

Trust by Design: A Unified Framework for Explainable, Secure, and Reliable Agentic Artificial Intelligence

Naseef Talukder

Faculty of ITHM Department, Primeasia University, Dhaka

Zulfiker Chowdhury

BBA, North South University, Dhaka

Abstract: Agentic artificial intelligence systems combine autonomous planning, memory, tool use, and multi-agent coordination, creating risks that extend beyond the behavior of an individual model to include interactions with external environments and operational workflows. Trustworthy deployment therefore requires assurance that integrates explainability, security, reliability, and governance across system layers and throughout the lifecycle. This paper proposes a unified framework that connects decision- and action-level explanations with threat modeling, bounded permissions, runtime verification, traceability, and reliability monitoring. It treats explanations as one component of assurance rather than a sufficient basis for trust, and situates them alongside calibrated human oversight and enforceable operational controls. The framework provides a structured basis for designing and evaluating agentic systems whose actions are interpretable, constrained, and auditable.

Keywords: *Agentic AI; Explainable AI; Trustworthy AI; Runtime Assurance; AI Security; Reliability Monitoring.*

I. INTRODUCTION

Agentic artificial intelligence (AI) systems extend large language models with capabilities for autonomous reasoning, planning, memory, and tool use, enabling them to undertake multi-step tasks and coordinate across agents. These capabilities also complicate assurance: interaction with external tools and environments creates security risks, while autonomous and multi-agent operation raises challenges for safety, verification, and oversight.

Existing approaches address important aspects of trustworthy AI, but explaining individual predictions does not necessarily account for the sequences of actions taken by an agent. Nor can explanation alone be assumed to produce appropriate trust, as the relationship between explainability and user trust is not automatic. Trust by Design develops a unified framework for explainable, secure, and reliable agentic AI, bringing these concerns together across agent behavior, tool interaction, and assurance.

From Predictive Models to Autonomous Perception–Reasoning–Action Loops

Agentic AI Differs From a Predictive Model Because It Converts Observations into Plans, Tool Calls, Environmental Updates, And Subsequent Observations. Contemporary Descriptions Identify Tool Use, Reflection, React-Style Interaction, Planning, And Multi-Agent Collaboration as Recurring Design Patterns [1]. The System Therefore Carries State Across Time, Operates Under Incomplete Information, and Can Create External Effects Before a Human Reviews Its Final Response. Trust Must Consequently Attach to the Execution Process, Not Only to The Accuracy of an Isolated Prediction.

A useful abstraction is a closed loop: perception supplies evidence, reasoning constructs a plan, orchestration assigns actions, tools alter an environment, and feedback revises the next state. Memory and inter-agent messages extend this loop beyond a single context window. Reviews of agentic architectures distinguish these systems from conventional agents through persistent execution, adaptive planning, memory augmentation, and coordination among specialized agents [2], [3]. A trustworthy system must therefore make each consequential transition admissible, attributable, reversible where feasible, and intelligible to authorized overseers.

Corpus Map: Reviews, Meta-Analyses, Human Studies, Benchmarks, Formal Models, and Governance Frameworks

The literature combines broad reviews, human-subject studies, security benchmarks, formal assurance proposals, and governance frameworks. Reviews organize the problem around explainability, robustness, privacy, accountability, lifecycle governance, and human oversight [4], [5]. Human evidence includes a meta-analysis of 65 AI-trust studies [6], a review of empirical trust research [7], and experiments on explanations and confidence [8], [9]. Security evidence includes end-to-end prompt-injection testing,

RAG attack benchmarks, adaptive bypass studies, and MCP threat analysis [10], [11], [12], [13]. Formal contributions specify temporal properties, runtime verification, trace contracts, or admissibility gates [14], [15], [16], [17].

Conceptualizing Trust, Risk, Security, Explainability, and Reliability Across the AI Lifecycle

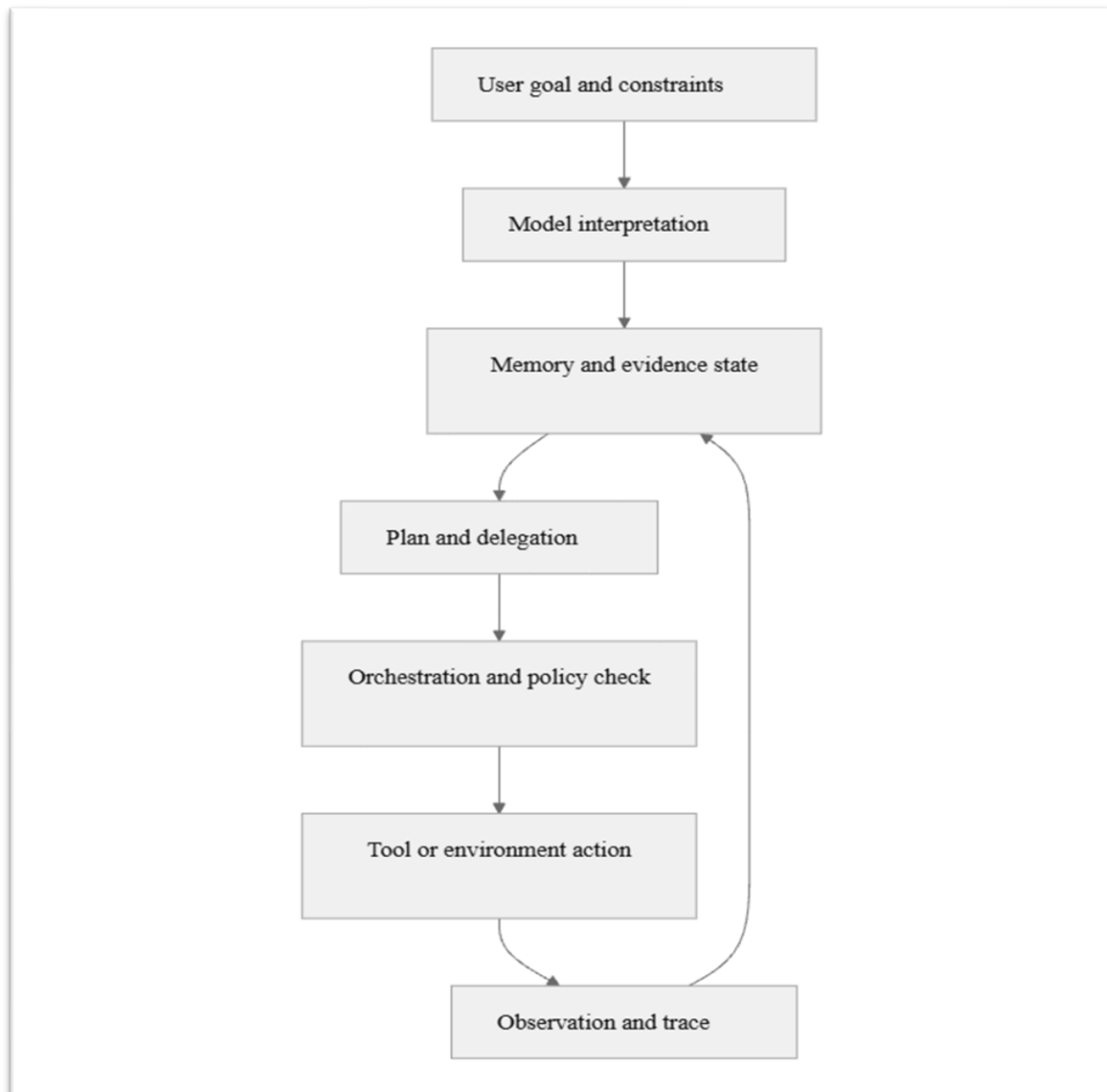
Here, **trust** means a justified willingness to depend on an agent under uncertainty; **trustworthiness** concerns the system properties that warrant that reliance. **Risk** concerns the possibility and consequence of harmful deviation, whereas **security** concerns resistance to unauthorized influence, disclosure, and action. **Explainability** concerns usable accounts of evidence, uncertainty, decisions, and execution traces. **Reliability** concerns consistent achievement of authorized goals without violating safety or operational constraints. These concepts intersect but cannot substitute for one another: an intelligible action may remain unsafe, a secure system may still reason poorly, and high task accuracy may coexist with unacceptable state transitions. This distinction motivates a cross-layer treatment of trust as an executable system property.

Agentic AI as a Cross-Layer Trust, Risk, and Security Management Problem Architectural Layers of Agentic Systems: Models, Memory, Planning, Orchestration, Tools, and Environments

A deployed agent is best represented as a stack rather than as a model surrounded by optional services. The model interprets context and proposes reasoning; memory stores prior observations or commitments; planning decomposes goals; orchestration selects agents and tools; tools expose capabilities; and the environment returns observations or absorbs effects. Multi-agent systems add delegation, message routing, role boundaries, and shared resources. The architecture review literature identifies coordination failures, prompt manipulation, memory risks, tool misuse, and governance gaps as distinct but interacting sources of exposure [3], [18].

Trust controls must follow information and authority through this stack. Provenance should distinguish user instructions, retrieved data, tool metadata, model-generated text, and agent messages. Memory writes require validation because a false or adversarial observation can influence later plans after the original context has disappeared. Planning requires competence boundaries, resource budgets, and termination conditions. Orchestration requires explicit role and capability constraints, while tools require authentication, parameter validation, isolation, and post-action confirmation. Environment-facing actions require transaction records, rollback or compensation procedures, and a policy decision at the point of effect.[19]

The security literature frames this as a shift from input perturbation toward system-level vulnerability across perceptual, cognitive, and executive layers [20]. The reliability literature reaches a parallel conclusion: agent failure can originate in long-horizon interaction, stochastic decisions, external side effects, role drift, unsupported claims, or non-termination [16]. Trust management therefore cannot be assigned solely to model training or prompt design.



Trust Antecedents in Human–Autonomous Teams and Multi-Agent Collaboration

Human trust depends on more than observed task performance. The meta-analysis of 65 studies found that human characteristics, AI attributes, and interaction context all predict trust, with reliability and anthropomorphism among the reported antecedents [6]. A broader empirical review links cognitive trust to transparency, reliability, tangibility, and immediacy, while associating emotional trust particularly with anthropomorphic representation [7]. These findings caution against treating a trust score as a direct readout of system competence. Users may trust a fluent, familiar, or socially presented agent despite weak evidence about its reliability.[21]

Human-autonomous teaming adds role allocation and mutual dependence. A scoping review reports that existing studies do not systematically connect agent architecture to the trust antecedents they measure [22]. This gap matters because a passive recommender, a delegated software agent, and a network of coordinating agents impose different demands on transparency, predictability, intervention, and accountability. A system that delegates subtasks also requires users to judge whether the agent selected an appropriate collaborator, preserved the task objective, and respected authority boundaries. Coordination quality thus becomes part of perceived trustworthiness, even when every individual model appears competent.[23]

Mutual trust extends the analysis beyond whether humans trust agents. In two studies involving 1,373 participants, perceived that an AI trusted the human was measured with a validated scale and was associated with trust in the agent and behavioral intentions [24]. This construct should not be confused with technical authorization. It describes a user's perception that the system recognizes human competence and grants meaningful responsibility. Interfaces that communicate confidence in human review,

preserve override authority, and explain escalation decisions may therefore affect cooperation independently of model accuracy.[25]

A Comparative Architecture Table for Mapping Trust, Risk, and Control Points

The following mapping joins architectural location to the principal trust question and a control that can be evaluated during operation.

Layer	Trust question	Typical failure	Control point
Model	Is the interpretation supported and appropriately uncertain?	Hallucination, overconfident classification, instruction confusion	Evidence checks, calibrated confidence, refusal conditions
Memory	Is retained state accurate, authorized, and traceable?	Memory poisoning, stale commitments, privacy leakage	Provenance, write validation, quarantine, expiry
Planning	Does the plan preserve the goal and constraints?	Unsafe decomposition, scope drift, non-termination	Plan contracts, resource budgets, competence limits
Orchestration	Are roles, messages, and delegations authorized?	Cross-agent poisoning, role drift, shadow coordination	Identity, capability scopes, message validation
Tools	Does the proposed call match authorized intent?	Parameter abuse, tool poisoning, excessive privilege	Schema validation, sandboxing, action mediation
Environment	Is the state transition safe and recoverable?	Unauthorized write, irreversible side effect	Human approval, transaction control, rollback
Oversight	Can an accountable person reconstruct and intervene?	Untraceable failure, automation bias	Receipts, alerts, replay, escalation

This cross-layer view also clarifies why conventional AI TRiSM categories require adaptation. Explainability, ModelOps, security, privacy, and lifecycle governance remain relevant, but agentic systems add coordination, persistent memory, tool efficacy, and trajectory-level monitoring [3]. The proposed Component Synergy Score and Tool Utilization Efficacy illustrate attempts to measure inter-agent enablement and whether tools are invoked correctly and efficiently [3]. Such measures should complement, rather than replace, security and assurance properties.

Explainability, Uncertainty, and the Calibration of Human Reliance from Feature-Level Explanations to Action-, Plan-, and Trace-Level Explanations

Feature-level explanation identifies input attributes associated with a prediction under a relatively fixed decision structure. This representation is insufficient when an agent selects tools, revises goals, delegates subtasks, and changes the environment. The relevant question becomes why the agent adopted a plan, why it selected a particular tool, what evidence it treated as authoritative, and which intermediate state caused a later action. Recent comparative work reports stable feature rankings in static settings, with Spearman's rho of 0.86, while finding that attribution methods do not reliably diagnose execution-level failures in agentic trajectories [26].

Trace-level explanation should therefore expose the sequence from observation to state update, plan revision, tool invocation, result interpretation, and final action. A useful trace records the relevant evidence, confidence or uncertainty, policy checks, rejected alternatives, delegation route, and environmental effect. It should distinguish a model belief from an externally verified fact and an intended action from an executed action. This distinction supports both user understanding and engineering diagnosis.[27]

The same study found that state-tracking inconsistency was 2.7 times more prevalent in failed runs and reduced success probability by 49% [26]. These findings support explanations that identify state divergence rather than merely listing influential tokens. Trace explanations can also expose whether failure arose from retrieval contamination, memory corruption, a planning error, a tool response, or an unauthorized transition. They consequently connect explainability with reliability and security rather than treating it as a presentation layer.

Experimental and Meta-Analytic Evidence on Explanation, Confidence, and Trust Calibration Trust calibration

concerns the alignment between reliance and actual system competence. It differs from high trust: a calibrated user accepts correct recommendations, questions weak ones, and intervenes when uncertainty or task conditions exceed the agent's competence. In two human experiments, confidence information helped calibrate trust in an AI model, whereas a local explanation raised problems for decision support; calibration alone did not improve joint performance when humans lacked knowledge that complemented the model's errors [8]. This result separates an attitudinal endpoint, reported trust, from a functional endpoint, team decision quality.

Automatically generated explanations produced a more favorable result in a simulated human-robot team. Added explanations improved transparency, trust, and team performance, with explanation content affecting the observed calibration outcome [9]. The contrast with local explanations may reflect differences in task structure, explanation design, and whether users could use the information to identify system limitations. It does not establish that explanations are uniformly beneficial.

A meta-analysis of 90 studies found a statistically significant but moderate positive correlation between explainability and trust [28]. The magnitude indicates that explanation is one antecedent among several, consistent with the earlier meta-analysis linking trust to human, system, and contextual factors [6]. Uncertainty-aware interpretability has therefore been proposed as a means of helping decision makers identify likely failures under limited or poor-quality data, especially in high-stakes coalition operations [29]. The evidence supports calibrated reliance as the target, not maximum reported trust.[30]

Evidence source	Design or scope	Reported endpoint	Main implication
Trust meta-analysis	65 articles	Antecedents of AI trust	Reliability, human factors, AI attributes, and context all matter [6]
Human-AI experiments	Two experiments	Confidence, explanation, trust calibration, joint performance	Confidence can aid calibration; calibration may not improve performance without complementary expertise [8]
Human-robot testbed	Survey and behavioral data	Transparency, trust, team performance	Automatically generated explanations improved reported and behavioral outcomes [9]
Explainability meta-analysis	90 studies	Explainability-trust association	Positive association was moderate, not determinative [28]
Agentic explanation comparison	Static attribution versus traces	Failure localization and success	Trace-grounded evaluation better addresses execution breakdowns [26]

The available numerical evidence is not directly combinable into a common effect-size graph because the studies use different populations, tasks, explanation types, and endpoints. A table is therefore more defensible than a synthetic chart. It also makes visible the distinction between trust, calibration, team performance, and failure diagnosis.

The Explainability–Trust Debate: Why More Explanation Does Not Necessarily Produce Better Reliance

Explanations can increase trust without increasing justified reliance. A persuasive explanation may be faithful to the model's internal basis yet irrelevant to the user's decision, or it may disclose a plausible rationale without exposing uncertainty or constraint violations. Excessive detail can impose cognitive cost, obscure the decisive evidence, and encourage users to accept a coherent narrative. The moderate association in the 90-study meta-analysis is consistent with this non-equivalence [28].

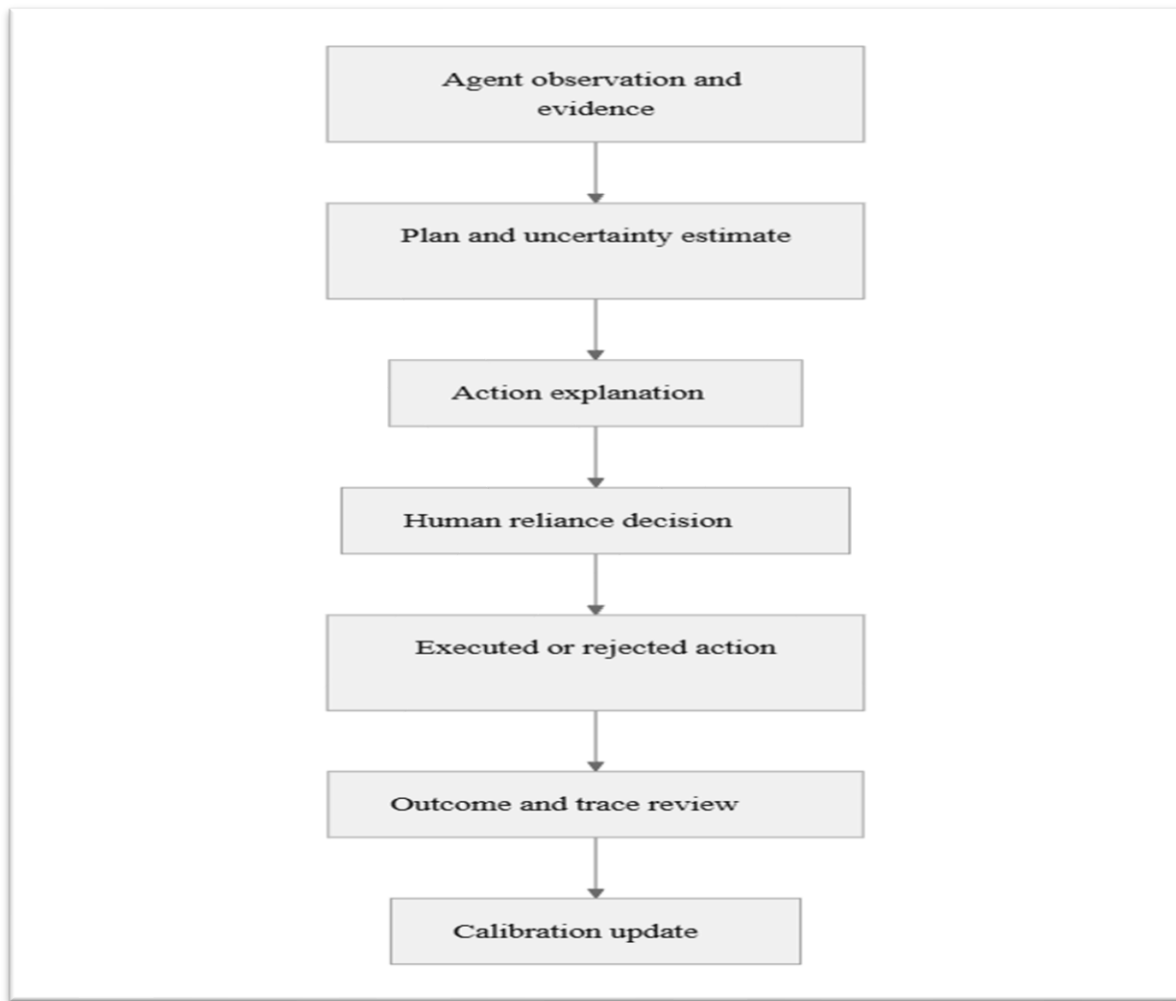
Explanation quality should consequently be assessed along several separate dimensions: fidelity to the system process, stability under input changes, complexity, usefulness for the task, uncertainty disclosure, and capacity to support error detection. In a cybersecurity assessment framework, alternative explanation methods were compared for complexity and robustness, with the stated aim of selecting an approach that performed well on both dimensions in phishing detection [31]. A compact explanation is not automatically robust, and a robust explanation is not automatically comprehensible.

For agentic systems, explanation can itself disclose sensitive context or reveal attack-relevant control logic. Security boundaries should determine which trace fields are visible to users, operators, auditors, or collaborating agents. The design objective is therefore conditional transparency: enough information to support appropriate reliance and accountability, while withholding secrets and preventing explanation channels from becoming instruction channels. This requirement links explanation design directly to access control and privacy. [32]

Comparative Tables and Graphs for Explanation Complexity, Robustness, Uncertainty, and Calibration Outcomes

A practical evaluation should report explanation complexity and robustness beside calibration outcomes. Complexity may include length, number of state transitions, or cognitive effort, but the selected measure must be defined before comparison. Robustness should test whether the explanation remains materially stable under benign perturbations, retrieval variation, or repeated stochastic

execution. Calibration should measure reliance against correctness or competence, not trust ratings alone. The framework in cybersecurity demonstrates why method selection should consider complexity and robustness jointly [31], while agentic trace research shows why execution diagnosis requires state-aware measures [26].



The diagram represents a feedback relation rather than a claim that every implementation uses these exact stages. Its central implication is that explanation should precede consequential reliance and remain connected to post-action evidence. Confidence should be presented with its reference class and uncertainty source, not as an unqualified scalar. The next problem is to protect this explanatory and decision channel from adversarial context, poisoned tools, and confused authority.

Securing Autonomous Agents Against Prompt Injection, Tool Poisoning, and Authority Confusion the Expanded Threat Surface of Retrieval, Memory, Model Context Protocols, and Inter-Agent Communication

Agent security begins with an authority distinction: text that the model can read is not thereby entitled to direct what the system does. Retrieval results, web pages, documents, memory entries, tool descriptions, tool outputs, and inter-agent messages may all enter a common generative context. When data and instructions share that channel, an untrusted string can influence a secret-bearing response or an action proposal without application-level authorization [33]. Prompt annotations may describe a boundary, but only a technical projection, capability restriction, or output validator can enforce it [33].

Persistent memory expands the time horizon of attack. A malicious observation can be stored, later retrieved as apparently trusted context, and used to justify an action after the initiating event is no longer visible. Inter-agent communication adds provenance and delegation problems: a message may be authentic but outside the sender's authority, or it may carry an instruction that the receiving agent should treat only as data. Reviews of orchestration identify cross-agent poisoning, cognitive drift, and emergent shadow behaviors as distinct threats in multi-agent systems [18].

Model Context Protocols introduce a further boundary among hosts, clients, language models, servers, data stores, and authorization services. Threat modeling across these components identifies tool poisoning through malicious instructions embedded in metadata as a prominent client-side vulnerability, with insufficient static validation and parameter visibility reported across tested clients [13]. Protocol-level security must therefore address how decisions are made, not merely who is authenticated.[34]

Prompt Injection, Indirect Instruction Attacks, Privacy Leakage, and Tool-Use Integrity Failures

Direct prompt injection attempts to override trusted instructions in the user-facing context. Indirect prompt injection places the attack in retrieved or environmental content so that the agent encounters it while performing an apparently benign task. In an investigation of commercial LLM-integrated applications, 31 of 36 examined applications were reported susceptible to the described black-box attack technique [35]. The endpoint was application susceptibility, not universal exploitability across all deployments, and should be interpreted accordingly.

RAG-enabled agents face several attack categories, including direct injection, context manipulation, instruction override, data exfiltration, and cross-context contamination. A benchmark contained 847 adversarial cases across these categories and evaluated seven language models [11]. Web-agent evaluation adds realism by testing end-to-end tasks: attacks partially succeeded in up to 86% of cases, although agents often failed to complete the attacker's full objective [10]. Partial attack success remains a security failure when it triggers disclosure, an unauthorized tool call, or an irreversible intermediate effect.[36]

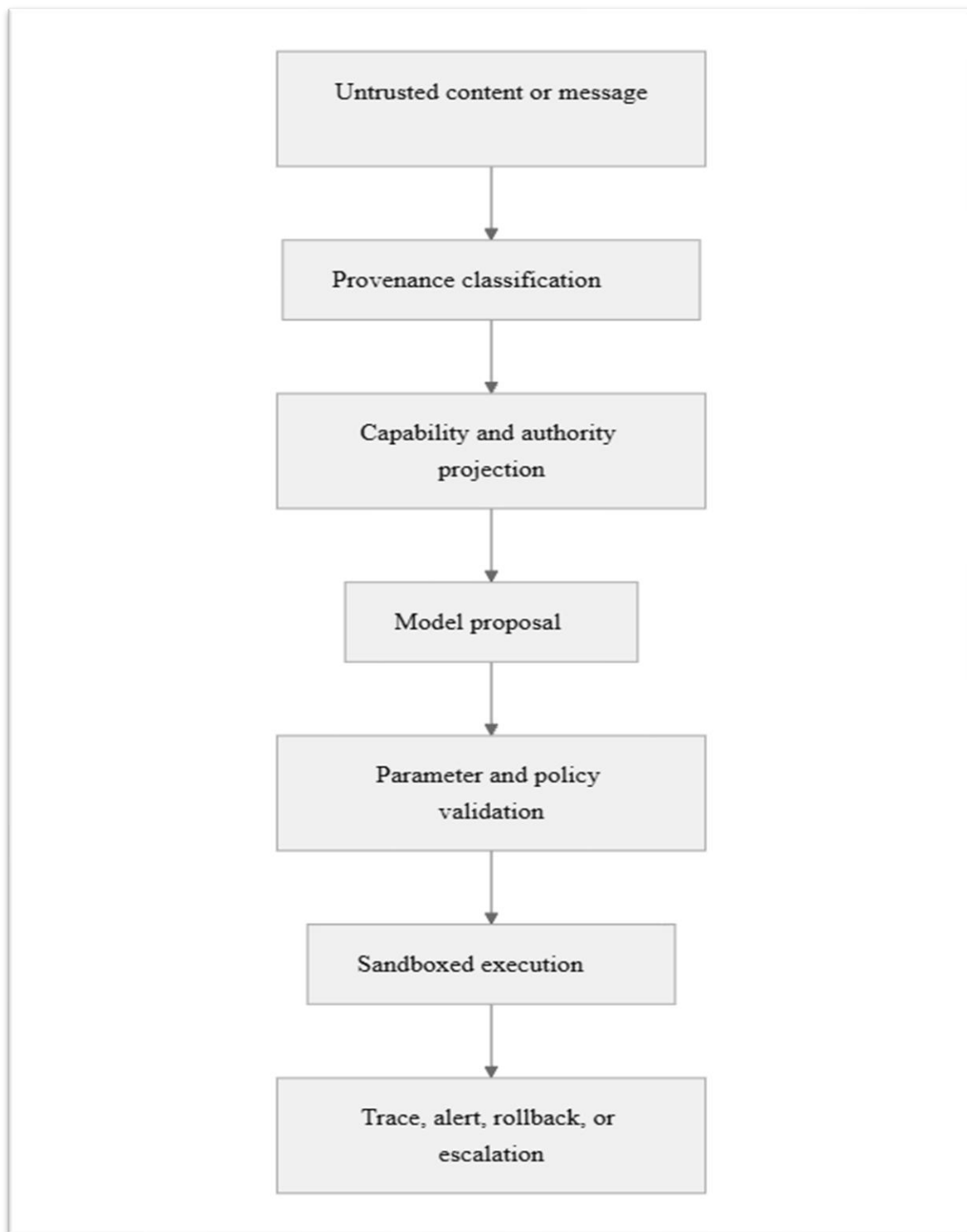
Adaptive attacks expose the limits of static defenses. An evaluation of eight defenses bypassed each with adaptive attacks and reported attack success above 50% across the tested defenses [12]. This result does not show that every defense fails in every environment, but it establishes that non-adaptive testing can overstate robustness. Privacy leakage and tool-use integrity must therefore be evaluated with benign controls, forbidden-action distinguishers, and explicit measurement of whether the adversarial channel was closed before generation [33].

Defense Architectures: Authentication, Security Boundaries, Sandboxing, Guardrails, and Runtime Policy Enforcement

A defensible architecture uses layered controls with different failure assumptions. Authentication establishes identity, but it does not establish that an authenticated agent's proposed action is appropriate. Authorization limits capabilities, provenance separates trusted from untrusted observations, and sandboxing limits damage when reasoning fails. Input filtering can remove known attack patterns, while hierarchical guardrails preserve instruction priority. Output validation checks whether proposed content or parameters violate policy. Runtime monitoring observes the trajectory and can block, rewrite, quarantine, or escalate actions.

The benchmark with 847 adversarial cases reported that a combined framework using embedding-based anomaly detection, hierarchical system-prompt guardrails, and multi-stage response verification reduced successful attacks from 73.2% to 8.7% while retaining 94.3% of baseline task performance [11]. These results support defense in depth, but they should not be generalized beyond the tested models, attacks, and task setting. A separate hybrid approach combined dynamic prompt assembly with runtime security monitoring and reported the lowest attack success among its compared configurations, with increased latency, memory, and CPU use described as manageable [37].

Runtime policy enforcement should sit at the language-to-action boundary. The proposed constitutional governance architecture AEGIS places deterministic policy after reasoning and before execution, and frames the boundary through reference-monitor properties [38]. Controlled Agentic AI Systems similarly model governance as an operator that transforms proposed actions into admissible executed actions, with audit traces and replayability conditions [39]. These approaches address authority confusion more directly than prompt wording because the final action is mediated independently of the model's narrative.[40]

**Benchmark-Based Comparison of Attack Success, Defense Efficacy, Computational Cost, and Residual Risk**

The benchmark evidence is heterogeneous, but it exposes a consistent pattern: attack success depends on the endpoint and the point at which defenses intervene. Application susceptibility, partial web-task success, forbidden disclosure, and full attacker-goal completion are distinct outcomes. A security report should identify which of these was measured rather than reporting “robustness” as a single value.

Study or framework	Threat setting	Quantitative finding	Cost or residual-risk implication
Commercial application study	Black-box prompt injection	31 of 36 applications susceptible to the tested technique [35]	Application-level exposure remained despite varied deployments
RAG benchmark	Five injection categories, seven models	Success reduced from 73.2% to 8.7%; 94.3% baseline performance retained [11]	Combined defenses improved attack rate, but scope was benchmark-bound
Adaptive-defense study	Indirect prompt injection	Eight defenses bypassed; attack success consistently above 50% [12]	Adaptive evaluation leaves substantial residual risk
Web-agent benchmark	Realistic end-to-end tasks	Partial attack success reached 86% in some cases [10]	Failure to complete the full goal can mask harmful partial effects
AgentSecBench	Integrity, confidentiality, capability games	Evaluated six defense classes on two Qwen model sizes [33]	Channel closure, not prompt annotation, is the relevant security test
Hybrid defense study	Direct, indirect, hidden injections	Combined approach had lowest attack success among compared methods [37]	Security improvement incurred latency, memory, and CPU overhead

The table shows why residual risk must be reported alongside efficacy and cost. A defense that blocks a disclosure marker may leave a semantic or protocol-level action channel open. Conversely, a strict sandbox may reduce harm while leaving the model vulnerable in an isolated test. Protocol testing also warns that gateway or identity controls can create false confidence when they do not evaluate adversarial decision behavior [41]. Security assurance therefore requires attack adaptation, tool and memory fault injection, authorization tests, and post-action effect analysis.

Reliability, Verification, and Assurance Across Multi-Step Agentic Execution Why Final-Output Accuracy Is Insufficient for Persistent, Tool-Using Agents

A final answer can appear correct after an unsafe route, unauthorized data access, excessive tool calls, or an unrecorded state mutation. Persistent agents may also leave behind memory entries, files, messages, or transactions that affect later tasks. The reliability problem consequently concerns the trajectory and its effects, not only the terminal text. Agent evaluation surveys identify long-horizon coherence, uncertainty handling, tool effectiveness, autonomy, collaboration, memory consistency, and safety as dimensions inadequately represented by conventional short-horizon metrics [42].

Trace-based assurance makes this distinction explicit. Message-Action Traces can record each step, provide machine-checkable contracts, localize the first violation, support deterministic replay, and enable fault injection at service, retrieval, and memory boundaries [16]. A run that reaches the correct output after violating a no-disclosure rule should fail the relevant contract. A run that produces an imperfect answer while preserving authorization, termination, and recovery constraints may require a different diagnosis.[43]

Operational monitoring must also distinguish novelty from failure. A proposed reliability framework combines out-of-distribution detection with a transparency layer so that operators can judge whether an output is potentially unreliable and intervene [44]. This approach reflects the practical limit of static accuracy estimates under changing environments. Reliability is thus a property of the agent-environment relation over time.

Runtime Verification, Trace-Based Contracts, Temporal Logic, and Probabilistic Safety Guarantees Runtime verification evaluates observed execution against formal or machine-checkable properties while the system operates. Trace contracts can specify permitted message order, role boundaries, tool parameters, termination, factual support, and governance outcomes [16]. Temporal logic extends this approach from individual events to sequences, such as requiring that every high-impact action be preceded by authorization or that every created task eventually terminate or escalate. A formal host-agent and task-lifecycle model defines 16 host properties and 14 task properties across liveness, safety, completeness, and fairness [14].

Because learned agents are stochastic and adaptive, deterministic proof of every future behavior may be unavailable. AgentGuard therefore abstracts raw input and output into formal events, builds an online Markov Decision Process, and applies probabilistic model checking to quantitative properties [15]. Such assurance should be interpreted as conditional on the abstraction, learned transition model, monitored state, and specified property. It does not prove that unobserved semantic failures cannot occur.

Formal assurance proposals also differ in their scope. Assured Intelligence Systems separates representation, memory, planning, action, governance, verification, release, and self-edit into typed layers and uses a conjunctive admissibility relation over support, policy, verification, and recovery [17]. AEGIS Algebra similarly treats assurance as a non-compensatory state combining structural feasibility, constraints, partial-evidence logic, and temporal validity [45]. These frameworks provide design formalisms, not empirical evidence that current agents satisfy the proposed conditions.

Bounded Autonomy and the Control of Consequential State Transitions

Bounded autonomy preserves initiative within an explicit behavioral envelope. The envelope should constrain authority, resources, time, data access, delegation depth, environmental effects, and recovery obligations. High-consequence actions require stronger evidence, narrower capability scopes, and more direct human or procedural authorization than low-consequence informational actions. A system should also degrade gracefully when evidence is incomplete, tools conflict, or competence boundaries are reached.

The key control point is the consequential state transition. An agent may generate any number of candidate plans internally, but execution should occur only after policy, support, verification, and recovery conditions are assessed. Write-capable tools require transaction semantics, terminal-state resolution, and rollback or quarantine behavior; these properties are specified in the Assured Intelligence Systems architecture [17]. Governance as a projection can repair or block proposed actions, but the projection must preserve auditability and expose when the executed action differs from the plan [39].

Human approval should be selective rather than indiscriminate. Constant approval can overload operators and produce routine rubber-stamping, while absent approval leaves high-impact transitions unreviewed. The appropriate threshold depends on consequence, reversibility, uncertainty, competence, and available recovery. A behavioral specification approach frames this as preserving beneficial initiative while enforcing an auditable and secure envelope [46].

Reliability Metrics, Evaluation Benchmarks, and a Proposed Dashboard of Safety, Security, and Functional Properties

A useful dashboard should report separate dimensions rather than collapse performance into one score. Functional measures include task success, factual support, plan completion, termination reliability, tool utilization, and memory consistency. Safety measures include constraint compliance, unsafe-action rate, escalation adequacy, recovery success, and severity-weighted side effects. Security measures include injection success, unauthorized disclosure, capability-integrity failure, provenance violations, and containment. Human measures include calibrated reliance, intervention quality, workload, and override latency.[47]

The eleven-dimensional evaluation taxonomy includes task success, output quality, planning and reasoning, uncertainty decisions, tool use, autonomy and intervention, adaptation, safety and alignment, efficiency, multi-agent collaboration, and long-term memory consistency [42]. Trace-based work adds contract compliance, containment rate, factuality indicators, and governance-outcome distributions [16]. These measures should be stratified by scenario, threat type, consequence level, model version, random seed, and orchestration topology. Reproducibility requires retaining the relevant trace, tool versions, policy version, and environmental state.

A dashboard should also display confidence intervals or repeated-run distributions when stochastic behavior is material. It should record the first violated invariant, not only the final failure category, and distinguish blocked actions from actions that reached the environment. This structure makes reliability evidence actionable for deployment decisions and links directly to governance, where admissibility and accountability are determined.

Governance, Human Oversight, and the Institutional Conditions of Trustworthy Agency Constitutional Governance, Reference-Monitor Properties, and Policy Enforcement at the Action Boundary Constitutional governance expresses system-level rules that constrain what an agent may propose or execute. Its effectiveness depends on enforcement outside the model's discretionary text generation. The AEGIS architecture places deterministic policy at the post-reasoning, pre-execution boundary and aligns that boundary with reference-monitor properties [38]. This arrangement separates reasoning from authorization: the model can recommend an action, but a policy mechanism decides whether the action is admissible.

Governance should cover design, runtime, and audit stages. AGENTS SAFE maps risks to scenario banks, dynamic authorization, anomaly detection, interruptibility, cryptographic tracing, and organizational controls [48]. Broader trustworthy-AI frameworks likewise treat human oversight, robustness, privacy, transparency, fairness, wellbeing, and accountability as lifecycle requirements rather than isolated interface features [49]. Institutional governance must therefore define who owns policies, who can change them, how exceptions are recorded, and how affected parties can contest outcomes.[50]

Human-in-the-Loop, Human-on-the-Loop, and Epistemic Accountability Mechanisms

Human-in-the-loop approval places a person within the action pathway; **human-on-the-loop** oversight permits autonomous execution while retaining monitoring and intervention authority. Neither arrangement is sufficient by label alone. Effective oversight requires accessible evidence, meaningful time to act, authority to stop or reverse the system, and a clear allocation of responsibility. An approval button that presents an opaque or time-pressured recommendation may preserve formal involvement without preserving epistemic control.

A review of human-in-the-loop frameworks reports recurring tensions between explainability and performance, autonomy and accountability, over-trust and under-trust, and participation and effectiveness [51]. The same review describes full delegation as feasible for only 0–20% of tasks despite reports that developers use AI assistance in approximately 60% of their work [51]. These figures should be treated as the review's synthesis rather than a universal boundary, but they illustrate why autonomy levels require task-specific evidence.

Epistemic accountability requires a record of what the agent knew, what it inferred, what it did not know, which policy applied, and why an action crossed the release boundary. Runtime accountability frameworks propose structured evaluation of claims, decisions, and actions under uncertainty [52]. Trace contracts, provenance, replay, and escalation records make that account inspectable by operators, auditors, and affected institutions.[53]

Mutual Trust, Perceived Trustworthiness, and Oversight in Human-Agent Teams

Oversight succeeds when users can form an accurate model of agent competence and when the institution supports intervention. Trust research shows that reliability, transparency, system representation, user characteristics, and task context shape reliance [6], [7]. The design target is therefore neither maximal trust nor maximal skepticism. It is calibrated cooperation in which humans understand the agent's authority, limitations, uncertainty, and recovery behavior.

Mutual trust adds a relational condition: people may cooperate more readily when they perceive that the agent recognizes their competence and preserves their role [24]. Interfaces should communicate when the agent is requesting expertise, escalating because evidence is weak, or declining an action because authority is absent. Such signals can support human agency without anthropomorphizing the system or implying moral reciprocity.

Institutional trust also depends on auditability beyond the immediate interaction. Governance frameworks connect lifecycle controls with legal, ethical, technical, and social requirements, emphasizing that responsibility includes the actors and processes surrounding an AI system [49]. A trustworthy agent is therefore a governed arrangement of model, memory, tools, policies, operators, records, and review procedures. The remaining design challenge is to make those arrangements executable without reducing trust to a score.

Trust by Design: Integrating Explainability, Security, Reliability, and Governance

Design Principles for a Unified Cross-Layer Trust Architecture

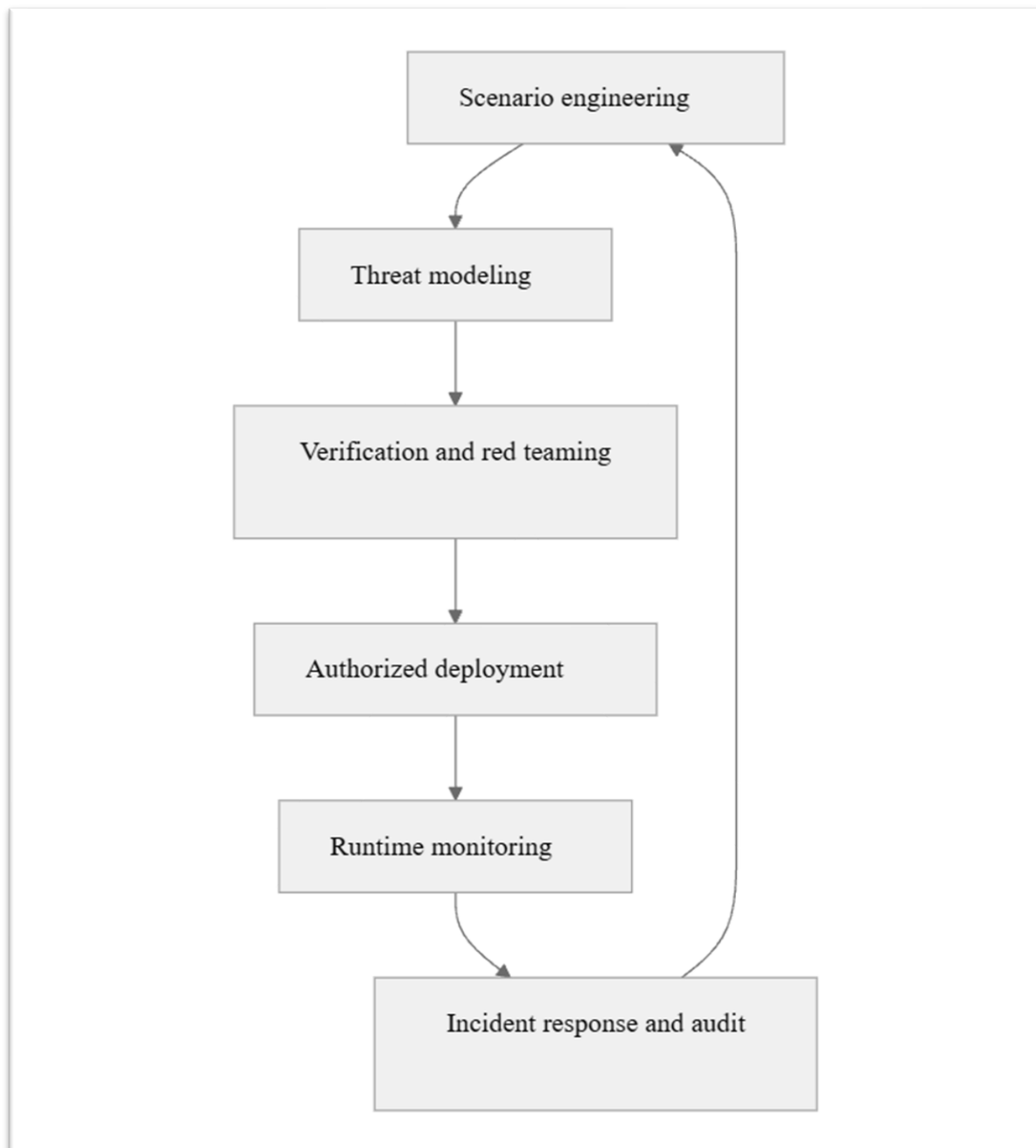
Trust by design treats an agent as a governed socio-technical system rather than as a language model with added tools. Four principles follow. First, explanations must describe evidence, uncertainty, planned actions, and execution traces, because feature-level attributions cannot reliably diagnose failures distributed across agent trajectories [26]. Second, security controls must separate data from authority: retrieved text, memory entries, tool metadata, and inter-agent messages may inform reasoning but must not acquire permissions merely by entering context [33]. Third, reliability must concern state transitions, termination, recovery, and side effects, not final-answer accuracy alone [16]. Fourth, governance must operate at the action boundary, where proposed effects can be authorized, rewritten, blocked, or escalated before execution [38], [39].

These principles are **non-compensatory**. High task performance cannot offset an unverified transfer, an unauthenticated tool, or an absent recovery path. Formal assurance proposals express this through conjunctive admissibility over support, policy, verification, and recovery [17]. The practical implication is consequence-scaled autonomy: advisory actions may use lighter controls, whereas irreversible or safety-sensitive actions require stronger evidence, explicit authorization, and replayable records. Such architecture links technical controls with institutional accountability across the system lifecycle [49].

A Lifecycle Model Linking Scenario Engineering, Threat Modeling, Verification, Monitoring, and Audit

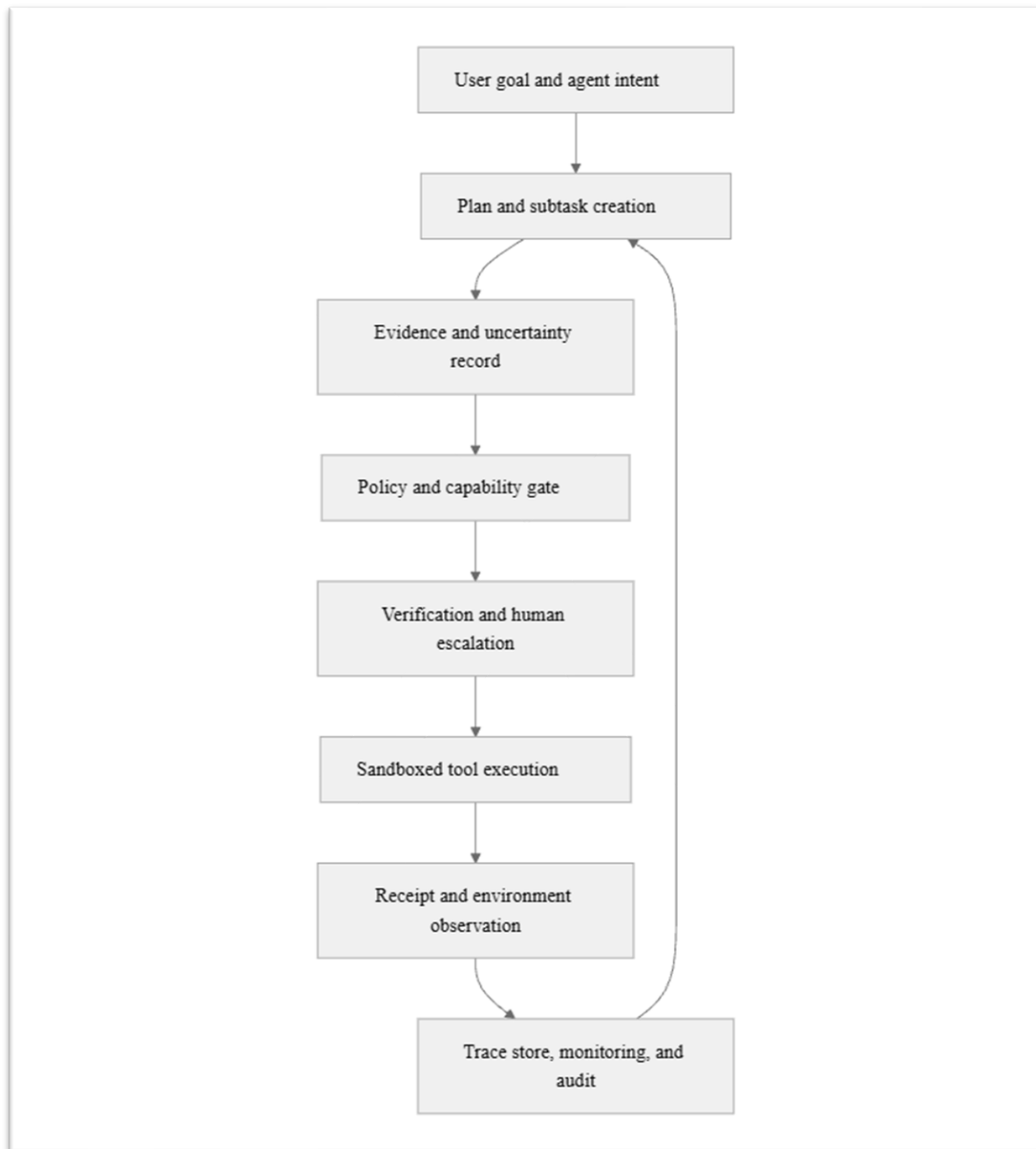
The lifecycle begins with **scenario engineering**, which specifies operational contexts, competence boundaries, failure consequences, and validation criteria before model selection [54]. Threat modeling then maps assets, trust boundaries, authorities, attack paths, and failure propagation across models, memory, retrieval, orchestration, tools, and environments. This cross-layer scope is necessary because adversarial influence can propagate from perception through policy to actuation [20].

Pre-deployment verification should combine scenario tests, adversarial evaluation, contract checking, and human-oversight trials. During operation, telemetry records provenance, uncertainty, policy decisions, tool calls, state changes, interventions, and recovery outcomes. @Runtime verification can abstract observed inputs and outputs into formal events and evaluate probabilistic properties as behavior evolves [15]. Audit then reconstructs the trajectory, tests whether controls operated as specified, and feeds failures into revised scenarios, policies, and training data. This creates a closed assurance loop rather than a one-time certification exercise.[55]



Integrated Architecture Diagram: From Agent Intent and Planning to Authorized, Verified Execution

The architecture separates **intent**, planning, evidence, authorization, execution, and observation. An agent may generate a plan and a proposed tool call, but a reference-monitor layer independently evaluates identity, provenance, capability scope, policy, uncertainty, and consequence. Only an admissible proposal proceeds to a transactional tool boundary. Receipts and observations return to the trace store, while failed or ambiguous transitions enter quarantine or human review. This arrangement limits authority confusion and supports deterministic reconstruction through message-action traces [16].



Evidence-to-Control Matrix for Matching Failure Modes with Explanations, Defenses, and Assurance Mechanisms

Controls should be selected against failure mechanisms rather than attached as generic “trust” features. Prompt injection requires provenance separation, capability restriction, output validation, and adversarial testing. Tool poisoning requires metadata validation and visibility into parameter effects, since MCP analyses identify malicious instructions in tool descriptions as a recurring client-side vulnerability [13]. Unsupported claims require evidence links, uncertainty displays, and epistemic trace review [52]. Non-termination, role drift, and unsafe side effects require trace contracts, budgets, rollback, and release gates [16], [17]. Human oversight requires actionable explanations and authority to intervene, rather than approval interfaces that merely document a decision after execution [51].

Research Gaps and Methodological Limits in the Current Agentic AI Trust Literature

Fragmented Definitions and the Lack of Standardized Cross-Layer Trust and Reliability Metrics

Trust remains conceptually divided among subjective reliance, perceived transparency, system reliability, security, accountability, and formal assurance. Reviews identify diverse trust measures and weak links between agent architecture and measured antecedents [7], [22]. Agentic evaluation surveys likewise report disconnected metrics for task success, tool use, autonomy, safety, adaptation, collaboration, and memory consistency [42]. CSS and TUE offer useful coordination and tool-use constructs, but they

do not by themselves establish security, recovery, or governance [3]. A common evaluation vocabulary should therefore report task, trajectory, security, calibration, intervention, recovery, and resource outcomes separately.[56]

Limited Longitudinal, Real-World, and Multi-Agent Evidence Beyond Reviews, Frameworks, and Benchmarks

The empirical base remains dominated by short experiments, reviews, simulations, and benchmark episodes. A review of AI trust research identifies overreliance on small samples, short durations, and experimental settings that may not represent high-stakes field use [7]. Agent benchmarks often test isolated tasks or bounded attack scenarios, while real deployments involve changing tools, users, policies, data, and organizational incentives. Security evidence also varies in strength: one RAG benchmark reports attack reduction from 73.2% to 8.7% under a combined defense, whereas adaptive attacks bypassed eight evaluated defenses with success above 50% [11], [12]. These findings demonstrate why static benchmark gains cannot establish longitudinal resilience.[57]

Open Problems in Adaptive Attacks, Emergent Behavior, Human Calibration, and Cost–Performance Trade-offs

Adaptive adversaries can exploit residual model-visible channels after nominal defenses are installed [12]. Multi-agent systems add coordination failures, cross-agent poisoning, and behaviors that are difficult to attribute to one component [18]. Human studies also show that confidence information can improve calibration without improving joint decisions when users lack complementary knowledge, while local explanations may mislead [8]. Future work must measure security, intervention burden, latency, compute, and task utility together, including the cost of false alarms and unnecessary escalation.

II. CONCLUSION

Synthesis of the Trust-by-Design Framework and Its Practical Deployment Implications

Trustworthy agency requires a bounded execution system in which explanations expose evidence and uncertainty, security controls constrain authority, verification checks trajectories, and governance determines whether effects may be released. The agent should therefore produce inspectable plans and receipts, while independent policy and verification layers control consequential transitions. This design supports calibrated reliance without treating user confidence as evidence of system trustworthiness.

Priorities for Empirical Validation, Standardized Evaluation, and Governed Autonomy

Research should establish shared metrics for trajectory reliability, attack resistance, explanation fidelity, calibration, intervention cost, recovery, and audit completeness. Evaluation must combine adaptive attacks, fault injection, multi-agent scenarios, field deployments, and longitudinal observation. Governed autonomy should then be expanded only when evidence demonstrates that competence boundaries, release gates, monitoring, and human intervention remain effective under distribution shift.

REFERENCES

- [1]. U. Nisa, M. R. Shirazi, M. A. Saip, and M. S. M. Pozi, “Agentic AI: The age of reasoning—A review,” *Journal of Automation and Intelligence*, vol. 5, no. 1, pp. 69–89, Aug. 2025, doi: 10.1016/j.jai.2025.08.003.
- [2]. I. Evtimov, A. Zharmagambetov, A. Grattafiori, C. Guo, and K. Chaudhuri, “WASP: Benchmarking web agent security against prompt injection attacks,” *arXiv (Cornell University)*. Apr. 22, 2025. doi: 10.48550/arxiv.2504.18575.
- [3]. B. Ramakrishnan and A. Balaji, “Securing AI agents against prompt injection attacks,” *arXiv (Cornell University)*. Nov. 19, 2025. doi: 10.48550/arxiv.2511.15759.
- [4]. Q. Zhan, R. Fang, H. S. Panchal, and D. Kang, “Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents,” in *Findings of the Association for Computational Linguistics: NAACL 2025*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2025, pp. 7101–7117. doi: 10.18653/v1/2025.findings-naacl.395.
- [5]. [C. Huang, X. Huang, N. P. Tran, and A. M. Fard, “Model context protocol threat modeling and analysis of vulnerabilities to prompt injection with tool poisoning,” *Journal of Cybersecurity and Privacy*, vol. 6, no. 3, p. 84, May 2026, doi: 10.3390/jcp6030084.
- [6]. A. Edoardo, S. Ananth, and C. Berkay Z., “Formalizing the Safety, Security, and Functional Properties of Agentic AI Systems,” *ArXiv.org*, 2025, [Online]. Available: <https://arxiv.org/pdf/2510.14133>
- [7]. R. Koohestani, “AgentGuard: Runtime verification of AI agents,” *arXiv (Cornell University)*. Sep. 28, 2025. doi: 10.48550/arxiv.2509.23864.
- [8]. C. Păduraru, P.-L. Bouruc, and A. Ștefănescu, “A trace-based assurance framework for agentic AI orchestration: Contracts, testing, and governance,” *arXiv (Cornell University)*. Mar. 18, 2026. doi: 10.48550/arxiv.2603.18096.
- [9]. J. C. J. K. Cliff, “Assured intelligence systems: A governed architecture for reliable, auditable, and controllable agentic AI,” *Zenodo (CERN European Organization for Nuclear Research)*. 2026. doi: 10.5281/zenodo.20122776.

- [10]. I. A. Rafe, M. R. Dewan, and M. K. Islam, "Orchestration and verification of agentic AI systems: A survey of multi-agent collaboration and safety," *European Journal of Applied Science Engineering and Technology*, vol. 4, no. 2, pp. 238–256, Apr. 2026, doi: 10.59324/ejaset.2026.4(2).16.
- [11]. R. Haider, "The algorithmic memory of brands: Developing a longitudinal theory of how large language models learn, retain, distort and correct brand knowledge proposal field details," Jul. 2026, doi: 10.13140/rg.2.2.26017.36966.
- [12]. I. ADABARA, B. O. Sadiq, A. N. Shuaibu, Y. I. Danjuma, and M. Venkateswarlu, "Trustworthy agentic AI systems: a cross-layer review of architectures, threat models, and governance strategies for real-world deployment," *F1000Research*, vol. 14, p. 905, Sep. 2025, doi: 10.12688/f1000research.169927.1.
- [13]. P. Radanliev, O. Santos, and C. Maple, "Threats and vulnerabilities in artificial intelligence and agentic AI models," *Frontiers in Artificial Intelligence*, vol. 9, p. 1731566, Feb. 2026, doi: 10.3389/frai.2026.1731566.
- [14]. R. Haider, "From search visibility to answer share: Developing and testing a causal theory of brand discoverability and consumer conversion in generative engine optimization," Jul. 2026, doi: 10.13140/rg.2.2.24339.64807.
- [15]. A. Zamecnik, W. Febriantoro, M. Cukurova, S. Joksimović, G. Grossmann, and G. Siemens, "Linking AI agent architectures to trust antecedents in human-autonomous teams: a scoping review," *Artificial Intelligence Review*, Apr. 2026, doi: 10.1007/s10462-026-11572-z.
- [16]. R. Haider, "Attention sovereignty in digital markets: Designing and testing consumer-owned preference agents as an alternative to surveillance-based advertising," Jul. 2026, doi: 10.13140/rg.2.2.27695.09125.
- [17]. Y. Xie, R. Zhou, A. H. S. Chan, and D. Xiong, "Do you trust me? measuring people's perception of being trusted by AI in a human-agent team," *International Journal of Human-Computer Interaction*, vol. 41, no. 20, pp. 13041–13058, Mar. 2025, doi: 10.1080/10447318.2025.2468783.
- [18]. "European journal of prosthodontics and restorative dentistry," May 2026, doi: 10.1922/ejprd.v34i.
- [19]. S. Chaduvula *et al.*, "From features to actions: Explainability in traditional and agentic AI systems," *Open MIND*. Feb. 06, 2026. doi: 10.48550/arxiv.2602.06841.
- [20]. R. Haider *et al.*, "Forensic consumer psychology: An evidence-based framework for detecting deception and digital market manipulation," *Natural Resources for Human Health*, vol. 6, no. 16s, pp. 58–73, 2026.
- [21]. Z. Atf and P. R. Lewis, "Is trust correlated with explainability in AI? a meta-analysis," *IEEE Transactions on Technology and Society*, vol. 7, no. 1, pp. 70–77, Apr. 2025, doi: 10.1109/tts.2025.3558448.
- [22]. R. Tomsett *et al.*, "Rapid trust calibration through interpretable and uncertainty-aware AI," *Patterns*, vol. 1, no. 4, p. 100049, Jul. 2020, doi: 10.1016/j.patter.2020.100049.
- [23]. [3] S. Raza, R. Sapkota, M. Karkee, and C. Emmanouilidis, "TRiSM for agentic AI: A review of trust, risk, and security management in LLM-based agentic multi-agent systems," *AI Open*, vol. 7, pp. 71–95, 2026, doi: 10.1016/j.aiopen.2026.02.006.
- [24]. R. Haider, Md. R. Amin, Md. S. Arafat, I. Hossain, and R. Ahmad, "Smart Automation: Revolutionizing Business Operations, Advertising Costs and Customer Satisfaction Through Technology," *European Economic Letters*, vol. 16, no. 1, p. null, 2026.
- [25]. M. C. Calzarossa, P. Giudici, and R. Zieni, "An assessment framework for explainable AI with applications to cybersecurity," *Artificial Intelligence Review*, vol. 58, no. 5, Feb. 2025, doi: 10.1007/s10462-025-11141-w.
- [26]. RaiyanHaider, "Generative AI and quantum-inspired optimization: Redefining portfolio risk management and real-time capital allocation in volatile markets," *Journal of Informatics Education and Research*, Mar. 2026, doi: 10.52783/jier.v6i1.4540.
- [27]. F. Alpay and T. Alpay, "AgentSecBench: Measuring Prompt Injection, Privacy Leakage, and Tool-Use Integrity in LLM Agents," *ArXiv.org*, 2026, [Online]. Available: <https://arxiv.org/pdf/2605.26269>
- [28]. [34] R. Haider, T. Dwivedi, A. G. G. Girish, N. Verma, B. Kashyap, and V. S. Dubey, "Neuromarketing approaches to shaping healthy consumer choices: An integrative analysis of methods, efficacy, and ethical considerations," *CLOK (University of Central Lancashire)*, Mar. 2026, doi: 10.25258/ijddt.16.4s.62.
- [29]. Y. Liu *et al.*, "Prompt injection attack against LLM-integrated applications," *arXiv (Cornell University)*. Jun. 08, 2023. doi: 10.48550/arxiv.2306.05499.
- [30]. R. Haider, Md. F. I. Shaif, R. Ahmed, N. H. Nafi, M. R. Sumon, and M. Rahman, "Assessing the impact of influencer marketing on brand value and business revenue: An empirical and thematic analysis," *International Journal of Science and Research Archive*, vol. 16, no. 2, pp. 471–482, Aug. 2025, doi: 10.30574/ijrsra.2025.16.2.2355.
- [31]. D. Daniel and J. Sunny, "A Hybrid Defense Against Prompt Injection in LLM Agents : Evaluating Security Efficacy and Computational Performance Trade-offs," *Diva portal (Dalarna University Library)*, 2026, [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-59221>
- [32]. K. A. Tannenbaum, "AEGIS: A constitutional governance architecture for autonomous AI agents," *Zenodo (CERN European Organization for Nuclear Research)*. Mar. 25, 2026. doi: 10.5281/zenodo.19223924.
- [33]. [39] T. Miller, "Controlled agentic AI systems: A governance-driven architecture for auditable and reproducible decision pipelines," *Machine Learning and Knowledge Extraction*, vol. 8, no. 5, p. 125, May 2026, doi: 10.3390/make8050125.

- [34]. D. Kaur, S. Uslu, K. J. Rittichier, and A. Durrezi, "Trustworthy artificial intelligence: A review," *ACM Computing Surveys*, vol. 55, no. 2, pp. 1–38, Jan. 2022, doi: 10.1145/3491209.
- [35]. R. Haider, Md. F. I. Shaif, R. Ahmed, N. H. Nafi, M. R. Sumon, and M. Rahman, "The influence of social media branding on consumer purchase behavior: A comprehensive empirical and thematic analysis," *International Journal of Science and Research Archive*, vol. 16, no. 2, pp. 460–470, Aug. 2025, doi: 10.30574/ijrsra.2025.16.2.2354.
- [36]. M. Saleme, "Beyond identity governance: A protocol-level security testing framework for multi-agent AI systems," *Zenodo (CERN European Organization for Nuclear Research)*. Mar. 23, 2026. doi: 10.5281/zenodo.19343034.
- [37]. M. Baduwal and P. Paudel, "Evaluating agentic artificial intelligence: A comprehensive survey of metrics, benchmarks, and methodologies." Institute of Electrical and Electronics Engineers (IEEE), Feb. 27, 2026. doi: 10.36227/techrxiv.177162480.04513202/v2.
- [38]. R. Haider, J. T. Juba, and S. Chowdhury, "Exploring the link between suicidal ideation and digital environments: The hidden impact of marketing content," *International Journal of Science and Research Archive*, vol. 16, no. 2, pp. 607–614, Aug. 2025, doi: 10.30574/ijrsra.2025.16.2.2353.
- [39]. N. Flehmig, M. A. Lundteigen, and S. Yin, "Perspectives on a reliability monitoring framework for agentic AI systems," *arXiv (Cornell University)*. Nov. 12, 2025. doi: 10.48550/arxiv.2511.09178.
- [40]. D. Miruke, "AEGIS algebra: A unified mathematical framework for executable assurance of agentic AI artifacts," *Zenodo (CERN European Organization for Nuclear Research)*, Jun. 2026, doi: 10.5281/zenodo.20813698.
- [41]. H. Gough, "Bounded autonomy: Behavioral specification languages and runtime enforcement architectures for trustworthy agentic AI systems." Wiley, Feb. 11, 2026. doi: 10.22541/au.177083908.89981049/v1.
- [42]. R. Haider, W. A. Megha, J. T. Juba, A. Alamgir, and L. Ahmad, "The conversational revolution in health promotion: Investigating chatbot impact on healthcare marketing, patient engagement, and service reach," *International Journal of Science and Research Archive*, vol. 15, no. 3, pp. 1585–1592, Jun. 2025, doi: 10.30574/ijrsra.2025.15.3.1937.
- [43]. R. N. Khan, D. Joyce, and M. Habiba, "AGENTS SAFE: A unified framework for ethical assurance and governance in agentic AI," *arXiv (Cornell University)*. Dec. 02, 2025. doi: 10.48550/arxiv.2512.03180.
- [44]. N. Díaz-Rodríguez, J. Del Ser, M. Coeckelbergh, M. L. de Prado, E. Herrera-Viedma, and F. Herrera, "Connecting the dots in trustworthy artificial intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation," *Information Fusion*, vol. 99, p. 101896, Jun. 2023, doi: 10.1016/j.inffus.2023.101896.
- [45]. B. Li *et al.*, "Trustworthy AI: From principles to practices," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–46, Aug. 2022, doi: 10.1145/3555803.
- [46]. R. Haider, F. A. I. Bari, Osru, N. Afia, and M. A. Khan Mugdho, "Leveraging internet of things data for real-time marketing: Opportunities, challenges, and strategic implications," *International Journal of Science and Research Archive*, vol. 15, no. 3, pp. 1657–1663, Jun. 2025, doi: 10.30574/ijrsra.2025.15.3.1936.
- [47]. S. A. Adedokun, D. A. Adedokun, B. O. Ishola, R. I. Adeniran, and C. O. Olaleye, "Agentic AI and autonomous decision-making: A review of human-in-the-loop frameworks, oversight mechanisms, and trust calibration," *International Journal of Research and Innovation in Applied Science*, vol. 11, no. 3, pp. 1350–1374, 2026, doi: 10.51584/ijrias.2026.11030104.
- [48]. P. Papanokechi, "Agent epistemic accountability layer: A framework for runtime verification and trust calibration of autonomous AI agent outputs," Apr. 2026. doi: 10.5281/zenodo.19565086.
- [49]. R. Haider, M. Bari, Md. F. I. Shaif, M. Rahman, Md. N. H. Ohi, and K. M. M. Rahman, "Quantifying the impact: Leveraging AI-Powered sentiment analysis for strategic digital marketing and enhanced brand reputation management," *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 1103–1121, May 2025, doi: 10.30574/ijrsra.2025.15.2.1524.
- [50]. X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F. Wang, "From features engineering to scenarios engineering for trustworthy AI: I&I, c&c, and v&v," *IEEE Intelligent Systems*, vol. 37, no. 4, pp. 18–26, Jul. 2022, doi: 10.1109/mis.2022.3197950.
- [51]. [55] R. Haider and J. Sabatina, "Harnessing the power of micro-influencers: A comprehensive analysis of their effectiveness in promoting climate adaptation solutions," *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 595–610, May 2025, doi: 10.30574/ijrsra.2025.15.2.1448.
- [52]. R. Haider, M. Bari, Osru, N. Afia, and T. Al Karim, "Illuminating the black box: Explainable AI for enhanced customer behavior prediction and trust," *International Journal of Science and Research Archive*, vol. 15, no. 3, pp. 247–268, Jun. 2025, doi: 10.30574/ijrsra.2025.15.3.1674.
- [53]. R. Haider, M. Bari, Md. F. I. Shaif, and M. Rahman, "Engineering hyper-personalization: Software challenges and brand performance in AI-driven digital marketing management: An empirical study," *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 1122–1141, May 2025, doi: 10.30574/ijrsra.2025.15.2.1525.
- [54]. A. D. Kaplan, T. T. Kessler, J. C. Brill, and P. A. Hancock, "Trust in artificial intelligence: Meta-Analytic findings," *Human Factors The Journal of the Human Factors and Ergonomics Society*, vol. 65, no. 2, pp. 337–359, May 2021, doi: 10.1177/00187208211013988.
- [55]. E. Glikson and A. W. Woolley, "Human trust in artificial intelligence: Review of empirical research," *Academy of Management Annals*, vol. 14, no. 2, pp. 627–660, Mar. 2020, doi: 10.5465/annals.2018.0057.

- [56]. Y. Zhang, Q. V. Liao, and R. K. E. Bellamy, “Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, New York, NY, USA: ACM, Jan. 2020, pp. 295–305. doi: 10.1145/3351095.3372852.
- [57]. N. Wang, D. V. Pynadath, and S. G. Hill, “Trust calibration within a human-robot team: Comparing automatically generated explanations,” *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 109–116, Mar. 2016, doi: 10.1109/hri.2016.7451741.