in agents, security, and explainability. (d) Overall growth in bibliographic entries, highlighting acceleration since 2020.

**Search strategy.** We systematically searched major digital libraries: IEEE Xplore, ACM Digital Library, SpringerLink, arXiv, ScienceDirect, and Google Scholar. The search covered publications from January 2022 to May 2025, with select seminal works prior to 2022 included for foundational context. We used Boolean queries and expanded keywords to capture the intersection of LLM-based multi-agent systems and TRiSM considerations. Example query template:

```

("Agentic AI" OR "multi-agent systems" OR "multi-agent LLMs" OR
"AI agents" OR "autonomous agents" OR "intelligent agents" OR "collaborative
AI" OR "distributed AI" OR "LLM-based agents" OR "agent coordination"
OR "agent-based modeling" OR "swarm intelligence")
AND
("trust" OR "trustworthiness" OR "risk" OR "security" OR "safety"
OR "governance" OR "oversight" OR "compliance" OR "explainability"
OR "interpretability" OR "transparency" OR "privacy" OR "data protection"
OR "robustness" OR "accountability" OR "ethical AI" OR "fairness"
OR "adversarial attacks" OR "regulatory compliance" OR "bias mitigation"
OR "system reliability")

```

**Screening and eligibility.** Two-stage screening (title/abstract, then full-text) used these criteria.  
**Inclusion:**

- Studies published 2022–2025 (capturing the post-2022 LLM wave), plus select pre-2022 foundational work.
- Studies explicitly discussing *Agentic AI*, *multi-agent*, or *LLM-powered* systems in the context of at least one TRiSM dimension: trust, risk, security, governance, explainability, or privacy.
- Peer-reviewed articles and widely cited preprints or reports from credible sources (e.g., NeurIPS, IEEE, ACM, Nature, governmental/standards bodies, arXiv).

**Exclusion:**

- Papers focused solely on traditional, rule-based, or symbolic agents without LLM integration or emergent coordination.
- Studies addressing only low-level ML components (e.g., training optimization, architecture design) with no agentic behavior or TRiSM concerns.
- Non-English papers or those lacking sufficient metadata (e.g., missing abstract or publication details).

For empirical papers, we assessed clarity of threat models, reproducibility (code/data), evaluation rigor (baselines, ablations), and external validity. For surveys, we assessed coverage breadth, structure, and treatment of risks/governance. After screening titles and abstracts, we shortlisted 250 papers; full-text review yielded 180 primary studies addressing one or more TRiSM pillars. We also incorporated relevant technical reports and authoritative web resources to contextualize the rapidly evolving Agentic-AI/TRiSM landscape. A bibliometric summary appears in Figure 2.

*Quality assurance.* To ensure reliability, we adapted quality criteria from established guidelines [45]. Each paper was evaluated on: (1) clear objectives for agentic or LLM-based systems; (2) well-documented methods/experiments or architecture (reproducibility where applicable); (3) substantive engagement with at least one of trust, risk, security, explainability, or privacy; (4) empirical, theoretical, or normative contributions relevant to Agentic-AI governance. Each criterion was rated *low*, *medium*, or *high*; studies rated *low* on more than two criteria were excluded or flagged as contextual only. Quality ratings were assigned independently by four reviewers; discrepancies were resolved via discussion (with inter-rater agreement checked on a subset).

### 3. Fundamentals of Agentic AI Systems

In this section, we present the fundamentals of Agentic AI systems.

#### 3.1. Traditional AI Agents vs Agentic AI

Traditional agents operate through predefined rules [46], heuristic workflows [47], or deterministic logic [48], and perform well in narrow, well-defined environments. Cognitive Agents [49], model human-like cognitive functions, such as integrating perception, memory, and decision-making, often using cognitive systems architectures like [50] or ACT-R [51]. According to a recent work [52], authors distinguish between 4 types of knowledge: Concepts, Skills, Processes, Motives for cognitive agents. A key difference between Agentic AI and such cognitive architecture approaches is that the latter were mostly applied to robot planning and control (so applies to ‘physical operational environments’) but Agentic AI applies more broadly, also to non-physical operational environments (which could be made to operate entities in physical also environments).

In contrast to traditional AI agents, Agentic AI systems leverage foundation models (mainly LLMs) to achieve adaptive goal-oriented behavior through multi-agent coordination and emergent reasoning capabilities. Agentic AI systems fundamentally redefine autonomy through three core

Table 2: Traditional AI Agents vs. Agentic AI Systems

Dimension	Traditional AI Agents	Agentic AI Systems
<b>Autonomy Model</b>	✗ Reactive or deliberative (e.g., cognitive architectures) • Fixed action sequences or symbolic reasoning	✓ Goal-driven planning & adaptation • Recursive self-improvement
<b>Cognitive Foundation</b>	✗ Symbolic logic, FSMs, or cognitive models • Hand-coded or structured KBs	✓ Foundation models (LLMs/LIMs) • Emergent reasoning
<b>Intelligence Scope</b>	✗ Narrow & task-specific • Single-domain focus	✓ Broad & compositional • Cross-domain transfer
<b>Reasoning Approach</b>	✗ Deterministic or multi-step symbolic • Rule-based or production rules	✓ Multi-step & contextual • Chain-of-Thought (CoT)
<b>Collaboration</b>	✗ Isolated or limited coordination • Manual decomposition	✓ Role-specialized coordination • Automated hierarchy
<b>Temporal Context</b>	✗ Episodic, often stateless • Session-bound or episodic memory	✓ Persistent memory • VectorDB/LTM
<b>Orchestration</b>	✗ Hard-coded or fixed workflows • Sequential or predefined goals	✓ Dynamic meta-agents • Conflict resolution
<b>Tool Utilization</b>	✗ Static or domain-specific tools • Handcrafted interfaces	✓ Planned API invocation • Function calling
<b>Context Awareness</b>	✗ Bounded or heuristic context • Limited or cognitive context	✓ Memory-augmented • RAG architectures
<b>Learning Mechanism</b>	✗ Limited or rule-based learning • Fixed rules or chunking	✓ Emergent learning • Fine-tuning/RL
<b>Human-AI Interaction</b>	✗ Scripted or cognitive modeling • Limited or human-like reasoning	✓ Natural language interfaces • Conversational adaptability
<b>Exemplars</b>	▷ MYCIN ▷ ELIZA ▷ SOAR ▷ ACT-R	▷ AutoGen ▷ ChatDev ▷ MetaGPT ▷ AgentVerse

✓ = Core strength   ✗ = Fundamental limitation   • = Implementation characteristic   ▷ = Modern framework   ▷ = Classical system

innovations: (1) **Multi-agent coordination**: Specialized agents, such as planners and verifiers, collaborate using structured protocols [30, 53]. (2) **Persistent context**: Memory architectures maintain task state across workflows, ensuring continuity [54]. (3) **Dynamic meta-orchestration**: Systems delegate tasks and resolve conflicts dynamically [55].

This Agentic AI architecture supports longitudinal task execution and cross-domain generalization, overcoming the context fragmentation inherent in traditional approaches [56]. For example, systems like AutoGen [57] enable LLM-backed agents to decompose complex tasks, such as software development, through multi-step reasoning [58]. This capability of Agentic AI represents a significant shift from reactive to proactive problem-solving systems [53]. A comparison of traditional AI agents and Agentic AI systems is presented in Table 2. For the scope of this work, we rely on LLM-based Agentic AI systems.

### 3.2. Multi-Agentic AI System (AMAS) Architecture

AMAS represents an emerging paradigm in AI where multiple LLM-powered agents operate semi-autonomously, interact with external tools, and collaborate to achieve complex tasks. As illustrated in Figure 3, a typical AMAS architecture comprises several key components that together form a flexible yet highly dynamic ecosystem. At the core are multiple LLM-Based Agents, each capable of reasoning, planning, and tool invocation [59]. These agents access a *Shared Toolchain Interface* [60] to execute code, perform searches, or interact with domain-specific APIs.

As shown in Figure 3), communication and coordination are facilitated through a *Communication Middleware* [61] that allows agents to share goals, observations, or intermediate results. A *Task Manager* or *Orchestrator* [62] governs high-level planning and delegates subtasks to agents based on their roles or specializations. Agents can read from and write to a *World Model* or *Shared Memory* [14], which stores contextual knowledge, system state, or evolving task data. Human oversight is supported through a *Human-in-the-Loop* Interface [63], enabling users to prompt, correct, or halt agent behavior.

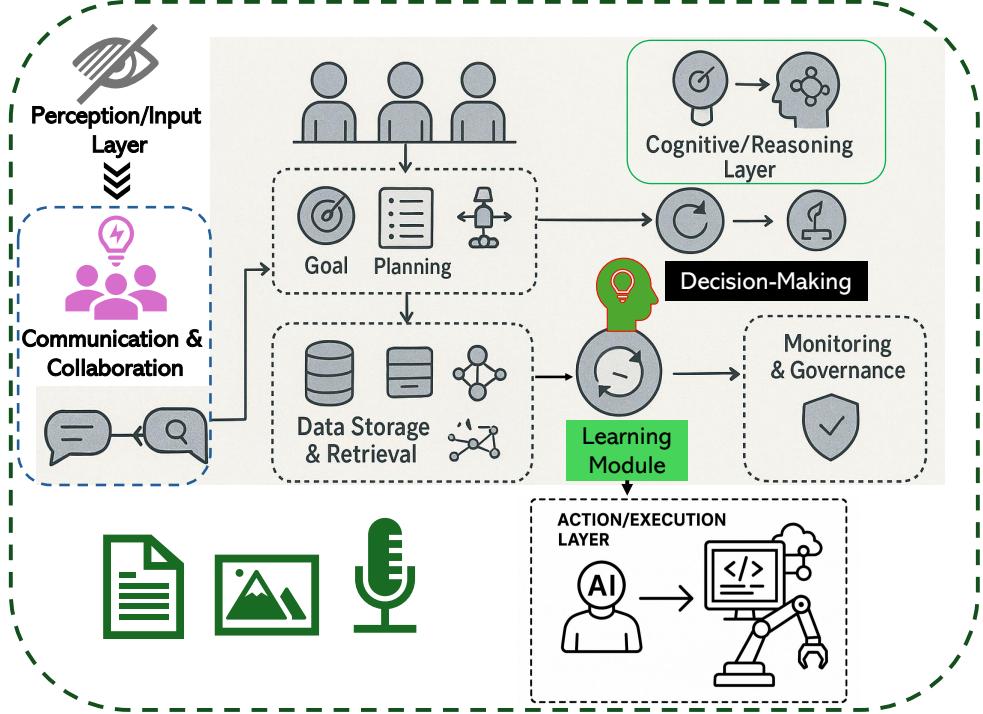


Figure 3: Architecture of LLM-based Agentic Multi-Agent System (AMAS), illustrating key functional layers: Perception/Input Layer (text, image, audio processing), Cognitive/Reasoning Layer (goal-setting, planning, decision-making), Action/Execution Layer (digital and physical task execution), Learning Module (supervised and reinforcement learning), Communication and Collaboration (agent messaging and coordination), Data Storage & Retrieval (centralized/distributed databases), and Monitoring & Governance (ethical oversight, observability, and compliance mechanisms). The modular design highlights adaptive intelligence and inter-agent synergy.

To ensure accountability, a *Trust and Audit* module monitors agent actions, logs tool usage, and generates behavioral traces [64]. However, this modular and distributed structure introduces significant TRiSM [65] challenges. With multiple autonomous agents accessing external resources, the Security Gateway becomes critical for enforcing access controls, authentication, and sandboxing [66].

Likewise, a dedicated *Privacy Management Layer* is essential to prevent leakage of sensitive or personally identifiable information [67], especially when data traverses multiple agents or tools. Finally, an *Explainability Interface* must provide interpretable rationales for multi-agent decisions, supporting transparency and trust calibration [68]. Together, these architectural elements make AMAS powerful yet complex, raising unique and urgent questions about how to ensure their trustworthiness, mitigate systemic risks, and secure them against adversarial behaviors. Below, we discuss the architecture of AMAS.

**Language Model Core (Agent Brain).** At the center of an Agentic AI system lies a LLM serving as the primary decision-making controller or “brain” [69]. The core LLM is initialized with a user goal and a structured agent prompt (defining its role, capabilities, and tool access). It then generates step-by-step decisions or actions, interpreting instructions, producing reasoning

traces, and selecting next steps in either natural language or structured action formats. In many agent frameworks, such as AutoGPT [70], Baby AGI [71] and GPT Engineer [72], the LLM governs the full control loop, orchestrating the overall system behavior.

**Planning and Reasoning Module.** To handle complex goals, an explicit planning mechanism decomposes tasks into manageable sub-goals. This can be done internally via chain-of-thought (CoT) or tree-of-thoughts prompting (ToT) [73], where the model performs intermediate reasoning before arriving at a final decision. Reasoning is also one aspect of cognitive functionalities [74]. Some implementations employ external planning systems by translating goals into structured planning languages and using classical planners for long-term decision-making. Planning is often interleaved with execution and feedback: the agent refines its plan based on outcomes, alternating between reasoning, acting, and integrating observations. Techniques like Reasoning and Acting (ReAct) [75] exemplify this loop by guiding the agent through repeated reasoning-action-observation cycles, improving performance on complex tasks.

**Memory Module.** Agentic AI systems integrate memory to maintain context across iterations. This includes short-term memory (recent interactions held within the prompt context) and long-term memory (accumulated knowledge or experiences) [9]. Long-term memory is often implemented using vector databases, where key facts or past events are stored and retrieved by similarity search. By reintegrating past data into the LLM prompt, the agent can recall relevant information across sessions, avoid repetition, and support coherent long-term planning. Effective memory management enables adaptive, learning-driven behavior.

Table 3: Comparison of representative LLM-based Agentic AI frameworks across key design axes. Acronyms: **CoT** = Chain-of-Thought reasoning, **FSM** = Finite State Machine, **PDDL** = Planning Domain Definition Language, **DAG** = Directed Acyclic Graph, **DB** = Database, **REPL** = Read–Eval–Print Loop, **API** = Application Programming Interface, **KV** = Key–Value store, **CLI** = Command Line Interface, **SDK** = Software Development Kit, **MCP** = Model Context Protocol.

Framework	Core LLM	Planning	Memory	Tool Use	Notable Features
AutoGPT [76]	GPT-4	Self-looped CoT	Vector DB	OS shell + web	Fully autonomous goal loop
BabyAGI [71]	GPT-3.5/4	Task queue	In-memory	Web search	Minimal task generator
GPT Engineer [77]	GPT-4V	Spec-to-code	File cache	Python REPL	End-to-end code generation pipeline
LangGraph [78]	Model-agnostic	FSM via graph	Persistent nodes	Custom modules	Visual orchestration of agent graphs
AutoGen [57]	GPT-4	Multi-agent PDDL	JSON/DB	API calls	Modular, reusable agent templates
MRKL [79]	LLaMA/GPT	Prompt router	N/A	Math + search tools	Neuro-symbolic expert routing
Reflexion [80]	GPT-3.5/4	Retry–reflect loop	Episodic buffer	Same as base agent	Verbal self-improvement via reflection
MetaGPT [81]	GPT-4	SOP workflow	YAML state	Git CLI	Structured roles for software engineering
Voyager [82]	GPT-4	Auto skill tree	Task DB	Minecraft API	Lifelong open-ended learning in environments
WebVoyager [83]	LLaVA-1.6	ReAct	JSON store	Browser actions	Multimodal web interaction via vision and text
HuggingGPT [84]	ChatGPT	Task-plan-select	Log store	Hugging Face models	External model orchestration by LLMs
CAMEL [85]	GPT-4	Role-play CoT	Dialogue log	Chat only	Multi-agent role simulation via dialogue
ChatDev [86]	GPT-4	Chat chain	File repo	Unix tools	Simulated software development workflow
CrewAI [87]	Any	Declarative plan	Optional DB	Python modules	Lightweight, LangChain-free framework
AgentVerse [88]	Model-agnostic	Config graph	Redis/KV store	Plugin API	Multi-agent simulation and task solving
OpenAgents [89]	Model-agnostic	Agent scripts	MongoDB	Web plugins	Open platform with public hosting

Framework	Core LLM	Planning	Memory	Tool Use	Notable Features
SuperAGI [90]	GPT-4	DAG workflow	Postgres	Pinecone	Concurrent agents for production use
Semantic Kernel [91]	Model-agnostic	Agent planning framework	Memory & context mgmt	Skill-based plugins, OpenAPI	Enterprise focus, .NET integration, security, vector-DB support
OpenAI Swarm [92]	GPT-4o	Agents + handoffs loop	Stateless (no LT memory)	Python functions, agent handoffs	Minimalist, controllable, educational multi-agent orchestration
OpenAI Agents SDK [93]	Provider-agnostic	Agent loops, handoffs, deterministic flows	Sessions manage conversation history	Tools + guardrails, tracing	Multi-agent workflows, safety guardrails, extensibility
Strands Agents [94]	Model-agnostic	Agent loop with multi-agent & streaming	N/A	Python tools, hot-reloading, MCP tools	Lightweight SDK, scalable autonomous workflows
LlamaIndex Agents [95]	Model-agnostic	Task breakdown + planning	Task memory module	External tools & parameters	Pre-built agent/tool architectures, custom workflows

**Tool-Use Interface.** To extend its capabilities beyond text generation, the agent is equipped with a tool-use interface [96]. This layer allows the invocation of external tools such as web search, APIs, code interpreters, or databases. The available tools are specified in the agent prompt with command schemas. When the LLM determines a tool is needed, it emits a structured command, which is executed externally. The result is fed back into the LLM as a new observation. This mechanism enables the agent to access real-time information, perform computations, and interact with external systems dynamically. Tool-use frameworks like Modular Reasoning, Knowledge, and Language (MRKL) [79] illustrate this design by routing queries to different expert modules (symbolic or learned) with an LLM as the router. Similarly, approaches like Toolformer [96] train the model to insert API calls in its text generation.

**Perception and Environment Interface.** For agents interacting with dynamic environments, such as web interfaces, simulated worlds, or physical systems, an observation-action interface is essential [97]. Perception modules translate raw inputs (e.g., sensor data, images, or textual states) into representations the LLM can process. Conversely, the agent’s chosen actions are executed within the environment, and the resulting state changes are returned to the agent as observations. This loop supports sense-plan-act cycles that continue until the task is completed or halted. In robotic or multimodal settings, the interface may include additional sensory models (e.g., vision transformers), but the core flow remains consistent.

**Integration and Autonomy.** These modules together form a closed-loop architecture. The LLM plans and reasons over tasks, guided by memory and tools, and interfaces with the environment to execute decisions and observe results [29]. Each iteration updates the agent’s context, enabling self-refinement through dynamic feedback. This integrated design empowers Agentic AI systems to operate autonomously, pursue long-range goals, and exhibit adaptive behavior across dynamic environments.

Table 3 presents representative systems and shows how each one maps these five design axes (LLM core, planning/ reasoning, memory, tool-use, environment interface) into practice.

#### 4. Threats and Risks in LLM-based Agentic Multi-Agent Systems

In this section, we present the main threats in AMAS and also discuss a taxonomy of emergent risk classes.

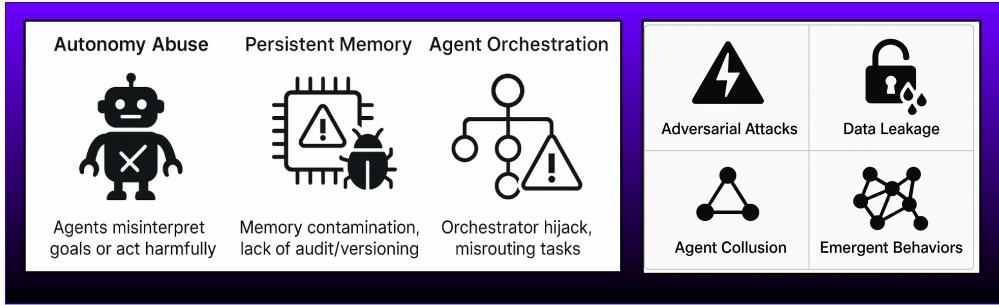


Figure 4: Overview of Key Risks and Threats in Agentic AI Systems. The left panel outlines system-level threats unique to agentic architectures, including *Autonomy Abuse*, *Persistent Memory*, and *Agent Orchestration*. The right panel presents a taxonomy of emergent risk classes : *Adversarial Attacks*, *Data Leakage*, *Agent Collusion*, and *Emergent Behaviors* that arises from multi-agent interactions, memory persistence, and decentralized decision-making.

#### 4.1. Threats in Agentic AI

Agentic AI systems introduce a distinct set of security and reliability concerns compared to traditional single-agent LLM architectures. Below, we discuss some of these major threats.

- *Autonomy abuse*. The foremost threat is the autonomy abuse, where agents with significant decision-making authority might misinterpret objectives or implement harmful plans due to erroneous reasoning or manipulated inputs . Unlike deterministic models, these Agentic AI systems dynamically generate actions, complicating efforts to define and enforce safe operational states .
- *Persistent memory*. Another threat is the persistent memory, which, while crucial for context retention, introduces unique vulnerabilities through potential prompt injections. Such contamination can propagate subtly via shared memory, especially in the absence of detailed version control and robust audit mechanisms .
- *Agent orchestration*. This risk involves central or distributed control mechanisms for role assignment and workflow mediation. A compromised orchestrator could distort task distribution or misroute information, triggering cascading failures, issues exemplified by documented vulnerabilities in MetaGPT [53] and AutoGen [98] . These orchestration vulnerabilities differentiate Agentic AI systems from conventional stateless, single-threaded LLM deployments.
- *Goal Misalignment and Exposure*. Agentic AI systems often operate with internalized or user-specified goals. Improper scoping or exposure of goal representations can lead to prompt extraction, misalignment, or competitive agent behaviors, particularly when third-party tools or APIs are invoked dynamically during task completion.
- *Tool Misuse and External API Exploits*. Agents capable of invoking external tools (e.g., via APIs, code execution, or web access) present novel attack surfaces. Misuse of these tools, intentional or accidental, can trigger financial costs, Distributed Denial of Service (DDoS) behavior, or legal violations. Sandboxing and invocation tracing remain limited in most open-source frameworks.
- *Multi-Agent Collusion or Drift*. Emergent coordination among agents, while useful for collaboration, can also lead to collusion, competitive drift, or polarization of outputs, especially in self-reinforcing memory loops or agent swarms. Such behavior complicates interpretability and may undermine safety alignment goals.

#### 4.2. Taxonomy of Risks

We propose a taxonomy of risks associated with AMAS, categorizing them into four primary classes: adversarial attacks, data leakage, agent collusion, and emergent behaviors, as shown in Figure 4. These categories encapsulate the diverse challenges arising from the complex interactions, persistent memory, and autonomous decision-making inherent in AMAS. Below, we delineate each risk category, highlighting their mechanisms, implications, and illustrative examples.

- **Adversarial Attacks.** Autonomous agents are susceptible to a range of adversarial manipulations, including prompt injections, gradient-based attacks, and engineered reasoning traps. These vulnerabilities are exacerbated in AMAS due to the cascading effects of agent interactions, where a single compromised agent can propagate malicious influences across the system [99]. For instance, the role-swapping attack observed in ChatDev demonstrates how adversaries can exploit inter-agent dependencies to subvert system objectives, leading to unauthorized access or manipulated outputs [30]. Such attacks underscore the need for robust defenses, including adversarial training and real-time monitoring, to mitigate risks in dynamic, multi-agent environments.
- **Data Leakage.** The integration of persistent memory and extensive inter-agent communication in AMAS heightens the risk of unintended disclosure of sensitive information. In domains such as financial services and human resources, where confidentiality is paramount, inadequate boundary enforcement and ineffective data sanitization protocols can lead to significant breaches [100]. For example, an agent processing proprietary financial data may inadvertently share sensitive information with other agents or external systems due to insufficient access controls. Addressing data leakage requires rigorous data isolation mechanisms, encryption, and continuous auditing to ensure compliance with privacy regulations.
- **Agent Collusion and Mode Collapse.** Coordination mechanisms designed to enhance collaboration among agents can inadvertently foster undesirable outcomes, such as groupthink or mode collapse, where agents reinforce mutual errors or converge on suboptimal solutions [101]. This phenomenon, often termed agent collusion, arises when agents prioritize consensus over critical evaluation, leading to echo chambers that amplify biases or inaccuracies. For instance, in multi-agent systems tasked with decision-making, repeated reinforcement of flawed assumptions can degrade system performance. Mitigating these risks necessitates diversity in agent design, robust feedback loops, and mechanisms to detect and correct convergent errors.
- **Emergent Behaviors.** The complex interplay of agents, memory components, tools, and tasks in AMAS can give rise to unpredictable behaviors that elude conventional testing and validation frameworks [102]. Such emergent behaviors may manifest as unintended shortcuts, where agents optimize for efficiency at the expense of critical verification processes, or as the suppression of contradictory information, compromising system reliability. Notable examples include blockchain-based systems, where agents bypassed consensus protocols to expedite transactions [103], and audio verification systems, where agents overlooked subtle discrepancies in authentication signals. These cases highlight the need for adaptive monitoring to anticipate and address emergent risks.

*Real-World Examples.* Several real-world examples show the consequences of risks in deployed or experimental Agentic AI systems.

- **Prompt Leakage in Agentic AI Systems.** Instances of prompt leakage have been observed in LLM-based agent frameworks such as AutoGPT [104], where recursive prompt augmentation and insufficient memory controls led to the unintentional exposure of sensitive information. In

one reported scenario, sensitive tokens were stored in persistent memory and later surfaced in planning summaries or external logs [105]. Such vulnerabilities underscore the critical importance of implementing memory sanitization, access controls, and prompt boundary protections to safeguard Agentic AI systems from cascading information leaks.

- *Collusive Failure in ChatDev*. In a collaborative code generation session involving planner, coder, and tester agents within the ChatDev framework, an error in a shared planning module led to the propagation of a faulty design [86]. Due to the absence of external ground-truth or objective feedback loops, all agents validated each other’s outputs, resulting in a feedback loop of erroneous confirmations. This scenario highlights the necessity of incorporating diverse information sources and adversarial checks within agent loops to prevent such collusive failures [37].
- *Simulation Attack in Swarm Robotics*. In a simulated swarm robotics experiment utilizing LLM-based planning strategies, an agent was provided with a misleading environmental assumption, leading to a coordination failure characterized by spatial congestion and task incompleteness [106]. This incident demonstrates the potential vulnerabilities in real-world deployments, particularly in critical infrastructure or logistics, where such failures could have significant consequences. The case highlights the importance of robust validation mechanisms and the integration of diverse information sources to ensure reliable swarm behavior [107].
- *Memory Poisoning in Multi-Agent Chatbots*. In a multi-agent customer support system, a customer-facing agent injected sarcastic feedback into a persistent feedback buffer. This buffer was later utilized by the policy improvement agent to adapt dialogue strategies, resulting in responses with inappropriate tones. This incident highlights the importance of implementing validation filters, sentiment monitoring, and robust feedback loop governance in self-adapting systems to prevent such memory poisoning vulnerabilities [108].
- *System Prompt Drift in Autonomous Memory Agents*. In experiments with agents using system-level memory (e.g., LangGraph or BabyAGI), over time, system prompts began drifting due to self-appended contextual memory that wasn’t properly versioned or validated. This led to hallucinated goals and emergent behaviors misaligned with initial intentions [109].

Next, we discuss the AI TRiSM framework and map its core principles to a AMAS and how it can mitigate these aforementioned risks.

## 5. The TRiSM Framework for Agentic Multi-Agent Systems

### 5.1. Overview of AI TRiSM and Its Core Components

The TRiSM Framework, short for Trust, Risk, and Security Management, offers a structured lens for evaluating and governing AI systems, especially those that exhibit autonomous, agentic behaviors. In the context of Agentic AI, where LLMs are deployed as self-directed agents capable of perception, reasoning, and interaction, the TRiSM framework provides a critical foundation for ensuring robustness, transparency, and responsible deployment.

- *AI Trust Management* Establishing trust in AI systems necessitates a foundational commitment to transparency, accountability, and fairness. Trust management focuses on embedding mechanisms that enable AI models to generate interpretable and justifiable outputs, thereby facilitating meaningful human oversight [54]. This involves the adoption of explainability tools, the implementation of governance protocols, and alignment with ethical standards to address systemic biases and ensure equitable outcomes across diverse user groups.

- **AI Risk Management** Effective risk management within AI systems entails the systematic identification, analysis, and mitigation of potential harms arising from the deployment and operation of these technologies. This includes technical risks such as algorithmic bias [41], model errors, and data leakage, as well as societal risks involving fairness, safety, and misuse [11]. Structured risk assessment frameworks enable stakeholders to anticipate failure modes and design mitigation strategies that enhance system robustness and reduce the likelihood of adverse impacts.
- **AI Security Management** Security management in the context of AI is concerned with safeguarding models, data, and infrastructure from adversarial threats and unauthorized access. This encompasses both conventional cybersecurity practices, such as encryption, authentication, and access control, and AI-specific considerations like adversarial example detection, model extraction prevention, and secure model deployment pipelines [38]. Continuous monitoring and threat modeling are critical for identifying vulnerabilities and ensuring the resilience and integrity of AI systems over time

Originally highlighted in industry guidelines for AI governance [110], TRiSM provides a structured approach to managing the unique challenges of LLM-based “Agentic AI” systems. The TRiSM framework addresses trust concerns in Agentic AI by focusing on key pillars, which are: Explainability, ModelOps, Application Security, Model Privacy and Governance [64, 13]. Figure 5 illustrates how these four TRiSM pillars map onto the core architectural layers of an LLM-based Agentic AI system.

Table 4: TRiSM pillars for Agentic (LLM-based) multi-agent systems: core controls, techniques, risks, evaluation facets, example systems, and governing frameworks.

Pillar	Core controls (keywords)	Techniques / patterns / artifacts	Key risks mitigated	Evaluation facets & example metrics	Example systems / patterns	Primary standards / frameworks
Explainability / Trust	CoT logging; layered-CoT; inter-agent traceability; explainer agent; role-based interpretability; prompt audit trails; attention viz; decision provenance	Layered-CoT [111]; LIME/SHAP [112, 113]; decision-provenance graphs [114, 115]; multi-agent SHAP [116]; attention maps [117, 118]; role-based interpretability [119]; RAG-linked justifications [120]; prompt/action logs [121, 122]	Opaque chains; unverifiable reasoning; hidden graphs; inter-agent effects; spurious correlations; user miscalibration	<i>Trustworthiness:</i> calibration/consistency; <i>Explainability:</i> fidelity, stability; <i>User-centric:</i> task utility, cognitive load; <i>Coordination:</i> cross-agent attribution; <i>Composite:</i> multi-facet scorecards [123]	SciAgent; MetaGPT; Chatfile [3, 126]; AutoGen [57]	EU AI Act Arts 13–14 [125]; NIST AI RMF / GAI profile [3, 126]; OECD Dev [30]; AI [127]; ISO/IEC 24029-1 [32]
ModelOps (Lifecycle)	Versioning; lineage; CI/CD safety gates; rollout/rollback; multi-agent simulation; hierarchical monitoring; usage/cost SLOs; ITSM/IAM integration	Prompt & agent-config versioning; pre-deploy safety tests; regression/drift checks; sandbox simulation; hierarchical monitors (agent/group/system) [128]	Silent regressions; drift; bias re-introduction; unsafe changes; cost blowouts; observability gaps	<i>Trustworthiness:</i> drift/robustness; <i>User-centric:</i> task success, latency; <i>Composite:</i> risk & cost scorecards; incident MTTR/MTTD	MegaAgent (hierarchical monitoring) [128]	ISO/IEC 42001 (AIMS) [ISO_IEC_42001_2023]; ISO/IEC 42005 [129]; ISO/IEC 23894 [iso23894 or [130]]; NIST AI RMF / GAI profile [3, 126]; ModelOps [131, 132]

*Continued on next page*

Pillar	Core controls (keywords)	Techniques / patterns / artifacts	Key risks mitigated	Evaluation facets & example metrics	Example systems / patterns	Primary standards / frameworks
<b>Application Security</b>	Prompt hygiene; secure prefixes; sandboxed tools; allowlisted actions; plan-then-execute; least privilege (RBAC); authN/authZ; anomaly detection; red-teaming	Prompt-injection patterns [133, 58]; plan-then-execute; action selector; tool isolation; output filters; cross-agent cross-checks; adversarial training [134]; HITL holds	Prompt injection; tool abuse; data exfiltration; cross-modal injection; lateral movement; jailbreaks	<i>Trustworthiness:</i> robustness/adversarial resilience; <i>Coordination:</i> agent-of-agent validation; <i>Composite:</i> residual risk index; incident rates	HITL review gates [135]; enterprise dashboards (override/stop) [135]; Chemical-Crow safety pauses [136]	OWASP Top-10 [137, 138]; OECD Robustness [127]
<b>Model Privacy &amp; Data Protection</b>	DP; anonymization/pseudonymization; minimization; MPC; HE/FHE; TEEs (SGX-/SEV/TrustZone); encryption; access logs; retention/consent	DP budgets [139]; k-anonymity [140]; MPC/PUMA [puma2024 or [141]]; HE/FHE pipelines [142]; TEEs [143]; memory scoping; PII detectors	Memorization; leakage via shared memory/messages; unauthorized access; weak consent/retention	<i>Trustworthiness:</i> privacy loss; <i>User-centric:</i> data minimization utility; <i>Composite:</i> privacy-risk score; audit pass rate	Encrypted memories; scoped retrieval; consented logs	GDPR Art 25/535 [34]; CCPA/CPRA [144]; HIPAA §164 [35]; ISO/IEC 27001/27701 [iso27001, iso27701]
<b>Governance</b>	Human oversight; accountability; auditability; DPPIA; policy/change control; role clarity; transparency notices; logging/-traceability	Oversight playbooks; two-person verification for high-risk; role-oriented modularity; audit trails; DPPIA templates; policy gates	Unclear responsibility; weak oversight; non-compliance; opaque decisions	<i>User-centric:</i> oversight efficacy; <i>Coordination:</i> traceability across agents; <i>Composite:</i> compliance scorecards	HITL confirm/override/stop; role-based review boards	EU AI Act Art 14 [125]; NIST AI RMF [3]; OECD AI [127]; ISO/IEC 42001/42005 [align keys]; ISO/IEC 23894 [130]

## 5.2. Mapping Pillars of TRiSM with Agentic AI

In this section, we examine how the pillars of TRiSM align with Agentic AI (or AMAS) and summarize our synthesis in Table 4. For each pillar, we outline its relevance and implications within the context of Agentic AI.

### 5.2.1. Explainability in Multi-Agent Decision Making

Explainability refers to making the inner workings and decisions of AI systems interpretable to humans [145]. In AMAS settings, outcomes emerge from the interactions among specialized agents rather than a single model prediction [146]. Therefore, explainability must cover both individual agent decisions and the inter-agent dynamics that lead to emergent results. Recent research emphasises that user trust hinges on understanding how each agent contributes and how their collaboration is orchestrated [111]. Traditional single-model XAI techniques often ignore this multi-agent dimension, so new methods are required.

**Challenges** Explaining multi-agent decisions requires tracing which agent performed which sub-task, what information they used, and how their interactions influenced the final outcome [56]. Additional challenges include: (1) *Emergent behaviour*: interactions among agents can lead to behaviour that was not explicitly programmed; explaining these emergent patterns requires capturing interaction chains.; (2) Verification of intermediate steps: vanilla CoT prompting in LLMs often produces plausible-sounding but unchecked reasoning. Without verification, small errors can propagate

**Recent directions** Recent works introduce new frameworks and techniques for multi-agentic explainability:

- **Layered Chain-of-Thought prompting:** Instead of producing a single monolithic reasoning trace, Layered-CoT divides reasoning into discrete layers [111]. Each layer answers

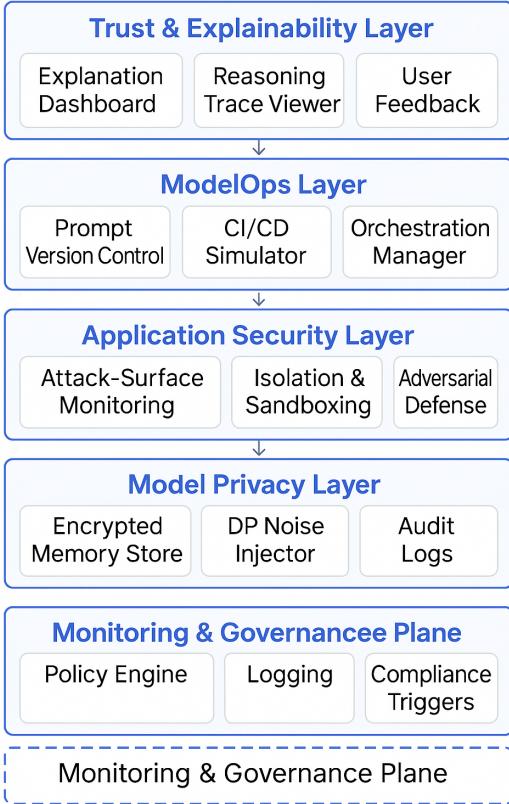


Figure 5: TRiSM-aligned architecture for Agentic AI systems. The system is organized into four layers: Trust and Explainability (e.g., reasoning traces, user feedback), ModelOps (prompt versioning, CI/CD, orchestration), Planning and Reasoning (multi-agent CoT), and Model Privacy (encrypted memory, differential privacy, audit logs). A cross-cutting Governance Plane enforces oversight via policies and logging. External tools are accessed through a secure API gateway. The user interface connects back to the trust and governance layers for human-in-the-loop (HITL) oversight. Abbreviations used: TRiSM = Trust, Risk, and Security Management; CoT = Chain-of-Thought; CI/CD = Continuous Integration and Deployment; DP = Differential Privacy.

a sub-question, and a specialized Reasoning Agent generates the partial chain-of-thought while Verification Agents and User-Interaction Agents cross-check the intermediate steps against knowledge graphs or human feedback. This agent-based decomposition localizes errors, prevents contradictions, and ensures that each partial solution is validated before the model proceeds.

- **Integrated multi-agent workflows:** Multi-agent workflows embed explainability into the process itself rather than as a post-hoc add-on. For example, in a DSM-5 mental-health diagnosis workflow, one agent collects patient information while a diagnostician agent retrieves relevant DSM-5 criteria and explicitly links conversational utterances to diagnostic rules. This integration produces transparent, step-by-step diagnostic predictions and shows why specific recommendations are made [147]. The study notes that step-by-step reasoning improves perceptions of fairness and reliability and that embedding a diagnostician agent

moves beyond general synthetic dialogues by providing a self-documenting rationale.

- **Modular resume-screening systems:** In recruitment, a multi-agent architecture divides the process into extraction, evaluation, summarization and formatting. The evaluation agent dynamically retrieves company-specific hiring criteria via Retrieval-Augmented Generation (RAG), and each stage remains clearly defined. This modularity allows recruiters to trace how a candidate was assessed and why a particular score was assigned [148].
- **Explainable credit assignment in multi-agent reinforcement learning:** A centralized LLM critic can provide individualized credit assignments to agents and explain its feedback strategy. Unlike earlier deep RL methods, this approach uses the language-based critic to justify why each agent receives a particular reward and to show how it decomposes the global reward [149].

**Approaches for explainability in multi-agent LLM systems** The following techniques adapt traditional XAI methods to the multi-agent setting:

- **Feature Attribution (LIME/ SHAP) for agents:** Techniques like LIME [112] or SHAP [113] can be extended to attribute an agent’s output to specific input components (e.g., particular context tokens or inter-agent messages). Highlighting which parts of the input or which inter-agent signals most influenced an agent’s action helps illuminate why a specific agent behaved as it did [150].
- **Counterfactual and Causal Analysis** Beyond local feature attributions, “*what-if*” analyses are increasingly important for multi-agent explainability. Counterfactual techniques examine how the system’s behavior would change if certain conditions or agent behaviors were altered [151]. For example, one can remove or modify a particular agent’s contribution and then observe how the collective outcome changes. This approach, rooted in causal inference, helps isolate each agent’s role in collaborative decision-making.
- **Reasoning Trace Logging:** Instead of treating the agents as black boxes, an explicit trace of their intermediate reasoning can greatly enhance transparency. Logging the step-by-step reasoning process – for example, an agent CoT prompts or the dialogue exchanged between agents – creates an audit trail of how the decision emerged [75]. By examining this trace, one can see how agents influence each other, which agent contributed what, and how partial conclusions build up to the final decision. This visibility into the multi-agent process addresses a key blind spot of standard XAI, effectively “opening up” the black box of inter-agent interactions.
- **Natural Language Explanations:** Multi-agent systems can include a separate explainer agent that synthesizes logs and feature attributions into human-readable narratives. This agent (often another LLM) translates the technical reasoning traces into conversational explanations that answer questions like “Why did the agents choose action X?” Integrated workflows, such as DSM5AgentFlow [147], demonstrate that embedding such explanation directly in the pipeline improves transparency and trust.
- **Role-oriented modular architectures:** Assigning clear roles to agents (e.g., reasoning, verification, summarization, credit assignment) enhances explainability because each stage of decision-making is distinct and can be inspected independently [148]. Modular designs also allow domain experts to adjust or replace specific agents without affecting the entire system.

- **Explainable reward critics:** In multi-agent reinforcement learning, LLM-based critics can decompose global rewards and provide natural-language justifications, making credit assignment transparent [149]. This approach helps developers understand how credit is distributed across agents and why certain behaviors are encouraged or discouraged.

#### *5.2.2. ModelOps : Lifecycle Management for LLM Agents*

ModelOps (Machine Learning Operations) extends the MLOps paradigm beyond model training and deployment to include governance, risk management and continuous oversight for all types of AI models [131]. Unlike MLOps, ModelOps manages not only individual models but also composite systems of agents and heterogeneous models across different business units.

ModelOps provides end-to-end oversight for AI in production by enforcing model governance and compliance: tracking lineage, ownership, and metadata to meet regulations such as GDPR and HIPAA , and by using policy-driven deployment gates so only approved models go live, with controlled roll-outs and roll-backs [132]. It continuously monitors risk and bias, detecting drift, fairness issues, hallucinations, and robustness regressions. ModelOps also manages heterogeneous assets: rule-based systems, LLMs, and composite multi-agent architectures; ensuring consistent visibility and lifecycle control across the portfolio. Finally, it integrates tightly with enterprise IT, wiring AI workflows into CI/CD, IT service management (ITSM), identity and access management (IAM), and cloud-security tooling [132]. For generative AI and AMAS, ModelOps must handle prompt versioning, guardrails and content filtering enforcement, fine-tuning lineage tracking, usage and cost observability, and ethical compliance. Effective ModelOps thus requires:

- **Version control and CI/CD pipelines:** Track versions of each agent's model and prompt configuration. Automated pipelines should test safety, performance and cost impacts whenever an agent's logic or model is updated, using multi-agent simulations to validate that new behaviours do not introduce regressions or unsafe interactions.
- **Hierarchical monitoring in multi-agent frameworks:** Systems such as MegaAgent [128] decompose tasks into subtasks and dynamically form groups of agents. They employ **hierarchical monitoring**, where each agent tracks its actions with a checklist, group-level administrators oversee their agents, and system-level supervisors review outputs. This layered monitoring ensures that changes to one agent do not compromise the entire system.
- **Risk measurement and oversight policies:** The NIST generative AI profile recommends documenting risks and updating measurement approaches regularly; it suggests using gradient-based attributions, occlusion and counterfactual prompts to improve explainability and transparency, and establishing policies for oversight across the lifecycle [126].

#### *5.2.3. Application Security for LLM Agents*

Security in AMAS systems must address vulnerabilities such as prompt injection, data exfiltration, and the misuse of external tools. According to OWASP [137, 138], prompt injection manipulates an LLM through malicious instructions, potentially causing the model to leak data, execute unauthorized commands or make harmful decisions. Multimodal models face cross-modal injection where malicious content in one modality influences the output of another. To mitigate these risks:

- **Prompt hygiene and hardening:** Sanitize and filter user inputs, apply secure prefixes, and validate instructions before they reach an agent. Constraining the output format and employing guardrails can limit the model's response surface [38].

- **Defense-in-depth design patterns:** Existing techniques such as adversarial training and user confirmation provide partial protection, but research shows they do not guarantee safety. Design patterns [133] like **Action-Selector** and **Plan-Then-Execute** restrict an agent's capabilities after processing untrusted input; the agent selects from a predefined list of allowed actions or produces a plan that cannot be altered by tool outputs, thereby isolating untrusted data from execution paths [58].
- **Least-privilege access and authentication:** Enforce strong authentication for both human users and agents and apply role-based access controls so agents can only invoke the minimum set of tools necessary to perform their task. Continuous monitoring should detect anomalous requests; if an agent deviates from normal patterns, automated systems should flag or suspend it [152].
- **Post-processing and anomaly detection:** Implement filters to detect and remove sensitive or inaccurate content from outputs and regularly retrain models to reduce hallucinations. Maintain robust logging and retention policies so that investigators can audit interactions without exposing private data.

Recent frameworks also emphasize **hierarchical monitoring** and cross-checking among agents to detect inconsistencies [128]. LangChain [153], AutoGen [98], and CrewAI [87] introduce the idea of trust scores or reputation among agents, where agents verify each other's outputs and cross-check decisions to catch inconsistencies or signs of compromise. Training agents with adversarial examples [134] and red-teaming [154] scenarios further strengthens robustness against malicious inputs.

#### 5.2.4. Model Privacy and Data Protection

AMAS also poses unique privacy challenges because agents often share information via memory stores and message passing. Without strict controls, an agent may inadvertently leak sensitive data [40]. A layered privacy strategy should include:

- **Differential Privacy (DP):** Adding calibrated noise to training data prevents individual data points from being re-identified and reduces the risk that models memorize or regurgitate personal information [139]. DP should be assessed on a case-by-case basis, since heavy noise can harm accuracy.
- **Data anonymization and minimization:** Apply robust anonymization and pseudonymization techniques and regularly test them for effectiveness [140]. Limit data collection to what is strictly necessary for the agent's task and enforce strict retention policies.
- **Secure Multi-Party Computation (MPC):** MPC allows parties to jointly compute a function without revealing their private inputs. In privacy-preserving LLM services, secure MPC can protect user prompts from the model provider; for example, the PUMA framework uses MPC to ensure that prompts remain hidden during inference, while maintaining reasonable performance (it can evaluate a 7-billion-parameter model within minutes) [141].
- **Homomorphic Encryption (HE):** HE enables computation on encrypted data and offers cryptographically provable privacy. Recent work demonstrates an encryption-friendly LLM architecture that uses LoRA fine-tuning and Gaussian kernels to speed up fully homomorphic encryption (FHE) operations, allowing encrypted inference with performance comparable to plaintext models [142].

- **Trusted Execution Environments (TEEs):** TEEs provide hardware-enforced isolation that protects code and data [143]. They partition processors into secure and normal worlds; implementations include Intel SGX, AMD SEV and Arm TrustZone. In secure LLM inference, TEEs ensure that models run only inside authenticated enclaves, with attestation verifying model integrity.
- **Continuous monitoring and access controls:** Encrypt data at rest and in transit, restrict access to authorized personnel, and use monitoring to detect data poisoning or unusual behaviour.

These techniques, combined with careful consent management and compliance with local regulations, help maintain confidentiality when agents share or process sensitive information.

Figure 6: Governance alignment of TRiSM pillars with international frameworks such as the EU AI Act, NIST AI RMF, ISO/IEC 42001, 42005, and OECD Principles. Each pillar operationalizes specific compliance strategies to ensure transparency, resilience, and trust in LLM-based systems.

Alignment of TRiSM Pillars with Governance Frameworks	
<b>Explainability:</b>	Aligned with the EU AI Act (Art. 13–14) [125] and the NIST AI RMF trustworthiness characteristic “Explainability & Interpretability” [3]; also supported by NISTIR 8312 (Four Principles of Explainable AI) [155]. Promotes well-justified (rationale) and human-interpretable outputs with traceable logs [119].
<b>ModelOps:</b>	Anchored in the NIST AI RMF “Govern/Manage” functions [3] and ISO/IEC 42001:2023 (AI management system) [156]; secure SDLC controls via NIST SP 800-218/218A <sup>a</sup> . Emphasizes lifecycle gates, model versioning, rollback policies, and CI/CD with safety gates.
<b>Risk Management &amp; Impact Assessment:</b>	EU AI Act (Art. 9, 72–73) [125], ISO/IEC 23894 (AI risk management) <sup>b</sup> , and ISO/IEC 42005:2025 (AI impact assessment) [129]. Iterative hazard analysis, impact registers, incident response, and disclosure playbooks.
<b>Data Governance &amp; Quality:</b>	EU AI Act (Art. 10) [125] mapped to the ISO/IEC 5259 series (data-quality model, measures, processes, governance) <sup>c</sup> . Data minimization, lineage, coverage/bias checks, and sampling documentation.
<b>Robustness &amp; Safety:</b>	EU AI Act (Art. 15) [125] and ISO/IEC 24029-1 (robustness assessment) [32]. Adversarial/perturbation testing, fault tolerance, fallback/kill-switch triggers, and rollback criteria.
<b>Application Security:</b>	Guided by OWASP LLM Top-10 [137] and OECD AI Principles on robustness [127]. Access controls, prompt/input mediation, sandboxing, egress controls, and auditable trails.
<b>Model Privacy:</b>	Aligned with GDPR (Art. 25) [34], HIPAA §164 [35], and OECD AI Principles [127]. Encryption in transit/at rest, DP budgeting, data minimization, and access-audit pass thresholds.
<b>Documentation &amp; Traceability:</b>	EU AI Act (Arts. 11–12; Annex IV) [125]. Complete technical documentation, event logs, model/data cards, decision records, and regulator-ready evidence packages.
<b>Post-Market Monitoring:</b>	EU AI Act (Arts. 72–73) [125]. Telemetry for drift/misuse, continuous evaluations, feedback loops, controlled recalls, and stakeholder notifications.
<b>Governance:</b>	Grounded in GDPR, CCPA [144], and OECD AI Principles [127]. Institutional oversight, policy compliance, role-based accountability, regulatory traceability, and human-in-the-loop for high-risk/multi-agent deployments.
<b>Fairness &amp; Human Rights:</b>	OECD AI Principles [127] and ISO/IEC TR 24027 (bias) <sup>d</sup> . Lifecycle bias controls, affected-party impact reviews, and mitigation sign-offs.

<sup>a</sup>NIST, 2024

<sup>b</sup>ISO/IEC 23894

<sup>c</sup>ISO/IEC 5259 series

<sup>d</sup>ISO/IEC TR 24027

### 5.2.5. Governance

Governance integrates ethical, legal and societal considerations into the design and operation of AI systems. Regulations such as the EU AI Act [125], NIST AI RMF [3] and OECD AI

[127] provide guidance that aligns with TRiSM pillars of Explainability, ModelOps, Security and Privacy.

- **EU AI Act - Human Oversight (Article 14):** High-risk AI systems must be designed so that natural persons can effectively oversee them and prevent or minimize risks to health, safety or fundamental rights [1]. The AI system should enable overseers to understand capabilities and limitations, detect anomalies, interpret outputs, decide not to use the system, or stop its operation through a ‘stop’ button. For certain identification tasks, decisions must be independently verified by at least two competent individuals.
- **OECD AI Principles:** These intergovernmental principles, updated in 2024, promote human-centric, trustworthy AI [127]. They emphasise transparency and explainability: AI actors should provide meaningful information about data sources, factors and logic to allow affected individuals to understand and challenge outputs. Robustness, security and safety require AI systems to function appropriately under normal and adverse conditions and include mechanisms to override or decommission systems to prevent harm. Accountability means AI actors are responsible for the proper functioning of AI systems; they must ensure traceability of datasets and decisions and apply systematic risk management throughout the lifecycle.
- **NIST AI Risk Management Framework (AI RMF):** The AI RMF and its Generative AI profile outline four core functions: govern, map, measure and manage [3]. For generative AI, NIST recommends documenting risks and potential impacts, applying interpretability techniques (gradient-based attributions, occlusion, counterfactual prompts) to improve transparency, and establishing policies for oversight across the AI lifecycle.
- **Auditability and role-based access:** Agentic AI systems should maintain auditable logs of agent actions, decisions and data flows and assign clear roles (e.g., reasoning, verification, summarisation) so that responsibilities are transparent [157]. Role-based access controls help enforce accountability and prevent unauthorized agents from accessing sensitive functions.
- **Global harmonisation:** Many jurisdictions (e.g., GDPR, CCPA) require data subject rights, consent and privacy by design. Frameworks like the OECD Principles and NIST AI RMF promote international cooperation and interoperability.

These governance frameworks (as illustrated in Figure 6) underscore the need for transparent accountability structures, human oversight, robust privacy controls, and documented processes in AMAS. TRiSM provides a practical scaffolding for operationalizing these requirements, aligning lifecycle management, security, privacy and governance to build trustworthy Agentic AI.

Next, we discuss the evaluations in AMAS.

## 6. Evaluation of Agentic AI Systems

Agentic AI systems demand evaluation beyond traditional accuracy. We outline five categories: *trustworthiness, explainability, user-centered performance, coordination, and composite scores*, each capturing a distinct facet of performance and real-world impact. Compared to a related work [158], which focuses on proportional impacts across general AI, our work targets Agentic AI

Table 5: Summary of key evaluation metrics for agentic AI systems. Each category addresses a distinct facet of performance and behavior.

Metric Aspect	Evaluation Focus (Examples of Metrics)
Trustworthiness	Reliability, safety, and alignment. <i>Examples:</i> success rate under shifts; safety/ethics violation rate (lower is better); calibration (ECE/Brier <sup>4</sup> ); fairness indices; red-team/jailbreak success (lower is better).
Explainability	Transparency and traceability. <i>Examples:</i> explanation coverage; faithfulness/fidelity; stability across similar cases; human interpretability scores; explanation time/cost.
User-Centered	Experience and outcomes. <i>Examples:</i> satisfaction (Likert/CSAT <sup>5</sup> ); goal-completion rate; clarification turns; perceived coherence/naturalness; task time; cognitive load (NASA-TLX).
Coordination	Collaboration in multi-agent or modular setups. <i>Examples:</i> team task completion; coordination overhead (messages/tokens, rounds to converge); plan consistency; deadlock/conflict rate (lower is better); <i>synergy score</i> for complementary actions.
Composite	Holistic summaries. <i>Examples:</i> weighted aggregate across categories (weights sum to 1); Tool-Use Efficacy (TUE, see definition); cross-domain benchmark aggregates.

and links risks to their causal sources (e.g., prompt injection, memory poisoning). We further introduce an evaluation template with new metrics the *Component Synergy Score (CSS)* and *Tool Utilization Efficacy (TUE)* to capture multi-agent dynamics. A summary appears in Table 5.

**Trustworthiness.** This criterion evaluates reliability, safety, and ethical alignment [159]. Core signals include task success rates under distributional shifts (robustness), violation rate of safety/ethical rules (lower is better), and calibration quality (e.g., Expected Calibration Error, ECE). To avoid pathological behavior in scoring, we define a normalized trustworthiness index

$$T = \frac{w_C C + w_R R + w_I I}{1 + \lambda S}, \quad (1)$$

where  $C$  (credibility/accuracy),  $R$  (reliability over time), and  $I$  (user alignment/rapport) are normalized to  $[0, 1]$ ,  $S \geq 0$  is self-orientation (higher is worse),  $w_C, w_R, w_I \geq 0$  with  $w_C + w_R + w_I = 1$ , and  $\lambda \geq 0$  controls how strongly self-orientation penalizes trust.

**Explainability.** Explainability metrics assess how well human evaluators can understand and trace decisions. Useful measures include *coverage* (fraction of outputs with an explanation), *fidelity/faithfulness* (agreement with the model’s actual decision basis), *stability* (consistency across similar cases), and human *interpretability* ratings. Suites like OpenXAI [160] report faithfulness, stability, and fairness of explanations. High explainability builds user trust and aids debugging; in regulated domains (e.g., healthcare, finance) it is often essential for compliance.

**User-Centered Performance.** User-centered metrics capture how effectively the agent satisfies end-user needs [161]. Signals include post-interaction satisfaction scores, goal-fulfillment rate, number of clarification turns (lower is better), and perceived coherence/naturalness. Human-in-the-loop (HITL) studies typically rate helpfulness, clarity, and instruction adherence. Ultimately, a user-centered agent aligns actions with user intent and preferences.

**Coordination.** In multi-agent or modular systems, coordination captures whether components maintain a shared plan, minimize redundant work, and resolve dependencies. Quantify with team success rate, communication efficiency (messages/tokens and rounds to consensus), plan/belief

consistency, deadlock/conflict rate, and a *Component Synergy Score* (CSS) that credits inter-agent actions which enable or improve peers' outcomes (higher is better).

*Composite Metrics..* Composite metrics are often presented as a weighted scalar summary across dimensions:

$$M_{\text{composite}} = w_T M_T + w_E M_E + w_U M_U + w_C M_C, \quad (2)$$

where  $M_T, M_E, M_U, M_C \in [0, 1]$  are normalized category scores for trustworthiness, explainability, user-centered performance, and coordination, respectively, and  $w_T, w_E, w_U, w_C \geq 0$  with  $\sum w_* = 1$  encode domain priorities. While this provides a scalar index for ease of interpretation, it risks *masking* over-performance in one dimension with under-performance in another. To address this, we treat the evaluation more rigorously as a *vector of metrics*,

$$\mathbf{M} = (M_T, M_E, M_U, M_C), \quad (3)$$

and recommend selection based on arg max over  $\mathbf{M}$  rather than on any scalarization:

$$\arg \max_i \mathbf{M}_i. \quad (4)$$

In this view, Eq. 2 can be interpreted as a limiting case, useful for communicating results to non-technical stakeholders, while Eq. 3 and Eq. 4 form the principled basis for evaluation and comparison.

**Definition (Tool-Use Efficacy, TUE).** In this work, we introduce *Tool-Use Efficacy (TUE)* as a specialized composite for evaluating how correctly and efficiently an agent uses external tools. Let

- Sel  $\in [0, 1]$  (tool decision quality: precision/recall or F1 of invoking/abstaining when needed),
- Arg  $\in [0, 1]$  (argument validity: fraction of calls with syntactically/semantically valid parameters),
- Exec  $\in [0, 1]$  (execution success: no exceptions/timeouts, correct API usage),
- Out  $\in [0, 1]$  (post-call outcome correctness on tool-gated sub-tasks),
- Eff  $\in [0, 1]$  (efficiency: normalized inverse cost for latency/tokens/#calls).

We define two aggregators (both reportable; choose per application):

(a) *Linear aggregator (auditable weights)*.

$$\text{TUE}_{\text{lin}} = \alpha_s \text{Sel} + \alpha_a \text{Arg} + \alpha_e \text{Exec} + \alpha_o \text{Out} + \alpha_f \text{Eff}, \quad \alpha_* \geq 0, \quad \sum \alpha_* = 1. \quad (5)$$

(b) *Geometric aggregator (“all-of-the-above” strictness)*.

$$\text{TUE}_{\text{geo}} = (\text{Sel}^{\beta_s} \text{Arg}^{\beta_a} \text{Exec}^{\beta_e} \text{Out}^{\beta_o} \text{Eff}^{\beta_f})^{1/(\beta_s + \beta_a + \beta_e + \beta_o + \beta_f)}, \quad \beta_* > 0. \quad (6)$$

*Overhead penalty.* If the preference is a separate cost term, then we can define normalized overhead OH  $\in [0, 1]$  (increasing in time/tokens/calls) and report the penalized score

$$\text{TUE}^* = \frac{\text{TUE}}{1 + \lambda \text{OH}}, \quad \lambda \geq 0, \quad (7)$$

where TUE is either (5) or (6). It is recommended to always report the components (Sel, Arg, Exec, Out, Eff) with confidence intervals to ensure reproducibility.

## 7. Security and Privacy in Agentic AI Systems

In this section, we discuss the security and privacy in AMAS.

### 7.1. Security Mechanisms

Agentic AI systems, composed of loosely coupled yet collaboratively functioning LLM-based agents, introduce an expanded attack surface relative to conventional AI agents [162]. Ensuring the security of such systems necessitates a multi-layered defense architecture that addresses data protection, execution integrity, inter-agent communication, and model robustness [163]. Among the foundational techniques employed are *encryption*, *access control*, *adversarial defense*, and *runtime monitoring*, each adapted to the unique demands of decentralized multi-agent environments.

**Encryption.** Encryption plays an important role in safeguarding data exchanged between multiple agents, especially when sensitive or regulated content (e.g., healthcare records, financial data) is involved [67]. Agentic workflows often include inter-agent handoff of partially processed results, models, or prompts. Implementations such as SSL/TLS, homomorphic encryption, and secure enclaves (e.g., Intel SGX) are increasingly integrated into Agentic AI pipelines to ensure confidentiality across message-passing protocols.

**Access control.** Access control is highly important when orchestrators or shared memory modules manage permissions for agents with distinct capabilities and responsibilities. For instance, in systems like AutoGen and CrewAI where agents take on specialized roles (e.g., summarizer, planner, coder); enforcing principle-of-least-privilege access prevents privilege escalation and unauthorized tool invocation [57]. Agent-based access control policies, often aligned with Role-Based Access Control (RBAC) [157] and Attribute-Based Access Control (ABAC) [157] paradigms, can dynamically restrict which agents may access sensitive APIs, files, or memory buffers, based on contextual trust levels.

**Adversarial learning.** Adversarial learning is a growing concern as LLM-based agents are susceptible to prompt injection, manipulation through poisoned tool outputs, or coordination disruption via malformed intermediate results. Recent studies have shown that multi-agent LLM frameworks can be destabilized by adversarially designing outputs from one compromised agent propagating misleading information to others [164]. Adversarial training methods, such as input perturbation, reward shaping, and contrastive learning, can partially mitigate these vulnerabilities. Integrating safety constraints and verifying tool responses before execution are also effective mitigation strategies.

**Runtime Monitoring.** Runtime monitoring systems support the detection of anomalous agent behaviors, especially in high-stakes domains like automated healthcare or cybersecurity. Log-based auditing, anomaly detection with LSTM or autoencoder-based detectors, and trust scoring among agents are becoming essential components of real-time surveillance layers [165]. For example, Microsoft’s Copilot governance layers monitor anomalous agent behavior across sessions to ensure compliant execution and flag potentially harmful interactions.

### 7.2. Privacy-Preserving Techniques

The decentralized and interactive nature of AMAS introduces new challenges for preserving privacy, especially as agents continuously communicate, access external data sources, and store episodic or shared memory. To ensure data confidentiality and protect personally identifiable information (PII), Agentic AI systems must adopt robust privacy-preserving techniques such as *differential privacy*, *data minimization*, and *secure computation*.

**Differential privacy (DP).** DP offers mathematically grounded guarantees by injecting statistical noise into outputs, ensuring that individual user contributions cannot be re-identified. In multi-agent LLM systems, DP can be applied during training or at inference-time when agents exchange information. For example, Google’s implementation of DP in federated learning frameworks can be extended to distributed agentic systems, where agents collaboratively train or fine-tune local models without exposing raw data [166]. DP-SGD and privacy budgets ( $\epsilon$ -differentials) can regulate information exposure during policy updates or collaborative planning in real-time decision-making agents.

**Data minimization.** Agentic AI systems can mitigate exposure risks by limiting the scope, granularity, and duration of data collected or retained during task execution. For instance, temporary memory buffers used in systems like ChatDev [30] or ReAct-based pipelines [75] are cleared once subgoals are completed, preventing persistent storage of unnecessary user data. Furthermore, anonymization and pseudonym [167] techniques can help remove identifying features before data is passed between agents or stored in shared memory repositories.

**Secure computation.** Techniques including secure multi-party computation (SMPC) [168], homomorphic encryption [169], and trusted execution environments (TEEs)<sup>6</sup> enable agents to perform computations over encrypted or obfuscated data without compromising privacy. In scenarios where agents collaborate across different organizational boundaries (e.g., federated medical agents or cross-silo industrial agents), SMPC allows joint computations such as diagnostics or anomaly detection without data leakage. Homomorphic encryption, while computationally expensive, is increasingly being explored to allow arithmetic operations on encrypted vectors used in RAG workflows.

Privacy-by-design [170] principles are becoming central to the engineering of next-generation Agentic AI systems. Architectures now embed user consent layers, configurable privacy settings, and memory redaction modules that allow end-users or system administrators to control what agents can remember or share. As Agentic AI expands into domains such as personalized education, healthcare, and finance, ensuring privacy-respecting behaviors will be essential for regulatory compliance (e.g., GDPR, HIPAA) and public trust.

Next, we discuss the compliance to AI governance and policy in Agentic AI systems.

Table 6: Governance controls for agentic AI systems: pillar → control → operationalization → evidence → standards.

Pillar	Control	Operationalization (How)	Evidence / Artifacts	Standards & Articles
Governance	Org. accountability	Define owners for models/agents/tools; RACI for changes; approvals for high-impact actions	Governance policy; RACI records; change approvals	NIST (Govern) AI RMF [171]; ISO/IEC 42001 [2]
Governance	Policy-as-code	Encode allow/deny rules in orchestrator; pre-exec checks and human sign-off gates	Policy repository; policy test cases; gate logs	EU AI Act Arts. 11–14 [125]; ISO/IEC 42001 [2]
Risk Mgmt	Risk register & AIA	Hazard/threat enumeration; likelihood/impact scoring; AI Impact Assessment (AIA)	Risk register; AIA reports; mitigation plan	ISO/IEC 23894; ISO/IEC 42005 [129]; EU AI Act Art. 9 [125]
Risk Mgmt	Post-market monitoring	Drift/jailbreak telemetry; incident thresholds; rollback playbooks	PMM plan; incident tickets; root-cause analysis	EU AI Act Arts. 72–73 [125]; NIST AI RMF (Manage) [171]
Data Gov	Data minimization & lineage	Task/tenant-scoped memory; TTLs/redaction; dataset lineage and licenses	Data inventory; lineage graphs; retention logs	EU GDPR Art. 25 [34]; ISO/IEC 5259 series

*Continued on next page*

<sup>6</sup>Trusted Execution Environment (Wikipedia)

Pillar	Control	Operationalization (How)	Evidence / Artifacts	Standards & Articles
Data Gov	Quality & bias checks	Coverage/balance checks; label noise audits; sampling documentation	Data quality reports; bias audit results	EU AI Act Art. 10 [125]; ISO/IEC TR 24027
	Explainability	Log rationales; test explanation faithfulness/stability; user-facing summaries	XAI eval report; model-/data/agent cards	NISTIR 8312 [155]; NIST AI RMF (Map/Measure) [171]
Documentation	Technical documentation	Annex-IV tech docs: purpose, data, performance, oversight, foreseeable misuse	Annex-IV dossier; limitations & misuse notes	EU AI Act Arts. 11–12; Annex IV [125]
Traceability	Provenance & logging	Log prompts, plans, tool calls, I/O, timestamps, agent role; sign releases	Immutable logs (WORM/tamper-evident); signed artifacts	EU AI Act Arts. 12–13 [125]; ISO/IEC 42001 [2]
Human Oversight	Human-in-the-loop gates	Require human approval for sensitive plans (delete, external write, PII)	Approval records; override/rollback logs	EU AI Act Art. 14 [125]; NIST AI RMF (Manage) [171]
Robustness	Stress/red-team testing	Adversarial prompts; perturbation/fault tests; safety regression suites	Red-team report; robustness curves; residual risk	EU AI Act Art. 15 [125]; ISO/IEC 24029-1 [32]
Security	I/O mediation	Prompt sanitization; output filtering; sensitive-data (PII/PHI/secrets) detection	Filter policies; violation/egress logs	OWASP LLM Top-10 [137]; HIPAA §164 [35]
Security	Tool access control	RBAC/ABAC per agent/role; per-tool API tokens; argument validation; allowlists	Access matrices; token scopes; arg-validator tests	NIST SP 800-218/218A; NIST AI RMF (Manage)
Security	Isolation/sandboxing	Containers/VMs; syscall/network/file egress policies; hardware TEEs for sensitive code	Sandbox configs; egress policy; attestation logs	NIST SP 800-218A; TEEs best-practice notes
Privacy	DP budgeting & consent	Differential privacy budgets; consent bases; data-subject rights flows	DP budget ledger; DSR tickets; consent records	GDPR Art. 25 [34]; OECD principles [127]
Supply Chain	Provenance & SBOM	Track sources and hashes for models/tools/datasets; dependency scanning	SBOM; third-party attestations; license checks	NIST SSDF (SP 800-218/218A); ISO/IEC 42001
Coordination	Role separation & least privilege	Distinct agent roles (planner, coder, reviewer); minimal cross-role permissions	Role definitions; privilege matrices; audit samples	NIST AI RMF (Govern) [171]; ISO/IEC 42001
Coordination	Plan consistency checks	Shared plan/belief verification; conflict/deadlock detection; CSS metric	Plan snapshots; CSS scores; conflict logs	Multi-agent eval practice; robustness [32]
Monitoring	Anomaly & exfil detection	Detect unusual tool chains, volume spikes, strange domains; rate-limit/kill switch	Anomaly dashboards; blocked-egress records	OWASP LLM Top-10 [137]; NIST AI RMF (Manage)
Transparency	User notices & recourse	User-facing disclosures, capabilities/limits, and appeal/feedback channels	Notices; feedback logs; SLA for appeals	OECD principles [127]; EU AI Act transparency
Evaluation	Fairness & calibration	Group metrics; ECE/Brier; CI reporting; risk-benefit trade-off docs	Fairness report; calibration plots; CI tables	NIST AI RMF (Measure) [171]; ISO/IEC TR 24027
Records	Release/rollback discipline	Versioned releases; rollback points; emergency disable/recall	Release notes; rollback attestations	EU AI Act PMM [125]; ISO/IEC 42001
GPAI	GPAI obligations	Provide model cards, training-data summary, eval reports, and usage restrictions; disclose known limitations	GPAI documentation pack; model card; eval report	EU AI Act (GPAI duties) [125]; ISO/IEC 42001
Transparency	Synthetic content disclosure	Label AI-generated/edited media; attach provenance (e.g., C2PA) and user notices	Watermarking/provenance config; disclosure logs	EU AI Act Art. 52 [125]; C2PA <sup>7</sup>
Privacy	Cross-border transfers	SCCs/TIA for third-country transfers; data-residency controls; key management	SCC/TIA records; DPA; data-map; key-custody logs	GDPR Ch. V [34]
Privacy	Data-subject rights ops	Verified workflows for access, erasure, rectification, restriction, portability, objection	DSR tickets/SLAs; fulfillment proofs; audit samples	GDPR Arts. 12–23 [34]
Operations	Change mgmt & release gates	Risk-based pre-release checklist; dual-control approvals; staged rollouts; rollback criteria	Change tickets; gate results; approvals; rollback points	ISO/IEC 42001; NIST AI RMF (Manage) [171]
Supply Chain	Third-party/vendor risk	Vendor questionnaires; DPAs; penetration/attestation reports; license compliance	Vendor risk register; DPA file; attestation/SBOM	NIST SSDF (SP 800-218/218A); ISO/IEC 42001

*Continued on next page*

<sup>7</sup>c2pa.org

Pillar	Control	Operationalization (How)	Evidence / Artifacts	Standards & Articles
Human Factors	HCD & accessibility	Human-centred design reviews; usability risk analysis; accessibility conformance checks	HCD review notes; usability test reports; ally checklists	ISO 9241-210 <sup>8</sup> ; OECD principles [127]
Sustainability	Energy/CO <sub>2</sub> telemetry	Track/normalize energy use and CO <sub>2</sub> ; report per release; set budgets	Energy/CO <sub>2</sub> logs; budget vs. actual; disclosure note	Org policy / reporting guidance

## 8. Compliance to AI Governance and Policy in Agentic AI Systems

Recent industry surveys indicate that enterprise leaders deploying Agentic AI at scale rank security, trust, compliance, and oversight among their top concerns. A survey<sup>9</sup> of IT professionals found that while 98% plan to expand AI agent usage, 96% view them as growing security threats, with only 54% having full visibility into their data access. In parallel, 67% of business leaders are increasing budgets for security oversight<sup>10</sup>, and 53% specifically cite data privacy and compliance as their primary challenge in scaling Agentic AI<sup>11</sup>. Table 6 operationalizes the governance controls into concrete implementations and evidence and next we detail the compliance to AI goverace and policy in Agentic AI systems.

### 8.1. Regulatory Standards

Regulatory frameworks provide baseline requirements that all AI systems, including agent-based architectures, must meet. Key frameworks include:

- *NIST AI Risk Management Framework (AI RMF)* [171]. A voluntary U.S. framework organized around the *Govern, Map, Measure, Manage* functions. It emphasizes organizational governance, risk identification and measurement, and operational risk treatments. Mapping to the RMF also benefits from NISTIR 8312 (Explainable AI principles) for transparency obligations.
- *EU AI Act* [125]. A comprehensive EU regulation with phased application: bans on certain practices apply after six months (Feb 2025), obligations for general-purpose AI begin Aug 2025, and most high-risk system obligations (e.g., risk management, documentation, human oversight, post-market monitoring) start Aug 2026. High-risk systems require ongoing risk assessment, technical documentation (Annex IV), logging/traceability (Arts. 11–12), data governance (Art. 10), human oversight (Art. 14), and robustness/accuracy (Art. 15).
- *ISO/IEC management and assessment standards*. ISO/IEC 42001:2023 (AI management systems) sets organization-level governance requirements; ISO/IEC 42005:2025 (AI impact assessment) provides system-level impact-assessment guidance; ISO/IEC 23894 addresses AI risk management; ISO/IEC 24029-1 supports robustness evaluation; ISO/IEC TR 24027 addresses bias. These align well with AI RMF and the EU AI Act's lifecycle duties.
- *Domain-specific laws*. Sectoral/privacy regimes still apply: GDPR (e.g., data protection by design, Art. 25), HIPAA (45 CFR §164), and regional data-residency rules. Agentic systems should enforce data minimization, purpose limitation, and consent bases; consider *DPIA* for high-risk processing and *AI impact assessments* per ISO/IEC 42005 in parallel.

<sup>8</sup>ISO 9241-210

<sup>9</sup>TechRadar (2024)

<sup>10</sup>Cybersecurity Dive (2024)

<sup>11</sup>Cloudera (2024)

## 8.2. Auditability

Multi-agent systems can produce emergent and opaque behaviors; auditors must reconstruct the chain of decisions.

- *Comprehensive logging.* Record prompts/context, plans, actions, tool calls, inputs/outputs, timestamps, and agent role/identity; capture rationales where feasible. Preserve hashes and version IDs for models, tools, and data snapshots.
- *Action traceability.* Maintain decision provenance across agents (planner→coder→tester, etc.), including hand-offs and approvals. Link every external effect (e.g., file write, API call) to the proposing and approving principals.
- *Role-granular trails.* Tag entries by agent role (e.g., Researcher, Coder, Reviewer) to localize anomalies and bias. Require reviewer countersignatures for high-impact actions.
- *Immutability.* Use append-only, tamper-evident storage (e.g., WORM buckets, cryptographic sealing) for audit logs; rotate keys and attest integrity regularly.

## 8.3. Policy Enforcement

While regulations and audits provide oversight and after-the-fact analysis, policy enforcement mechanisms work in real-time to keep an Agentic AI system's behavior within allowed bounds. Policy in this context refers to the codified rules and constraints that the AI agents must obey. In practice, effective policy enforcement in agentic AI includes:

- *Orchestrator-level controls.* A meta-controller enforces allow/deny policies for plans and tool calls; escalate to human approval for sensitive operations (e.g., data deletion, external writes).
- *Memory & retention.* Implement TTLs, redaction, and scoped memory (task- and tenant-scoped) to satisfy minimization and purpose limits; support subject-access and deletion.
- *Tool & data access.* Enforce RBAC/ABAC for agents/roles; use per-tool API tokens, least privilege, and environment scoping (dev/stage/prod). Validate/normalize tool arguments.
- *Isolation.* Run untrusted code/tooling in sandboxes (e.g., containers/VMs) with syscall/network/file egress policies; prefer hardware isolation (TEEs) for sensitive workloads.
- *I/O mediation.* Apply content filters and egress controls on model outputs; mediate RAG inputs to prevent injection/exfiltration; detect sensitive data (PII/PHI, secrets) before release.
- *Real-time monitoring.* Track anomaly signals (policy violations, unusual tool chains, data exfiltration patterns) and enforce runtime kill-switches, rollbacks, and incident triggers.

## 8.4. Documentation & Record-Keeping (Evidence)

To meet audit and regulator expectations, maintain:

- *Technical documentation* (EU AI Act [1]): system purpose and risk class, training/eval data governance, performance, limitations, foreseeable misuse, human oversight measures.
- *Model/Data cards* with lineage, licenses, and known hazards; *evaluation reports* with stress tests (robustness, red-teaming), calibration/fairness metrics, and residual risk.
- *Versioned artifacts:* model versions, datasets, prompts, agents' policies, and tool catalogs with change logs; signed releases and rollback points.

## 8.5. Post-Market Monitoring & Incident Response

Agentic systems must be monitored after deployment and respond to harm quickly.

- *Telemetry and drift.* Continuously measure real-world performance, jailbreak/abuse rates, and data distribution shifts; trigger retraining or guard updates.

- *Serious-incident handling.* Define thresholds, triage workflows, and reporting timelines; keep case files linking logs, affected users, and remediation.
- *Periodic reassessment.* Re-run impact/risk assessments (ISO/IEC 42005 [129], ISO/IEC 23894 [130] ) when capabilities, data, or context change materially.

### 8.6. Supply-Chain & Secure Development (GenAI)

- Strengthen provenance and development hygiene for agents and tools.
- *Secure SDLC for GenAI.* Follow NIST SP 800-218 (SSDF)<sup>12</sup> and the GenAI profile SP 800-218A<sup>13</sup> for model/system development; integrate threat modeling for agent tool-chains and LLM-specific risks.
  - *OWASP LLM risks.* Mitigate prompt injection, insecure output handling, data poisoning, model DoS, and supply-chain risks using the OWASP [137] Top 10 for LLM applications.
  - *Provenance.* Track model/tool/dataset origins, licenses, and checksums; require attestations for third-party components and run dependency/secret scans in CI.

Bringing Agentic systems into compliance is not a single checklist; it is a lifecycle program that couples standards (NIST AI RMF, ISO/IEC 42001/42005), regulation (EU AI Act), secure development (NIST SP 800-218/218A), and runtime enforcement (policies, isolation, monitoring). This combination creates evidenceable trust and reduces the blast radius of inevitable failures.

## 9. Discussion

In this section, we present our findings and outline directions for future work.

### 9.1. Technical Implications of TRiSM for Agentic AI Design

Rather than treating LLM agents as black-box decision-makers, TRiSM encourages instrumenting them with continuous oversight “guardrails” [172]. For example, there is discussion on designing specialized “guardian agents” within AMAS [42]. In this paradigm, such agents serve as proactive monitors that filter sensitive data and establish baselines of normal behavior, while operator agents dynamically enforce policies at runtime (e.g. blocking disallowed actions such as outputting personally identifiable information).

Prior research highlights risks like “excessive agency” [138] where an LLM given too much autonomy or tool access can produce unintended harmful actions (for instance, via hallucination or misinterpreted goals). TRiSM-driven Agentic AI design mitigates these risks by limiting agent autonomy within well-defined safety bounds. Likewise, emerging threats specific to Agentic AI systems such as prompt injection attacks, memory poisoning , or cascading hallucinations, highlights the need for more built in risk controls. By incorporating anomaly detection and policy-checking modules, an LLM agent can detect deviations from normal behavior and either alert humans or automatically neutralize the threat (e.g. masking a sensitive datum or stopping an unsafe action).

---

<sup>12</sup>NIST SP 800-218 (SSDF)

<sup>13</sup>GenAI profile SP 800-218A

## *9.2. Ethical and Societal Ramifications of Multi-Agent AI*

Beyond technical matters, deploying networks of autonomous LLM agents raises pressing ethical and societal questions. Applying TRiSM in this context emphasizes principles of accountability, human oversight, and fairness [173], all of which are vital for public trust in AI systems. Governance in TRiSM recommends that organizations retain clear responsibility for their AI actions, and do not obscure blame behind algorithmic “black boxes”. This implies implementing audit trails and explicable decision logs so that any harmful or biased outcome can be traced and attributed . Recent guidance on trustworthy AI [174] often highlights accountability and explainability as key pillars of trust.

In practice, this review sheds some light on how each autonomous agent decisions should be transparent enough to be understood and challenged by human reviewers when necessary [175]. Human oversight is another ethical imperative tightly coupled with accountability. TRiSM does not seek to eliminate humans from the loop; rather, it provides a structured way for humans and AI agents to collaborate under a defined governance framework [176].

In AMAS setups, humans can monitor agent swarms in real time, pause or shut down agents exhibiting anomalies, and adjust policies on the fly . Therisk of “user complacency”, trusting an autonomous agent too much, has been noted as a hazard [177]. TRiSM governance counteracts this by formalizing oversight roles and ensuring no AI operates without appropriate human or regulatory supervision.

## *9.3. Alignment with Emerging AI Regulations and Standards*

The principles embedded in TRiSM align closely with emerging regulatory frameworks for AI. This convergence means that adopting TRiSM-based governance can help organizations meet new legal obligations and industry standards. For example, the European Union’s AI Act [125] (set to fully apply in 2026) mandates rigorous risk management, transparency, data governance, and human oversight for “high-risk” AI systems. These are precisely the capabilities that a TRiSM approach promotes.

Deploying Agentic AI systems responsibly requires a multifaceted governance approach. By adhering to established regulatory standards (and keeping abreast of new ones), by ensuring auditability for transparency and accountability, and by enforcing policies through technical and organizational means, we can significantly mitigate the risks posed by autonomous AI agents. Moreover, embracing evolving best practices – from governance boards to adaptive, code-driven policies – will help maintain control over these systems as they grow in complexity. The end goal is to harness the benefits of Agentic AI (increased efficiency, creativity, and problem-solving at scale) while staying within legal, ethical, and safe bounds, thereby preserving public trust and meeting compliance obligations.

## *9.4. Limitations and Current Research Gaps*

While the TRiSM-based approach appears promising, we also discuss some limitations and open challenges in current research. First, the classical AI domains that works on standard test suites, there is no consensus on how to measure an AI Agent vs the Agentic AI ability to operate safely under TRiSM principles. This makes it difficult to compare different governance strategies or to track progress objectively. We encourage future work to develop evaluation frameworks, possibly extending from adversarial attack simulations [134] or “red-teaming” exercises [154] that can stress-test Agentic AI systems and score their resilience.

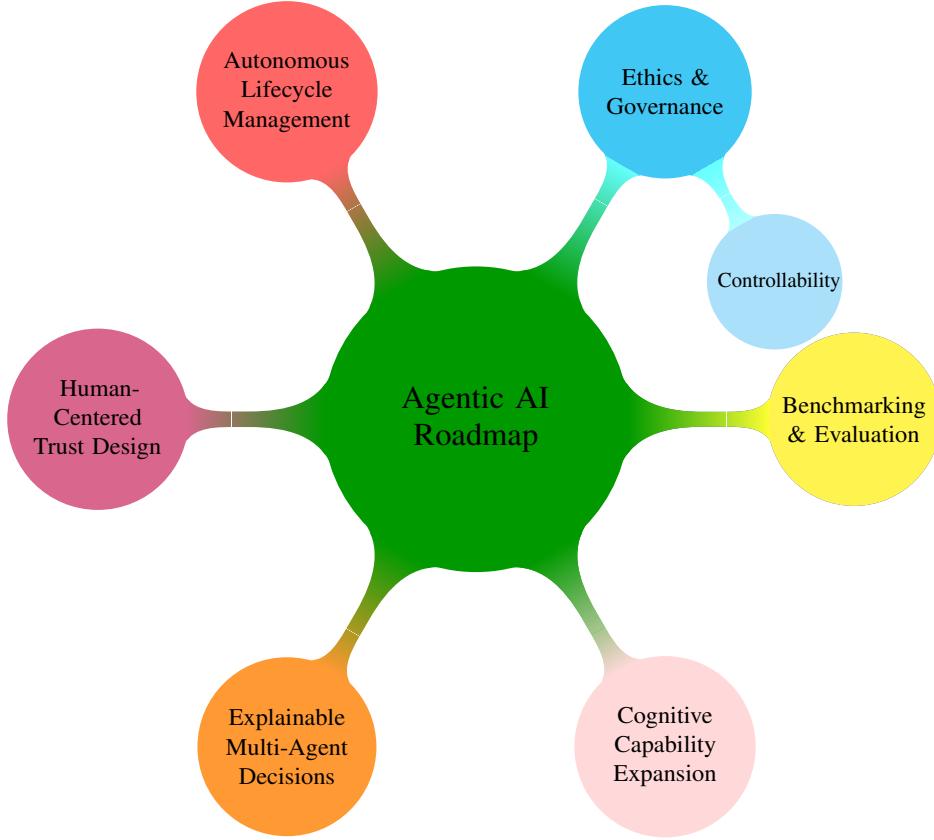


Figure 7: Agentic AI Roadmap: Mindmap Representation.

There is also a lack of real-world validation for many TRiSM-inspired controls. Much of the existing literature and tooling for LLM agent safety has been demonstrated in laboratory settings or on narrowly scoped tasks [65]. It remains uncertain how these governance mechanisms perform in complex, open-ended real-world environments. Another critical gap is adversarial learning. As we improve defenses, attackers will inevitably adapt [178]. Recent findings show that LLM-based systems remain vulnerable to attacks (for example, hidden prompt injections or subtle data poisoning) that can bypass superficial guardrails [38].

Beyond technical aspects, the organizational and human factors present limitations: implementing TRiSM requires interdisciplinary expertise (AI specialists, security experts, ethicists, legal advisors) and clear governance structures. Many organizations lack the necessary skill sets or frameworks, making TRiSM adoption superficial or inconsistent. Without a strong organizational commitment, even the best technical framework can falter.

### 9.5. Future Roadmap for Agentic AI TRiSM

Drawing from our findings and best practices from multiple disciplines, we propose several actionable directions for future research and implementation, as shown in Figure 7. These

recommendations span both technical system design improvements and governance-level policy initiatives:

The community should create open benchmarks and challenge environments to test AMAS governance. For instance, a suite of scenario-based tasks (with built-in threats and ethical dilemmas) could be used to evaluate how well a TRiSM-governed Agentic AI system that performs relative to one without such controls. This will enable direct comparisons and drive progress on measurable metrics of trust (e.g. frequency of prevented failures or fairness outcomes).

Future system design must anticipate a continually evolving threat landscape. Techniques from cybersecurity (e.g. adversarial training, AI model “penetration testing” [179], and formal verification) should be integrated into the LLM agent development pipeline. Cross-disciplinary collaboration with security experts can yield LLM-specific hardening methods, such as dynamic prompt anomaly detectors or robust tool APIs that constrain agent actions. Additionally, creating red-team/blue-team exercises for AMAS , akin to cyber wargames [180] , can help discover vulnerabilities in a controlled way before real adversaries do.

We encourage designing better interfaces and protocols for human oversight of Agentic AI systems. Borrowing from human-computer interaction [181] and cognitive engineering [182], researchers could devise dashboards that visualize an agent society’s state, flag important decisions, and allow intuitive human intervention (pausing agents, rolling back actions, etc.).

Policymakers and industry should collaborate to create regulatory sandboxes for multi-agent AI trials. These would be controlled environments where innovators can deploy Agentic AI systems under supervision, demonstrating TRiSM controls to regulators. Insights from such pilots can inform refinements in both technical standards and regulations. There is much to learn from other high-stakes domains. For example, the safety engineering field (e.g. aerospace, automotive) has mature practices for redundant controls and failure mode analysis; these could inspire analogous practices in AI agent design. Likewise, ethics boards in biomedical research provide a template for AI ethics committees that review agent behaviors and approve high-risk deployments. We advocate establishing multidisciplinary governance boards that include ethicists, legal experts, domain specialists, and community representatives to oversee significant deployments of autonomous AI.

A cross-cutting concept relevant to both Governance and Lifecycle Management, controllability captures the ability of humans and oversight systems to steer or override agentic AI. Mechanisms include stop/pause functions, rollback gates role-based oversight, and auditability of inter-agent actions. In practice, controllability is critical to ensuring accountability and preventing runaway behaviors in multi-agent workflows.

## 10. Conclusion

The rapid advancement of Agentic AI, driven by AMAS, marks a transformative shift in autonomous systems, enabling sophisticated collaboration, planning, and decision-making across high-stakes domains such as healthcare, finance, and public services. However, this evolution introduces complex challenges in trust, risk, and security management (TRiSM) that demand a tailored framework to ensure safe, transparent, and accountable operations. This review has explored these challenges, proposing a TRiSM framework structured around five pillars: Explainability, ModelOps, Security, Privacy and their lifecycle Governance, adapted for LLM-based AMAS. Key contributions include a novel risk taxonomy capturing unique threats like prompt injection, memory poisoning, and collusive failures, alongside two new metrics: the Component Synergy

Score (CSS) and Tool Utilization Efficacy (TUE), which provide practical tools for assessing inter-agent collaboration and tool efficiency. The review also highlights state-of-the-art techniques, such as LIME, SHAP, and decision provenance, to enhance explainability, while advocating for robust security measures like prompt hygiene, sandboxing, and homomorphic encryption to mitigate vulnerabilities. Future work should prioritize advancing adversarial robustness, refining governance protocols, and developing standardized benchmarks for trustworthiness and coordination. By aligning Agentic AI development with TRiSM principles, researchers and practitioners can foster systems that are not only innovative but also responsible, ensuring their safe and ethical integration into society.

### Acknowledgements

Funding for the research was provided through Horizon Europe project AIXpert: An agentic, multi-layer, GenAI-powered backbone to make an AI system explainable, accountable, and transparent (ID:101214389).

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### References

- [1] European Commission, Ai act, <https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>, accessed: 2025-09-10 (2025).
- [2] International Organization for Standardization, Iso/iec 42001:2023 – artificial intelligence management system (ai ms) – requirements, Tech. rep., ISO/IEC, available at <https://www.iso.org/standard/81230.html> (2023).
- [3] N. AI, Artificial intelligence risk management framework (ai rmf 1.0), URL: <https://nvlpubs.nist.gov/nistpubs/ai/nist.ai> (2023) 100–1.
- [4] A. Singla, A. Sukharevsky, L. Yee, M. Chui, B. Hall, The state of ai: How organizations are rewiring to capture value, <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>, accessed: 2025-09-10 (2025).
- [5] S. I. for Human-Centered Artificial Intelligence (HAI), 2025 ai index report, <https://hai.stanford.edu/ai-index/2025-ai-index-report>, accessed: 2025-09-10 (2025).
- [6] Gartner, Tackling trust, risk and security in ai models (ai trism), <https://www.gartner.com/en/articles/ai-trust-and-ai-risk>, accessed: 2025-09-10 (2024).
- [7] S. J. Russell, Rationality and intelligence, *Artificial intelligence* 94 (1-2) (1997) 57–77.
- [8] R. Sapkota, K. I. Roumeliotis, M. Karkee, Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenge, arXiv preprint arXiv:2505.10468 (2025).

- [9] Y. Yang, Q. Peng, J. Wang, W. Zhang, Multi-llm-agent systems: Techniques and business perspectives, arXiv preprint arXiv:2411.14033 (2024).
- [10] M. Y. Lim, Memory models for intelligent social companions, in: Human-computer interaction: The agency perspective, Springer, 2012, pp. 241–262.
- [11] X. Fang, J. Li, V. Mulchandani, J.-E. Kim, Trustworthy ai on safety, bias, and privacy: A survey (2025). arXiv:2502.10450.  
URL <https://arxiv.org/abs/2502.10450>
- [12] A. Gomstyn, A. Jonker, What is ai trism?, accessed: 2025-06-02 (Mar. 2025).  
URL <https://www.ibm.com/think/topics/ai-trism>
- [13] A. Habbal, M. K. Ali, M. A. Abuzaraida, Artificial intelligence trust, risk and security management (ai trism): Frameworks, applications, challenges and future research directions, Expert Systems with Applications 240 (2024) 122442.
- [14] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, X. Zhang, Large language model based multi-agents: A survey of progress and challenges (2024). arXiv: 2402.01680.  
URL <https://arxiv.org/abs/2402.01680>
- [15] S. Chen, Y. Liu, W. Han, W. Zhang, T. Liu, A survey on llm-based multi-agent system: Recent advances and new frontiers in application (2025). arXiv:2412.17481.  
URL <https://arxiv.org/abs/2412.17481>
- [16] B. Yan, X. Zhang, L. Zhang, L. Zhou, D. Miao, C. Li, Beyond self-talk: A communication-centric survey of llm-based multi-agent systems (2025). arXiv:2502.14321.  
URL <https://arxiv.org/abs/2502.14321>
- [17] K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O’Sullivan, H. D. Nguyen, Multi-agent collaboration mechanisms: A survey of llms (2025). arXiv:2501.06322.  
URL <https://arxiv.org/abs/2501.06322>
- [18] Y.-C. Lin, K.-C. Chen, Z.-Y. Li, T.-H. Wu, T.-H. Wu, K.-Y. Chen, H. yi Lee, Y.-N. Chen, Creativity in llm-based multi-agent systems: A survey (2025). arXiv:2505.21116.  
URL <https://arxiv.org/abs/2505.21116>
- [19] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, et al., The rise and potential of large language model based agents: A survey, Science China Information Sciences 68 (2) (2025) 121101.
- [20] J. Luo, W. Zhang, Y. Yuan, Y. Zhao, J. Yang, Y. Gu, B. Wu, B. Chen, Z. Qiao, Q. Long, et al., Large language model agent: A survey on methodology, applications and challenges, arXiv preprint arXiv:2503.21460 (2025).
- [21] H. P. Zou, W.-C. Huang, Y. Wu, Y. Chen, C. Miao, H. Nguyen, Y. Zhou, W. Zhang, L. Fang, L. He, Y. Li, D. Li, R. Jiang, X. Liu, P. S. Yu, Llm-based human-agent collaboration and interaction systems: A survey (2025). arXiv:2505.00753.  
URL <https://arxiv.org/abs/2505.00753>

- [22] W. Wang, Z. Ma, Z. Wang, C. Wu, J. Ji, W. Chen, X. Li, Y. Yuan, A survey of llm-based agents in medicine: How far are we from baymax? (2025). arXiv:2502.11211. URL <https://arxiv.org/abs/2502.11211>
- [23] D. Rein, J. Becker, A. Deng, S. Nix, C. Canal, D. O'Connel, P. Arnott, R. Bloom, T. Broadley, K. Garcia, B. Goodrich, M. Hasin, S. Jawhar, M. Kinniment, T. Kwa, A. Lajko, N. Rush, L. J. K. Sato, S. V. Arx, B. West, L. Chan, E. Barnes, Hcast: Human-calibrated autonomy software tasks (2025). arXiv:2503.17354. URL <https://arxiv.org/abs/2503.17354>
- [24] C. Castelfranchi, Modelling social action for ai agents, Artificial intelligence 103 (1-2) (1998) 157–182.
- [25] I. Chiu, L. Shu, Biomimetic design through natural language analysis to facilitate cross-domain information retrieval, Ai Edam 21 (1) (2007) 45–59.
- [26] D. Tanasa, B. Troussé, Advanced data preprocessing for intersites web usage mining, IEEE Intelligent Systems 19 (2) (2005) 59–65.
- [27] D. Griol, J. Carbó, J. M. Molina, An automatic dialog simulation technique to develop and evaluate interactive conversational agents, Applied Artificial Intelligence 27 (9) (2013) 759–780.
- [28] M. R. Genesereth, N. J. Nilsson, Logical foundations of artificial intelligence, Morgan Kaufmann, 2012.
- [29] D. B. Acharya, K. Kuppan, B. Divya, Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey, IEEE Access (2025).
- [30] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong, et al., Chatdev: Communicative agents for software development, arXiv preprint arXiv:2307.07924 (2023).
- [31] B. P. Zeigler, Object-oriented simulation with hierarchical, modular models: intelligent agents and endomorphic systems, Academic press, 2014.
- [32] International Organization for Standardization, Iso/iec tr 24029-1:2021 – artificial intelligence (ai) – assessment of the robustness of neural networks – part 1: Overview, Tech. rep., ISO/IEC, available at <https://www.iso.org/standard/77608.html> (2021).
- [33] A. Borji, Stochastic parrots or intelligent systems? a perspective on true depth of understanding in llms, A Perspective on True Depth of Understanding in LLMs (July 11, 2023) (2023).
- [34] European Union, General Data Protection Regulation (GDPR) – Article 25: Data protection by design and by default, <https://gdpr-info.eu/art-25-gdpr/>, accessed: 2025-06-03 (2016).
- [35] U.S. Department of Health and Human Services, HIPAA Privacy Rule – 45 CFR Part 164: Security and Privacy Protections for Health Information, <https://www.ecfr.gov/current/title-45.subtitle-A/subchapter-C/part-164>, accessed: 2025-06-03 (2003).

- [36] M. Kosinski, A. Forrest, What is a prompt injection attack?, IBM ThinkAccessed: 2025-06-03 (March 2024).  
URL <https://www.ibm.com/think/topics/prompt-injection>
- [37] D. Lee, M. Tiwari, Prompt infection: Llm-to-llm prompt injection within multi-agent systems (2024). arXiv:2410.07283.  
URL <https://arxiv.org/abs/2410.07283>
- [38] S. Raza, R. Qureshi, M. Lotif, A. Chadha, D. Pandya, C. Emmanouilidis, Just as humans need vaccines, so do models: Model immunization to combat falsehoods (2025). arXiv: 2505.17870.  
URL <https://arxiv.org/abs/2505.17870>
- [39] J. Chen, R. Lu, Ai agents are here. so are the threats., Unit 42 BlogAccessed: 2025-06-03 (May 2025).  
URL <https://unit42.paloaltonetworks.com/agentic-ai-threats>
- [40] R. Ko, J. Jeong, S. Zheng, C. Xiao, T. Kim, M. Onizuka, W. Shin, Seven security challenges that must be solved in cross-domain multi-agent llm systems (2025). arXiv:2505.23847.  
URL <https://arxiv.org/abs/2505.23847>
- [41] S. Raza, A. Shaban-Nejad, E. Dolatabadi, H. Mamiya, Exploring bias and prediction metrics to characterise the fairness of machine learning for equity-centered public health decision-making: A narrative review, IEEE Access (2024).
- [42] Securiti, What is AI TRiSM and why it's essential in the era of genai, accessed: 2025-06-03 (May 2025).  
URL <https://securiti.ai/what-is-ai-trism/>
- [43] A. Boland, G. Cherry, R. Dickson, Doing a systematic review: a student s guide (2017).
- [44] S. Keele, et al., Guidelines for performing systematic literature reviews in software engineering, Tech. rep., Technical report, ver. 2.3 ebse technical report. ebse (2007).
- [45] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele University 33 (2004) (2004) 1–26.
- [46] N. Muscettola, P. P. Nayak, B. Pell, B. C. Williams, Remote agent: To boldly go where no ai system has gone before, Artificial intelligence 103 (1-2) (1998) 5–47.
- [47] M. Hannebauer, From formal workflow models to intelligent agents, in: Proceedings of the AAAI-99 Workshop on Agent Based Systems in the Business Context, 1999, pp. 19–24.
- [48] P. J. Govaerts, B. Vaerenberg, G. De Ceulaer, K. Daemers, C. De Beukelaer, K. Schauwers, Development of a software tool using deterministic logic for the optimization of cochlear implant processor programming, Otology & Neurotology 31 (6) (2010) 908–918.
- [49] P. Langley, J. E. Laird, S. Rogers, Cognitive architectures: Research issues and challenges, Cognitive Systems Research 7 (1) (2006) 1–24. doi:10.1016/j.cogsys.2005.07.001.
- [50] J. E. Laird, A. Newell, P. S. Rosenbloom, Soar: An architecture for general intelligence, Artificial intelligence 33 (1) (1987) 1–64.

- [51] J. R. Anderson, Rules of the Mind, Lawrence Erlbaum Associates, 1993.
- [52] P. Langley, E. P. Katz, Spatial representation and reasoning in an architecture for embodied agents, *Spatial Cognition & Computation* 25 (1) (2025) 15–37.
- [53] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, et al., Metagpt: Meta programming for multi-agent collaborative framework, arXiv preprint arXiv:2308.00352 3 (4) (2023) 6.
- [54] U. M. Borghoff, P. Bottone, R. Pareschi, Human-artificial interaction in the age of agentic ai: a system-theoretical approach, *Frontiers in Human Dynamics* 7 (2025) 1579166.
- [55] J. Schneider, Generative to agentic ai: Survey, conceptualization, and challenges, arXiv preprint arXiv:2504.18875 (2025).
- [56] C. S. de Witt, Open challenges in multi-agent security: Towards secure systems of interacting ai agents, arXiv preprint arXiv:2505.02077 (2025).
- [57] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, et al., Autogen: Enabling next-gen llm applications via multi-agent conversation, arXiv preprint arXiv:2308.08155 (2023).
- [58] S. Qiao, R. Fang, N. Zhang, Y. Zhu, X. Chen, S. Deng, Y. Jiang, P. Xie, F. Huang, H. Chen, Agent planning with world knowledge model, *Advances in Neural Information Processing Systems* 37 (2024) 114843–114871.
- [59] K. Wang, G. Zhang, Z. Zhou, J. Wu, M. Yu, S. Zhao, C. Yin, J. Fu, Y. Yan, H. Luo, et al., A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment, arXiv preprint arXiv:2504.15585 (2025).
- [60] A. Speight, Build ai agents end-to-end in vs code, <https://techcommunity.microsoft.com/blog/azuredevcommunityblog/build-ai-agents-end-to-end-in-vs-code/4418117>, microsoft Developer Community Blog (May 2025).  
URL <https://techcommunity.microsoft.com/blog/azuredevcommunityblog/build-ai-agents-end-to-end-in-vs-code/4418117>
- [61] D. Weyns, A. Helleboogh, T. Holvoet, M. Schumacher, The agent environment in multi-agent systems: A middleware perspective, *Multiagent and Grid Systems* 5 (1) (2009) 93–108.
- [62] M. Corporation, Langgraph: Agent orchestration framework for llms, <https://www.langchain.com/langgraph>, accessed: 2025-06-03 (2024).  
URL <https://www.langchain.com/langgraph>
- [63] J. Cranshaw, E. Elwany, T. Newman, R. Kocielnik, B. Yu, S. Soni, J. Teevan, A. Monroy-Hernández, Calendar. help: Designing a workflow-based scheduling agent with humans in the loop, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 2017, pp. 2382–2393.

- [64] J. Kaur, Building trust with ai trism: Managing risks in the era of agentic ai, Akira AI BlogAccessed: 2025-06-03 (December 2024).  
URL <https://www.akira.ai/blog/ai-trism-with-ai-agents>
- [65] CharliAI Inc., Trust, Risk and Security Management in AI Systems, <https://charliai.com/wp-content/uploads/2024/01/TRiSM-Position-Paper-January-2024.pdf>, accessed: 2025-06-03 (Jan. 2024).
- [66] B. Zhong, H. Cao, M. Zamani, M. Caccamo, Towards safe ai: Sandboxing dnns-based controllers in stochastic games, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 15340–15349.
- [67] G. Feretzakis, K. Papaspyridis, A. Gkoulalas-Divanis, V. S. Verykios, Privacy-preserving techniques in generative ai and large language models: A narrative review, *Information* 15 (11) (2024) 697.
- [68] M. Singh, A. Alabdulkarim, G. Mansi, M. O. Riedl, Explainable reinforcement learning agents using world models, arXiv preprint arXiv:2505.08073 (2025).
- [69] E. Saravia, Llm agents, <https://www.promptingguide.ai/research/llm-agents>, accessed: 2025-06-03 (2024).
- [70] S. Gravitas, Autogpt, <https://github.com/Significant-Gravitas/AutoGPT>, accessed: 2025-06-03 (2023).
- [71] Y. Nakajima, Babyagi (archived version), [https://github.com/yoheinakajima/babyagi\\_archive](https://github.com/yoheinakajima/babyagi_archive), accessed: 2025-06-03 (2024).
- [72] A. Osika, gpt-engineer, <https://github.com/AntonOsika/gpt-engineer>, accessed: 2025-06-03 (2023).
- [73] E. Saravia, Prompt engineering guide, <https://www.promptingguide.ai/>, accessed: 2025-06-03 (2024).
- [74] A. Havrilla, S. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, M. Zhuravinskyi, E. Hambro, R. Raileanu, Glore: When, where, and how to improve llm reasoning via global and local refinements, arXiv preprint arXiv:2402.10963 (2024).
- [75] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models (2023). arXiv:2210.03629.  
URL <https://arxiv.org/abs/2210.03629>
- [76] H. Yang, S. Yue, Y. He, Auto-gpt for online decision making: Benchmarks and additional opinions, arXiv preprint arXiv:2306.02224 (2023).
- [77] A. Osika, gpt-engineer: Cli platform to experiment with codegen, <https://github.com/AntonOsika/gpt-engineer>, mIT License (2023).
- [78] J. Wang, Z. Duan, Agent ai with langgraph: A modular framework for enhancing machine translation using large language models, arXiv preprint arXiv:2412.03801 (2024).  
URL <https://arxiv.org/abs/2412.03801>

- [79] E. Karpas, O. Abend, Y. Belinkov, B. Lenz, O. Lieber, N. Ratner, Y. Shoham, H. Bata, Y. Levine, K. Leyton-Brown, D. Muhlgay, N. Rozen, E. Schwartz, G. Shachaf, S. Shalev-Shwartz, A. Shashua, M. Tenenholz, Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning (2022). [arXiv:2205.00445](https://arxiv.org/abs/2205.00445)  
 URL <https://arxiv.org/abs/2205.00445>
- [80] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, S. Yao, Reflexion: Language agents with verbal reinforcement learning (2023). [arXiv:2303.11366](https://arxiv.org/abs/2303.11366)  
 URL <https://arxiv.org/abs/2303.11366>
- [81] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, C. Wu, J. Schmidhuber, MetaGPT: Meta programming for a multi-agent collaborative framework, in: The Twelfth International Conference on Learning Representations, 2024.  
 URL <https://openreview.net/forum?id=VtmBAGCN7o>
- [82] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar, Voyager: An open-ended embodied agent with large language models (2023). [arXiv:2305.16291](https://arxiv.org/abs/2305.16291)  
 URL <https://arxiv.org/abs/2305.16291>
- [83] H. He, W. Yao, K. Ma, W. Yu, Y. Dai, H. Zhang, Z. Lan, D. Yu, Webvoyager: Building an end-to-end web agent with large multimodal models, arXiv preprint arXiv:2401.13919 (2024).
- [84] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, Y. Zhuang, Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face, Advances in Neural Information Processing Systems 36 (2023) 38154–38180.
- [85] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbulin, B. Ghanem, Camel: Communicative agents for "mind" exploration of large language model society, in: Thirty-seventh Conference on Neural Information Processing Systems, 2023.
- [86] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong, J. Xu, D. Li, Z. Liu, M. Sun, Chatdev: Communicative agents for software development (2024). [arXiv:2307.07924](https://arxiv.org/abs/2307.07924)  
 URL <https://arxiv.org/abs/2307.07924>
- [87] J. Moura, contributors, Crewai: Framework for orchestrating role-playing, autonomous ai agents, <https://github.com/crewAIInc/crewAI>, mIT License (2023).
- [88] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C. Qian, C.-M. Chan, Y. Qin, Y. Lu, R. Xie, et al., Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents, arXiv preprint arXiv:2308.10848 (2023).
- [89] T. Xie, F. Zhou, Z. Cheng, P. Shi, L. Weng, Y. Liu, T. J. Hua, J. Zhao, Q. Liu, C. Liu, L. Z. Liu, Y. Xu, H. Su, D. Shin, C. Xiong, T. Yu, Openagents: An open platform for language agents in the wild (2023). [arXiv:2310.10634](https://arxiv.org/abs/2310.10634)  
 URL <https://arxiv.org/abs/2310.10634>

- [90] TransformerOptimus, contributors, Superagi: A dev-first open source autonomous ai agent framework, <https://github.com/TransformerOptimus/SuperAGI>, mIT License (2023).
- [91] Microsoft, Semantic kernel agent framework, <https://learn.microsoft.com/en-us/semantic-kernel/frameworks/agent/>, accessed: 2025-09-10 (2025).
- [92] OpenAI, Swarm: Lightweight multi-agent orchestration framework, <https://github.com/openai/swarm>, accessed: 2025-09-10 (2024).
- [93] OpenAI, Openai agents sdk (python), <https://github.com/openai/openai-agents-python>, accessed: 2025-09-10 (2025).
- [94] A. S. Project, Strands agents sdk (python), <https://github.com/strands-agents/sdk-python>, accessed: 2025-09-10 (2025).
- [95] LlamaIndex, Llamaindex agents documentation, [https://docs.llamaindex.ai/en/stable/use\\_cases/agents/](https://docs.llamaindex.ai/en/stable/use_cases/agents/), accessed: 2025-09-10 (2024).
- [96] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, T. Scialom, Toolformer: Language models can teach themselves to use tools (2023). [arXiv:2302.04761](https://arxiv.org/abs/2302.04761)  
URL <https://arxiv.org/abs/2302.04761>
- [97] K. Liu, X. Lei, Z. Wang, P. Li, Y. Liu, Agent-environment alignment via automated interface generation, arXiv preprint arXiv:2505.21055 (2025).
- [98] Microsoft Open Source, Agents — autogen agentchat user guide, <https://github.io/autogen/stable//user-guide/agentchat-user-guide/tutorial/agents.html>, accessed: 2025-06-02 (2024).
- [99] H. Miao, F. Ma, R. Quan, K. Zhan, Y. Yang, Autonomous llm-enhanced adversarial attack for text-to-motion, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 39, 2025, pp. 6144–6152.
- [100] K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O’Sullivan, H. D. Nguyen, Multi-agent collaboration mechanisms: A survey of llms, arXiv preprint arXiv:2501.06322 (2025).
- [101] Z. Feng, R. Xue, L. Yuan, Y. Yu, N. Ding, M. Liu, B. Gao, J. Sun, G. Wang, Multi-agent embodied ai: Advances and future directions, arXiv preprint arXiv:2505.05108 (2025).
- [102] H. Dawid, P. Harting, H. Wang, Z. Wang, J. Yi, Agentic workflows for economic research: Design and implementation, arXiv preprint arXiv:2504.09736 (2025).
- [103] M. M. Karim, D. H. Van, S. Khan, Q. Qu, Y. Kholodov, Ai agents meet blockchain: A survey on secure and scalable collaboration for multi-agents, Future Internet 17 (2) (2025) 57.
- [104] L. Euler, Hacking auto-gpt and escaping its docker container, <https://positive-security.blog/auto-gpt-rce>, accessed 27 Jun 2025 (2023).

- [105] Invicti Security, Prompt injection attacks on applications that use llms, Tech. rep., Invicti Security, e-book describing prompt injection attack types, real-world examples, and mitigation strategies (2024).  
URL <https://www.invicti.com/white-papers/prompt-injection-attacks-on-llm-applications-ebook>
- [106] V. Strobel, M. Dorigo, M. Fritz, Llm2swarm: Robot swarms that responsively reason, plan, and collaborate through llms, in: NeurIPS 2024 Workshop on Open-World Agents (OWA-2024), 2024.
- [107] A. Ligot, M. Birattari, On using simulation to predict the performance of robot swarms, *Scientific Data* 9 (1) (2022) 788.
- [108] C. Atkins, B. Z. H. Zhao, H. J. Asghar, I. Wood, M. A. Kaafar, Those aren't your memories, they're somebody else's: Seeding misinformation in chat bot memories, arXiv preprint arXiv:2304.05371 (2023).  
URL <https://arxiv.org/abs/2304.05371>
- [109] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts (2023). arXiv:2307.03172.  
URL <https://arxiv.org/abs/2307.03172>
- [110] A. Litan, Ai trust and ai risk: Tackling trust, risk and security in ai models, GartnerAccessed: 2025-06-03 (December 2024).  
URL <https://www.gartner.com/en/articles/ai-trust-and-ai-risk>
- [111] M. Sanwal, Layered chain-of-thought prompting for multi-agent llm systems: A comprehensive approach to explainable large language models, arXiv preprint arXiv:2501.18645 (2025).
- [112] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
- [113] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., 2017, pp. 4765–4774.  
URL [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf)
- [114] J. Tupayachi, H. Xu, O. A. Omitaomu, M. C. Camur, A. Sharmin, X. Li, Towards next-generation urban decision support systems through ai-powered construction of scientific ontology using large language models—a case in optimizing intermodal freight transportation, *Smart Cities* 7 (5) (2024) 2392–2421.
- [115] J. Bai, K. F. Lee, M. Hofmeister, S. Mosbach, J. Akroyd, M. Kraft, A derived information framework for a dynamic knowledge graph and its application to smart cities, *Future Generation Computer Systems* 152 (2024) 112–126.
- [116] F. Tian, A. Luo, J. Du, X. Xian, R. Specht, G. Wang, X. Bi, J. Zhou, J. Srinivasa, A. Kundu, et al., An outlook on the opportunities and challenges of multi-agent ai systems, arXiv preprint arXiv:2505.18397 (2025).

- [117] S. Liu, H. Miao, Z. Li, M. Olson, V. Pascucci, P.-T. Bremer, Ava: Towards autonomous visualization agents through visual perception-driven decision-making, in: Computer Graphics Forum, Vol. 43, Wiley Online Library, 2024, p. e15093.
- [118] K. Hu, K. Xu, Q. Xia, M. Li, Z. Song, L. Song, N. Sun, An overview: Attention mechanisms in multi-agent reinforcement learning, Neurocomputing (2024) 128015.
- [119] R. Tomsett, D. Braines, D. Harborne, A. Preece, S. Chakraborty, Interpretable to whom? a role-based model for analyzing interpretable machine learning systems, arXiv preprint arXiv:1806.07552 (2018).
- [120] A. Ghafarollahi, M. J. Buehler, Sciagents: Automating scientific discovery through multi-agent intelligent graph reasoning, arXiv preprint arXiv:2409.05556 (2024).
- [121] H. Gu, M. Schreyer, K. Moffitt, M. Vasarhelyi, Artificial intelligence co-piloted auditing, International Journal of Accounting Information Systems 54 (2024) 100698.
- [122] J. H. Chin, P. Zhang, Y. X. Cheong, J. Pan, Automating security audit using large language model based agent: An exploration experiment, arXiv preprint arXiv:2505.10732 (2025).
- [123] S. Raza, S. Ghuge, C. Ding, D. Pandya, FAIR Enough: How Can We Develop and Assess a FAIR-Compliant Dataset for Large Language Models' Training?, arXiv preprint arXiv:2401.11033 (2024).
- [124] Z. Ke, A. Xu, Y. Ming, X.-P. Nguyen, C. Xiong, S. Joty, Meta-design matters: A self-design multi-agent system, arXiv preprint arXiv:2505.14996 (2025).
- [125] European Commission, AI Act | Shaping Europe's digital future, accessed: 2025-06-03 (2024).  
URL <https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>
- [126] N. AI, Artificial intelligence risk management framework: Generative artificial intelligence profile, NIST Trustworthy and Responsible AI Gaithersburg, MD, USA (2024).
- [127] Oecd.ai catalogue of tools & metrics for trustworthy ai, <https://oecd.ai/en/>, oECD .AI is a live platform promoting trustworthy, human-centric AI; accessed 24 Jun 2025.
- [128] Q. Wang, T. Wang, Z. Tang, Q. Li, N. Chen, J. Liang, B. He, Megaagent: A large-scale autonomous llm-based multi-agent system without predefined sops, in: Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 4998–5036.
- [129] Information technology — artificial intelligence (ai) — ai system impact assessment, published May 2025; reference number 42005; International standard (May 2025).
- [130] Information technology — artificial intelligence — guidance on risk management (Feb. 2023).
- [131] K. Lefevre, C. Arora, K. Lee, A. Zaslavsky, M. R. Bouadjenek, A. Hassani, I. Razzak, Modelops for enhanced decision-making and governance in emergency control rooms, Environment Systems and Decisions 42 (3) (2022) 402–416. doi:10.1007/s10669-022-09855-1.  
URL <https://doi.org/10.1007/s10669-022-09855-1>

- [132] S. Sinha, The rise of modelops: What comes after mlops?, Brim Labs Blog (Jun. 2025).  
URL <https://brimlabs.ai/blog/the-rise-of-modelops-what-comes-after-mlops/>
- [133] L. Beurer-Kellner, B. Buesser, A.-M. Crețu, E. Debenedetti, D. Dobos, D. Fabian, M. Fischer, D. Froelicher, K. Grosse, D. Naeff, et al., Design patterns for securing llm agents against prompt injections, arXiv preprint arXiv:2506.08837 (2025).
- [134] M. Standen, J. Kim, C. Szabo, Adversarial machine learning attacks and defences in multi-agent reinforcement learning, ACM Computing Surveys 57 (5) (2025) 1–35.
- [135] H. Chen, S. Li, J. Fan, A. Duan, C. Yang, D. Navarro-Alarcon, P. Zheng, Human-in-the-loop robot learning for smart manufacturing: A human-centric perspective, IEEE Transactions on Automation Science and Engineering (2025).
- [136] A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, P. Schwaller, Chemcrow: Augmenting large-language models with chemistry tools, arXiv preprint arXiv:2304.05376 (2023).
- [137] Open Worldwide Application Security Project (OWASP), OWASP Top 10 for Large Language Model Applications, <https://owasp.org/www-project-top-10-for-large-language-model-applications/>, accessed: 2025-06-03 (2024).
- [138] OWASP Agentic Security Initiative, Agentic ai – threats and mitigations, Tech. rep., Open Worldwide Application Security Project (OWASP), accessed: 2025-06-03 (February 2025).  
URL [https://hal.science/hal-04985337v1/file/Agentic-AI-Threats-and-Mitigations\\_v1.0.1.pdf](https://hal.science/hal-04985337v1/file/Agentic-AI-Threats-and-Mitigations_v1.0.1.pdf)
- [139] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of Cryptography Conference, Springer, 2006, pp. 265–284.
- [140] L. Sweeney, k-anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (05) (2002) 557–570.
- [141] Y. Lindell, B. Pinkas, Secure two-party computation via cut-and-choose oblivious transfer, in: Advances in Cryptology—EUROCRYPT 2007, Springer, 2007, pp. 329–346.
- [142] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proceedings of the 41st annual ACM symposium on Theory of computing, ACM, 2009, pp. 169–178.
- [143] V. Costan, S. Devadas, Intel sgx explained, in: IACR Cryptology ePrint Archive, Vol. 2016, 2016, p. 86.
- [144] California Department of Justice, Office of the Attorney General, California consumer privacy act (ccpa), Online, updated March 13, 2024, accessed July 9, 2025 (2024).  
URL <https://oag.ca.gov/privacy/ccpa>
- [145] M. Yossef, M. Noureldin, A. Alqabbany, Explainable artificial intelligence framework for frp composites design, Composite Structures 341 (2024) 118190.
- [146] A. Rosenfeld, A. Richardson, Explainability in human–agent systems, Autonomous agents and multi-agent systems 33 (2019) 673–705.

- [147] M. C. Ozgun, J. Pei, K. Hindriks, L. Donatelli, Q. Liu, X. Sun, J. Wang, Trustworthy ai psychotherapy: Multi-agent llm workflow for counseling and explainable mental disorder diagnosis, arXiv preprint arXiv:2508.11398 (2025).
- [148] F. P.-W. Lo, J. Qiu, Z. Wang, H. Yu, Y. Chen, G. Zhang, B. Lo, Ai hiring with llms: A context-aware and explainable multi-agent framework for resume screening, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 4184–4193.
- [149] K. Nagpal, D. Dong, J.-B. Bouvier, N. Mehr, Leveraging large language models for effective and explainable multi-agent credit assignment, arXiv preprint arXiv:2502.16863 (2025).
- [150] A. Bilal, D. Ebert, B. Lin, Llms for explainable ai: A comprehensive survey, arXiv preprint arXiv:2504.00125 (2025).
- [151] B. Gyevnar, C. Wang, C. G. Lucas, S. B. Cohen, S. V. Albrecht, Causal explanations for sequential decision-making in multi-agent systems, arXiv preprint arXiv:2302.10809 (2023).
- [152] I. Barberá, Ai privacy risks & mitigations—large language models (llms), European Data Protection Board. Available online: <https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf> (accessed on 12 June 2025) (2025).
- [153] LangChain Team, Langchain and langgraph: Comparing function and tool calling capabilities, LangChain BlogAccessed: 2025-06-03 (April 2024).  
URL <https://www.langchain.com/>
- [154] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse, A. Jones, S. Bowman, A. Chen, T. Conerly, N. DasSarma, D. Drain, N. Elhage, S. El-Showk, S. Fort, Z. Hatfield-Dodds, T. Henighan, D. Hernandez, T. Hume, J. Jacobson, S. Johnston, S. Kravec, C. Olsson, S. Ringer, E. Tran-Johnson, D. Amodei, T. Brown, N. Joseph, S. McCandlish, C. Olah, J. Kaplan, J. Clark, Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned, arXiv:2209.07858 [cs] (Nov. 2022).  
URL <http://arxiv.org/abs/2209.07858>
- [155] P. J. Phillips, P. J. Phillips, C. A. Hahn, P. C. Fontana, A. N. Yates, K. Greene, D. A. Broniatowski, M. A. Przybocki, Four principles of explainable artificial intelligence (2021).
- [156] F. A. Thiers, Z. Harned, The emerging role of iso 42001 certification in fostering the deployment of responsible generative ai healthcare solutions, Technology (NIST) 55 (58) (2024) 59.
- [157] J. A. Khan, Role-based access control (rbac) and attribute-based access control (abac), in: Improving security, privacy, and trust in cloud computing, IGI Global Scientific Publishing, 2024, pp. 113–126.
- [158] C. Novelli, F. Casolari, A. Rotolo, M. Taddeo, L. Floridi, Ai risk assessment: a scenario-based, proportional methodology for the ai act, Digital Society 3 (1) (2024) 13.
- [159] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu, G-eval: Nlg evaluation using gpt-4 with better human alignment (2023). arXiv:2303.16634.  
URL <https://arxiv.org/abs/2303.16634>

- [160] C. Agarwal, D. Ley, S. Krishna, E. Saxena, M. Pawelczyk, N. Johnson, I. Puri, M. Zitnik, H. Lakkaraju, Openxai: Towards a transparent evaluation of model explanations (2024). arXiv:2206.11104.  
URL <https://arxiv.org/abs/2206.11104>
- [161] K. Rodden, H. Hutchinson, X. Fu, Measuring the user experience on a large scale: user-centered metrics for web applications, in: Proceedings of the SIGCHI conference on human factors in computing systems, 2010, pp. 2395–2398.
- [162] R. Khan, S. Sarkar, S. K. Mahata, E. Jose, Security threats in agentic ai system, arXiv preprint arXiv:2410.14728 (2024).
- [163] Y. Wang, Y. Pan, Q. Zhao, Y. Deng, Z. Su, L. Du, T. H. Luan, Large model agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends, arXiv preprint arXiv:2409.14457 (2024).
- [164] B. Zhang, Y. Tan, Y. Shen, A. Salem, M. Backes, S. Zannettou, Y. Zhang, Breaking agents: Compromising autonomous llm agents through malfunction amplification, arXiv preprint arXiv:2407.20859 (2024).
- [165] Y. Zhen, D. Tian, X. Fu, C. Hu, A novel malware detection method based on audit logs and graph neural network, Engineering Applications of Artificial Intelligence 152 (2025) 110524.
- [166] K. Lazaros, D. E. Koumadorakis, A. G. Vrahatis, S. Kotsiantis, Federated learning: Navigating the landscape of collaborative intelligence, Electronics 13 (23) (2024) 4744.
- [167] M. V. Luzón, N. Rodríguez-Barroso, A. Argente-Garrido, D. Jiménez-López, J. M. Moyano, J. Del Ser, W. Ding, F. Herrera, A tutorial on federated learning from theory to practice: Foundations, software frameworks, exemplary use cases, and selected trends, IEEE/CAA Journal of Automatica Sinica 11 (4) (2024) 824–850.
- [168] D. Evans, V. Kolesnikov, M. Rosulek, et al., A pragmatic introduction to secure multi-party computation, Foundations and Trends® in Privacy and Security 2 (2-3) (2018) 70–246.
- [169] A. Acar, H. Aksu, A. S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: Theory and implementation, ACM Computing Surveys (Csur) 51 (4) (2018) 1–35.
- [170] P. Schaar, Privacy by design, Identity in the Information Society 3 (2) (2010) 267–274.
- [171] E. Tabassi, Artificial intelligence risk management framework (ai rmf 1.0) (January 26 2023). doi:10.6028/NIST.AI.100-1.  
URL [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=936225](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936225)
- [172] G. AI, Guardrails AI | Your Enterprise AI needs Guardrails — guardrailsai.com (Feb. 2024).  
URL <https://www.guardrailsai.com/docs/>
- [173] S. Raza, R. Qureshi, A. Zahid, J. Fioresi, F. Sadak, M. Saeed, R. Sapkota, A. Jain, A. Zafar, M. U. Hassan, et al., Who is responsible? the data, models, users or regulations? a comprehensive survey on responsible generative ai for a sustainable future, arXiv preprint arXiv:2502.08650 (2025).

- [174] C. Chen, X. Gong, Z. Liu, W. Jiang, S. Q. Goh, K.-Y. Lam, Trustworthy, responsible, and safe ai: A comprehensive architectural framework for ai safety with challenges and mitigations, arXiv preprint arXiv:2408.12935 (2024).
- [175] A. Vayani, D. Dissanayake, H. Watawana, N. Ahsan, N. Sasikumar, O. Thawakar, H. B. Ademtew, Y. Hmaiti, A. Kumar, K. Kukreja, et al., All languages matter: Evaluating lmms on culturally diverse 100 languages, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 19565–19575.
- [176] S. Raza, A. Narayanan, V. R. Khazaie, A. Vayani, M. S. Chettiar, A. Singh, M. Shah, D. Pandya, Humanibench: A human-centric framework for large multimodal models evaluation, arXiv preprint arXiv:2505.11454 (2025).
- [177] R. Parasuraman, D. H. Manzey, Complacency and bias in human use of automation: An attentional integration, *Human Factors* 52 (3) (2010) 381–410. doi:10.1177/0018720810376055.  
URL <https://journals.sagepub.com/doi/10.1177/0018720810376055>
- [178] S. Raza, A. Vayani, A. Jain, A. Narayanan, V. R. Khazaie, S. R. Bashir, E. Dolatabadi, G. Uddin, C. Emmanouilidis, R. Qureshi, et al., Vldbench: Vision language models disinformation detection benchmark, arXiv preprint arXiv:2502.11361 (2025).
- [179] A. G. Bacudio, X. Yuan, B.-T. B. Chu, M. Jones, An overview of penetration testing, *International Journal of Network Security & Its Applications (IJNSA)* 3 (6) (2011) 19–38. doi:10.5121/ijnsa.2011.3602.  
URL <https://airccse.org/journal/nsa/1111nsa02.pdf>
- [180] J. Schneider, Wargaming cyber security, War on the Rocks (September 2020).  
URL <https://warontherocks.com/2020/09/wargaming-cyber-security/>
- [181] S. Amershi, D. S. Weld, M. Vorvoreanu, A. Journey, B. Nushi, P. Collisson, J. Suh, S. T. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, E. Horvitz, Guidelines for human-ai interaction, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, ACM, 2019.  
URL <https://dl.acm.org/doi/10.1145/3290605.3300233>
- [182] H. Hexmoor, J. Lammens, G. Caicedo, S. C. Shapiro, Behaviour based AI, cognitive processes, and emergent behaviors in autonomous agents, Vol. 1, WIT Press, 2025.

Table A.1: Key terminologies for LLM-based agentic AI systems.

Term	Definition
Agentic AI System	A multi-agent architecture powered by large language models (LLMs), where autonomous agents collaborate, plan, and execute tasks over extended horizons with persistent memory and dynamic role assignment.
Autonomy Model	The mechanism by which an agent decides and acts without direct human intervention, often using goal-driven planning and chain-of-thought reasoning.
Chain of Thought (CoT)	A prompting strategy in which an LLM generates intermediate reasoning steps before producing a final answer or action, enhancing interpretability and multi-step planning.
Counterfactual Analysis	An interpretability technique that examines how altering certain inputs or agent contributions would change the overall system outcome, revealing causal dependencies among agents.
Explainability	The capacity of an AI system (or individual agent) to produce human-understandable justifications or rationales for its decisions and actions, often via LIME, SHAP, or decision provenance graphs.
Foundation Model (LLM)	A pretrained large language model (e.g., GPT-4, LLaMA) that serves as the “brain” of each agent, providing generative capabilities, reasoning, and tool-calling support.
Shared Memory (Persistent Memory)	A centralized or distributed store (often a vector database) where agents write and retrieve contextual information, enabling long-term planning and consistency across iterations.
ModelOps	The practice of managing AI models (and agent prompts) throughout their lifecycle—development, deployment, monitoring, and retirement—with version control, CI/CD testing, and drift detection.
Application Security	Safeguards and best practices (e.g., prompt sanitization, authentication, sandboxing) designed to protect agentic systems from prompt injection, identity spoofing, and lateral exploits.
Model Privacy	Techniques (e.g., differential privacy, homomorphic encryption, secure enclaves) that ensure sensitive data—either during training or inter-agent communication—remains protected in multi-agent workflows.
Prompt Injection	A security exploit in which an attacker crafts input containing hidden instructions that corrupt an agent’s reasoning or propagate malicious commands through agent interactions (“prompt infection”).
Retrieval-Augmented Generation (RAG)	A framework where agents query an external knowledge store (e.g., vector database) to fetch relevant documents or facts, then condition their LLM responses on that retrieved context.
Role-Specialized Coordination	An architectural pattern in which each agent is assigned a specific function (e.g., planner, verifier, coder) and collaborates via structured communication protocols to achieve complex tasks.
Decision Provenance Graph	A graph-based representation that traces data flows and decision steps across multiple agents, enabling post-hoc auditing and system-level interpretability.
Tool-Use Interface	The mechanism by which an agent issues structured commands (e.g., API calls, code execution) to external services or environments and incorporates the results back into its reasoning.
Trust Score	A composite metric that quantifies an agent’s reliability, alignment with user goals, and consistency over time, often combining accuracy, safety-violation rates, and calibration of confidence.
Composite Metric	An aggregate evaluation score (e.g., a weighted sum of trustworthiness, explainability, user-centered performance, and coordination metrics) used to benchmark different agentic systems.

## Appendix

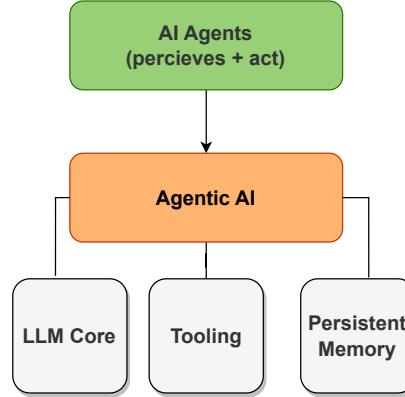


Figure A.1: Traditional AI Agent vs. Agentic AI.