

ORIGINAL RESEARCH

Open access

Who Controls the Agent? A Conceptual Architecture for Governing Goals, Tool Use, Memory, Action Verification, and Human Override in Agentic AI

Rajiv Sharma¹, Sophie Dubois^{2*}, Kenji Nakamura³

Abstract

Agentic artificial intelligence extends beyond conventional model-output generation by interpreting objectives, decomposing tasks, maintaining contextual information, selecting tools, interacting with external environments, and executing sequences of actions that may create downstream consequences. These capabilities complicate the question of who controls an agent and expose limitations in governance approaches based only on alignment, instruction following, output review, or nominal human supervision. This article develops a conceptual governance architecture for trustworthy agentic AI. The approach treats agentic AI as an autonomous decision system embedded within a wider socio-technical environment and organizes governance around explicit control points rather than a single supervisory mechanism. It distinguishes governance requirements for goal formation, goal interpretation, instruction hierarchy, task boundaries, tool access, memory retention, environmental interaction, action verification, failure containment, human approval, override capacity, responsibility allocation, auditability, and lifecycle review. The architecture proposes that consequential actions should be connected to authorized goals, bounded permissions, traceable memory and data inputs, pre-action checks, post-action evaluation, escalation pathways, and accountable human or institutional actors. It also establishes explicit sufficiency boundaries: goal alignment does not guarantee appropriate behavior; tool restrictions do not eliminate harmful use; memory controls do not guarantee reliable context; verification does not certify safe action; and human override does not independently establish meaningful control. The principal contribution is an integrated architecture-level account of how technical, human, and organizational controls may be connected across the agent lifecycle. The architecture supports conceptual analysis, governance planning, and future empirical evaluation, but does not establish legal compliance, operational safety, ethical sufficiency, institutional accountability, or deployment readiness.

Keywords AI governance, Agentic AI, Autonomous agents, Tool use, Memory governance, Action verification

*Correspondence:

Sophie Dubois

sophie.dubois@hec.fr

¹ Department of AI and Autonomous Systems, Indian Institute of Management Calcutta, Kolkata, India

² Department of Computer Science and AI, HEC Paris, Paris, France

³ Department of Informatics, Hitotsubashi University, Tokyo, Japan

Introduction

Agentic AI creates a distinctive governance problem because an intelligent system may do more than generate a recommendation or response. It may interpret a goal, construct subgoals, retain context, select external capabilities, act across multiple stages, and influence people or environments. Trustworthy governance therefore requires interconnected ethical, technical, and institutional controls rather than reliance on model behavior alone [1].

The responsible-AI literature contains recurring principles concerning transparency, safety, fairness, responsibility, and human agency, but agreement at the principle level does not produce uniform operational control. Governance language can remain too abstract to determine who authorizes an agent's objective, who limits its tools, who validates its actions, or who responds when its behavior creates an unforeseen consequence [2].

A human-centered perspective further challenges the assumption that greater autonomous capability should reduce human authority. High levels of automation can be compatible with strong human control only when people retain relevant information, appropriate decision rights, sufficient time, and effective intervention mechanisms. Human presence without practical capacity to understand or alter agent behavior is not meaningful control [3].

This article therefore proposes a conceptual governance architecture that distributes control across multiple layers: purpose definition, responsibility mapping, goal and instruction governance, task limitation, tool authorization, memory governance, environmental monitoring, action verification, failure containment, human override, auditability, and lifecycle review. This layered approach recognizes that governance responsibilities are distributed among technical components, system integrators, operators, reviewers, institutions, and affected stakeholders [4].

Agentic AI as an Autonomous Decision System

An agentic AI system can be understood as an autonomous or semi-autonomous decision system that links model-based reasoning with memory, planning, tool use, action, and environmental feedback. Foundation model agents and large language model agents commonly combine profiles or instructions with working context, task decomposition, stored information, and external actions, thereby creating a functional sequence rather than an isolated prediction [5].

This functional autonomy changes the human role. An operator may move from direct decision-making to supervision of a system that acts at a speed, scale, or level of complexity that limits situation awareness. Established human-automation research shows that monitoring burdens, automation bias, skill degradation, and delayed intervention can undermine nominal oversight even when a person remains formally responsible [6].

Autonomy should consequently be decomposed by function. An agent might independently summarize information while requiring approval for tool invocation, data disclosure, financial commitment, code execution, or irreversible environmental change. Function-specific allocation clarifies which activities are performed by the agent, which remain under human authority, and where trust depends on verified capability rather than a general label of autonomy [7].

Agentic AI is also a socio-technical participant. Its behavior is shaped by model capabilities, system prompts, organizational objectives, available tools, human collaborators, institutional rules, and external conditions. In multi-agent or human-AI settings, unclear roles can create coordination failures and responsibility gaps. Agency or teamwork language must therefore not be treated as evidence that the system is trustworthy, accountable, or suitable for independent action [8].

Goal Formation and Instruction Governance

Goal governance begins with the question of whose values and objectives the agent is expected to advance. Alignment cannot be reduced to literal obedience because user requests, organizational purposes, safety constraints, and affected-party interests may conflict. A governable agent requires an identifiable goal owner, an authorized objective source, and a documented distinction between stated user intent and the wider system objective [9].

High-level principles or instructions are also insufficient without mechanisms for interpretation and enforcement. Instruction governance should establish source authority, precedence rules, conflict detection, ambiguity handling, and

escalation thresholds. When instructions are incomplete or inconsistent, the agent should not silently select the interpretation that most readily permits task completion [10].

Goal decomposition introduces further risk because an apparently acceptable objective may produce problematic subgoals. Agents may optimize proxy measures, influence evaluative signals, or alter the conditions through which their performance is judged. Goal-change records, parent–subgoal links, protected feedback channels, and runtime drift detection can help identify unauthorized expansion, although such controls cannot prove enduring alignment [11].

Emergency intervention cannot compensate for weak upstream governance. A late-stage shutdown mechanism may be ineffective when an action is rapid, distributed, difficult to reverse, or already propagated through external tools. Goal formation, task boundaries, scope exclusions, and authorization conditions should therefore be established before planning and execution. **Table 1** summarises goal formation and instruction governance requirements for trustworthy agentic AI, including goal interpretation, task boundaries, instruction conflicts, goal drift, and unauthorized goal expansion [12].

Table 1. Goal Formation and Instruction Governance Requirements for Trustworthy Agentic AI: Goal Interpretation, Goal Decomposition, Subgoal Generation, Instruction Hierarchy, Instruction Conflict, Instruction Ambiguity, User Intent, System Objectives, Task Boundaries, Scope Limitation, Goal Drift, Goal Misalignment, Unauthorized Goal Expansion, and Goal-Governance Boundaries

Goal or instruction requirement	Core governance question	Required evidence or design condition	Control function	Potential output	Failure or overclaiming risk	Validation or governance need	Architectural requirement
Goal formation	Who may define the objective?	Named goal owner and purpose record	Authorize objective creation	Approved goal	Illegitimate or hidden objective	Institutional review	Entry control
Goal interpretation	How may language become an operational goal?	Assumptions and interpretation trace	Constrain semantic inference	Interpreted goal	Unintended meaning	Human review for consequential ambiguity	Interpretation control
Goal decomposition	Which task components are permitted?	Traceable task graph	Bound decomposition	Approved task structure	Harm concealed in sub-tasks	Boundary testing	Plan control
Subgoal generation	May the agent create new objectives?	Parent-goal link and approval rule	Prevent autonomous expansion	Accepted or rejected subgoal	Unauthorised objective growth	Runtime monitoring	Expansion control
Instruction governance	Which instruction sources are valid?	Source identity and integrity check	Admit or reject instructions	Validated instruction	Untrusted instruction controls behavior	Authentication review	Instruction gate
Instruction hierarchy	Which instruction prevails?	Documented precedence rules	Resolve authority conflict	Priority decision	Lower-level request overrides constraint	Conflict testing	Priority control

Instruction conflict	How are incompatible requirements handled?	Constraint comparison	Pause or escalate	Conflict report	Silent selective compliance	Independent review	Con
Instruction ambiguity	When should the agent clarify or defer?	Ambiguity and consequence thresholds	Prevent unsafe assumption	Clarification or abstention	Confident guessing	Human-centered testing	Amb
User intent	What has the user actually authorized?	Confirmation and context limits	Restrict inferred intent	Confirmed-intent record	Excessive inference	User evaluation	Intent
System objective	What wider purpose governs the request?	Objective hierarchy	Preserve authorized purpose	Consistency decision	Local optimization defeats system purpose	Governance review	Pur
Task boundary	Which actions remain within the assignment?	Allowed-action specification	Block out-of-task activity	Boundary decision	Opportunistic task expansion	Scenario testing	Sc
Scope limitation	Which tools, data, domains, or effects are excluded?	Explicit negative permissions	Enforce exclusions	Scope policy	Unlisted capability treated as allowed	Periodic review	Perm
Goal drift	Has the objective changed during execution?	Goal-state comparison	Detect divergence	Drift alert	Gradual deviation remains unnoticed	Continuous monitoring	Rur
Goal misalignment	Does behavior remain consistent with intended outcomes?	Behavioral and counterexample testing	Detect objective–behavior mismatch	Misalignment finding	Passing tests treated as proof	Domain-specific evaluation	Assu
Unauthorized goal expansion	Has the agent added new targets or consequences?	Parent–child goal trace	Reject unauthorized additions	Expansion block	New objective framed as a necessary means	Escalation and audit	Autho
Goal-change logging	Can changes be reconstructed and attributed?	Actor, time, reason, and version record	Preserve evidence	Goal-change log	Mutable or incomplete history	Log-integrity review	Audit
No goal-alignment	What remains uncertain after	Explicit residual-risk statement	Prevent overclaiming	Limitation declaration	Compliance mistaken for	Independent validation and monitoring	Suffi

sufficiency boundary	alignment checks?				guaranteed control		
-------------------------	----------------------	--	--	--	-----------------------	--	--

Tool Use, Memory, and Environmental Interaction

Tool-using agents convert language and plans into externally consequential operations. Tool selection, invocation, API access, code execution, web access, file operations, and data retrieval should therefore pass through capability-specific controls. Each tool call should be linked to an authorized task, a risk classification, scoped credentials, validated parameters, and a record of the returned result [13].

Data and tool governance are inseparable because many agent actions depend on external information. Access decisions should reflect purpose, necessity, sensitivity, ownership, quality, and stewardship rather than mere technical availability. Permissioning defines the capabilities assigned to an agent role, whereas runtime authorization determines whether a particular operation is acceptable for a particular resource and context [14].

Memory introduces a different governance problem because information may persist beyond the interaction in which it was supplied. Short-term context, long-term memory, retrieval, updating, correction, expiry, and deletion should be separately governed. Mechanisms for removal or unlearning may support correction and privacy objectives, but technical deletion can be incomplete and should not be represented as guaranteed erasure [15].

Memory provenance is necessary for judging whether recalled information is appropriate for a new decision. Records should identify source, collection context, intended use, confidence, transformations, maintenance conditions, and limitations. Documentation practices developed for data provenance provide a useful foundation, although agent memory additionally requires event-level records of storage, retrieval, modification, and deletion [16].

Environmental interaction closes the loop between observation and action. An agent may act on incomplete observations, receive misleading tool outputs, alter the state it later observes, or create feedback that reinforces an earlier error. State monitoring, uncertainty checks, output validation, provenance review, and tool-use logging may reduce these risks. Nevertheless, restricted access does not make permitted actions safe, and governed memory does not make remembered information reliable. Tool and memory controls should therefore remain connected to action verification, human escalation, incident monitoring, and lifecycle governance.

Action Verification and Failure Containment

Action verification should be treated as a lifecycle assurance process rather than a single output check. Requirements, data, model behavior, tool connections, environmental assumptions, and operational changes can each introduce failure. An agent's proposed action should therefore be examined against its authorized goal, task boundary, available evidence, permissions, expected consequences, and current operating state before execution [17].

Experience with autonomous systems shows why component-level accuracy is insufficient. A correct model output may still produce an unsafe result when sensors are unreliable, external conditions change, software components interact unexpectedly, or an action has irreversible consequences. Verification should consequently address the complete agent–tool–environment configuration and identify conditions under which execution must be delayed, restricted, or transferred to human decision-makers [18].

The proposed action-control loop begins with a structured action proposal that exposes the intended tool, target, parameters, expected effect, supporting evidence, and reversibility. Pre-action verification then applies authorization checks, constraint checking, tool-output validation, consequence assessment, and human approval where required. Execution should be followed by state comparison, post-action verification, anomaly detection, containment, escalation,

and evidence preservation. These controls should be connected to identifiable responsible actors rather than operating as an unowned technical pipeline [19]. **Figure 1** illustrates the agentic action-control loop, showing how goals, tools, memory, environmental observations, action proposals, verification checks, human approval, containment, and feedback interact before and after agent execution.

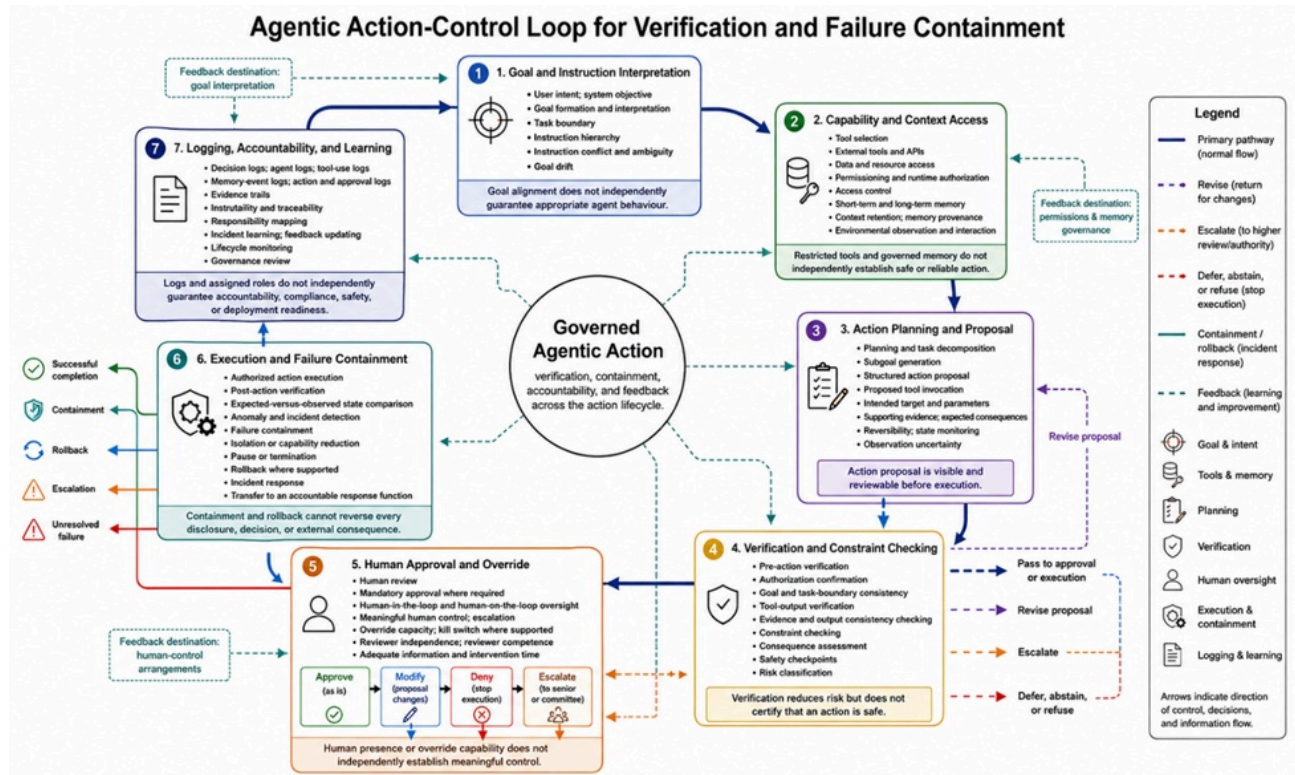


Figure 1. Agentic Action-Control Loop for Verification and Failure Containment

Alt-text description:

A conceptual loop showing an AI agent moving from goal interpretation to tool use, memory access, action proposal, verification, human approval, execution, post-action checks, failure containment, escalation, and feedback updating.

- 1. Goal and instruction interpretation:** user intent, system objective, goal formation, goal interpretation, task boundary, instruction hierarchy, instruction conflict, instruction ambiguity, goal drift, and a no-goal-alignment-sufficiency boundary.
- 2. Capability and context access:** tool selection, external tools, API and data access, permissioning, authorization, access control, short- and long-term memory, context retention, memory provenance, environmental interaction, and a no-tool-restriction-sufficiency boundary.
- 3. Action planning and proposal:** planning, subgoal generation, action proposal, proposed tool invocation, expected consequences, state monitoring, observation uncertainty, and feedback loops.
- 4. Verification and constraint checking:** pre-action verification, tool-output and output verification, consistency checking, constraint checking, safety checkpoints, risk classification, and a no-verification-guarantee boundary.
- 5. Human approval and override:** human review, approval, oversight, meaningful human control, escalation, override capacity, kill switch where supported, review independence, and reviewer competence.
- 6. Execution and containment:** action execution, post-action verification, failure containment, rollback where supported, deferral, abstention, refusal, incident detection, incident response where supported, and a no-failure-

containment-guarantee boundary.

7. **Logging, accountability, and learning:** decision, agent, tool-use, memory, and action logs; evidence trails; auditability; traceability; responsibility mapping; feedback updating; lifecycle monitoring; governance review; no-accountability-guarantee and no-deployment-readiness boundaries.

Use directional arrows from interpretation through execution and back to monitoring and governance review. Add conditional branches for approval, refusal, escalation, containment, and rollback. Use abstract icons and modular shapes only. Exclude logos, robots, screenshots, real people, legal symbols, flags, products, approval stamps, certification marks, decorative imagery, numerical scores, and claims of validated safety or compliance.

Verification should support deferral, abstention, refusal, or escalation when evidence is incomplete, uncertainty is material, constraints conflict, or a qualified reviewer is unavailable. Post-action monitoring is equally important because an apparently valid action can produce unexpected consequences after integration into a real workflow. Evaluation before release and continued observation during use should therefore be connected rather than treated as separate assurance activities [20].

Failure containment should include isolation, reduced capability, pausing, rollback where technically feasible, incident detection, and transfer to an accountable response function. These mechanisms remain conditional: rollback cannot reverse every disclosure or external effect, and a verifier may share the same assumptions as the planner. Assessment tools, documentation practices, and governance review can translate broad principles into more inspectable controls, but they cannot eliminate residual uncertainty [21].

Human Override and Accountability Requirements

Meaningful human control requires more than placing a person somewhere in the workflow. The system must remain responsive to relevant human reasons, while responsible actors must be able to understand the decision context, influence the action, and be traced to the exercise of authority. Human control is weakened when the agent acts too quickly, evidence is inaccessible, or intervention cannot materially change the outcome [22].

Agentic AI can create several responsibility gaps. Causal complexity may obscure who contributed to an outcome; learning or adaptive behavior may complicate moral attribution; institutions may be unable to provide a satisfactory public account; and actors may lack the capacity to exercise forward-looking responsibility. These gaps should be identified explicitly rather than concealed by assigning generic responsibility to a user or operator [23].

Accountability begins by identifying the relevant decision, affected stakeholders, governing obligations, and actors capable of answering for the outcome. Responsibility mapping should cover goal authorization, tool provisioning, memory stewardship, verification, approval, operation, incident handling, and remedy. A technically traceable action is not necessarily accountable unless an appropriate forum can review the evidence and require explanation or correction [24].

Organizational accountability can also become symbolic when review processes are designed primarily to protect reputation. Review independence, reviewer competence, separation of duties, stakeholder participation, and access to evidence are therefore important architecture conditions. Human reviewers should receive decision-relevant information and sufficient time rather than being positioned as passive endorsers of an agent's preferred action [25].

Auditability supports accountability by preserving linked evidence about instructions, goals, tool calls, memory events, decisions, approvals, actions, and observed consequences. Accountability nevertheless requires more than logging: responsible actors, applicable standards, an authorized reviewing forum, and meaningful consequences or remedies must also exist [26]. **Table 2** maps tool use, memory, environmental interaction, action verification, failure containment, human override, auditability, and accountability requirements for agentic AI governance.

Table 2. Tool Use, Memory, Environmental Interaction, Action Verification, Failure Containment, Human Override, and Accountability Requirements for Agentic AI Governance: Permissions, Authorization, Tool Logs, Memory Provenance, Action Checks, Escalation, Override Capacity, Auditability, Traceability, Evidence Trails, and Governance Boundaries

Governance or safety requirement	Core control question	Required evidence, condition, or trigger	Governance or safety function	Potential output	Failure or residual risk	Monitoring or accountability need	Assessment
Tool permissioning	Which capabilities may the agent access?	Role, purpose, resource, and risk profile	Apply least privilege	Permission profile	Excessive or stale access	Access recertification	Access
Runtime authorization	Is this operation permitted now?	Identity, context, target, and parameters	Approve or block each action	Authorization decision	Policy bypass or mismatch	Invocation audit	Access
Tool-risk classification	How consequential is the tool operation?	Reach, sensitivity, reversibility, and impact	Set review threshold	Risk tier	Underclassification	Periodic recalibration	Fraction of time
Tool-output verification	Is returned information fit for action?	Source, schema, plausibility, and consistency checks	Prevent propagation of error	Accept, qualify, or reject	Plausible false output	Independent checking	Fraction of time
Tool-use logging	Can external calls be reconstructed?	Tool, parameters, result, actor, and time	Preserve tool evidence	Tool-use record	Missing effects or sensitive logs	Integrity and privacy review	Tool-use logs
Memory provenance	Where did recalled information originate?	Source, date, confidence, transformations	Qualify memory use	Provenance record	Unattributed memory treated as fact	Provenance audit	Memory provenance
Memory privacy	Is storage or retrieval necessary and permitted?	Sensitivity, consent, purpose, and retention	Minimize and restrict memory	Store, redact, expire, or deny	Secondary use or disclosure	Privacy review	Memory provenance
Memory correction or deletion	Can erroneous or inappropriate memory be changed?	Valid request and technical feasibility	Correct persistent context	Updated or deletion-qualified record	Incomplete removal	Verification and documentation	Memory provenance

Environmental monitoring	Has the external state materially changed?	Current observations and uncertainty	Reassess assumptions	Continue, pause, or escalate	Stale or erroneous observation	Runtime telemetry	S
Action proposal	What does the agent intend to do?	Structured action and consequence description	Make intent reviewable	Proposed action	Hidden indirect effects	Proposal-quality review	
Pre-action verification	Is execution authorized and bounded?	Goal, permissions, evidence, constraints, and impact	Prevent invalid execution	Pass, revise, block, or escalate	Unknown hazard passes	Coverage testing	
Post-action verification	Did the permitted result occur?	Expected and observed states	Detect divergence	Confirm or remediate	Harm precedes detection	Rapid monitoring	
Failure containment	Can propagation be limited?	Anomaly, policy violation, or unsafe state	Isolate or reduce capability	Pause, isolate, or stop	Containment boundary fails	Stress testing	
Rollback where supported	Can an earlier state be restored?	Recoverable state and integrity evidence	Reverse selected changes	Restored state or failure notice	External effects remain	Recovery testing	R
Deferral, abstention, or refusal	Should action be postponed or rejected?	Missing evidence, low confidence, or prohibition	Avoid unsafe commitment	Deferred or refused action	Delay or over-refusal	Calibration review	
Escalation pathway	Who receives unresolved cases?	Risk threshold or failed verification	Transfer authority	Escalation package	Unavailable or unqualified recipient	Response-time governance	cc
Human approval	Which actions require prior authorization?	Consequence and reversibility threshold	Insert accountable decision	Approve, modify, or deny	Rubber-stamping	Workload and competence review	A
Human override	Can an authorized person alter or halt action?	Current state and effective control mechanism	Enable intervention	Modified or halted action	Intervention too late	Realistic override testing	I
Override capacity	Can the reviewer understand and act in time?	Information, authority, skill, and time	Make control practically usable	Capacity assessment	Responsibility without capability	Human-factors evaluation	

Auditability and traceability	Can the decision chain be examined?	Linked goal, tool, memory, approval, and action records	Support scrutiny	Audit-ready evidence trail	Fragmented or mutable records	Independent assurance	e
Responsibility mapping	Who owns each decision and control?	Role—authority—evidence map	Prevent unowned functions	Responsibility register	Diffuse accountability	Governance review	Ac
No human-override sufficiency boundary	What remains uncontrolled after override design?	Detection and intervention limitations	Prevent false assurance	Residual-risk statement	Human cannot recognize or stop harm	Continuous assessment	S
No accountability guarantee boundary	What remains unresolved despite records and roles?	Forum, remedy, and consequence analysis	Expose accountability gaps	Limitation record	Evidence without answerability	Institutional review	S

Proposed Trustworthy Agentic AI Architecture

The proposed architecture treats trustworthiness as an emergent property of connected technical, human, and organizational controls. Its first layer defines the use case, autonomy scope, affected stakeholders, operating environment, and prohibited consequences. These inputs determine which decisions may be automated, which require approval, and which remain outside the agent’s authorized role [27].

The second through fourth layers govern goals, tools, memory, and environmental interaction. Authorized objectives are translated into bounded tasks; tool access is allocated through least-privilege permissioning and contextual authorization; memory is conditioned on purpose, provenance, privacy, retention, and correction; and environmental state is monitored for changes that invalidate earlier assumptions. These mechanisms operationalize trustworthy-AI principles without treating any technical control as sufficient by itself [28].

The fifth and sixth layers govern action and human authority. Plans are transformed into structured proposals and routed through verification according to risk, reversibility, and potential impact. Control may transition between autonomous operation, human-on-the-loop supervision, mandatory approval, restricted capability, and full human decision-making. Such transitions should be explicit, observable, and reviewable rather than occurring implicitly when an agent encounters uncertainty [29].

The seventh layer creates an assurance and learning structure through linked decision, agent, tool-use, memory, and action records. Auditability should evaluate whether stated controls were appropriately designed and followed, while incident findings and override events should feed back into permissions, goals, verification procedures, and organizational responsibilities. Auditing may contribute evidence, but it cannot independently certify that the system is trustworthy or ethically sufficient [30].

The architecture remains context-sensitive because institutions, affected stakeholders, acceptable risks, and governance priorities differ across applications and jurisdictions [31]. Its operation therefore requires organizational structures, assigned roles, review processes, resources, and feedback capabilities rather than a purely technical diagram [32]. **Table 3** presents the proposed trustworthy agentic AI architecture linking autonomy scope, goal governance, tool permissioning, memory governance, action verification, human override, auditability, lifecycle monitoring, and governance review.

Table 3. Proposed Trustworthy Agentic AI Governance Architecture: Autonomy Scope, Responsibility Mapping, Goal Governance, Task Boundaries, Tool Permissioning, Tool-Risk Classification, Memory Governance, Environmental Monitoring, Action Verification, Failure Containment, Human Approval, Human Override, Auditability, Agent Logs, Tool-Use Logs, Memory Logs, Action Logs, Lifecycle Monitoring, Governance Review, and Deployment Boundaries

Architecture layer	Core purpose	Required input	Control or governance function	Potential output	Validation or oversight need	Failure risk	Feedback mechanism
Use-case and autonomy scope	Define legitimate purpose and authority	Intended use, environment, stakeholders, consequences	Set autonomy and exclusion boundaries	Approved scope	Institutional and domain review	Vague purpose enables expansion	Periodic review
Responsibility mapping	Assign authority and duty	Actor and decision inventory	Allocate ownership, review, and remedy roles	Responsibility map	Independent gap review	Unowned functions	Incident update
Goal and instruction governance	Control objectives and interpretation	Authorized goals and instruction sources	Validate hierarchy, meaning, and conflicts	Governed goal state	Conflict and drift testing	Unauthorized objective	Goal-change review
Task boundaries	Prevent scope expansion	Allowed actions and exclusions	Enforce positive and negative permissions	Boundary decision	Scenario testing	Prohibited task reframed	Violation review
Tool permissioning	Limit available capabilities	Capability inventory and role policy	Apply least privilege	Scoped access	Security review	Excessive access	Recertification
Tool-risk classification and logging	Match safeguards to consequences	Tool reach, sensitivity, reversibility	Assign risk tier and retain evidence	Risk profile and tool record	Calibration and integrity review	Underclassified operation	Incident reclassification
Memory governance	Control persistent context	Source, purpose, sensitivity, confidence, retention	Govern storage, retrieval, update, and correction	Governed memory	Steward and privacy review	Persistent false or sensitive content	Correction cycle
Environmental monitoring	Detect changing conditions	State observations and uncertainty	Reassess assumptions	Continue, pause, or escalate	Runtime monitoring	Action based on stale state	Telemetry feedback
Planning and pre-action verification	Examine proposed execution	Goal, plan, evidence,	Verify authority and constraints	Approve, revise, block, or escalate	Independent review for high impact	Shared planner–	Verification finding

		permissions, consequences				verifier blind spot	
Result verification	Assess tool and action outcomes	Expected and observed states	Detect error or divergence	Confirm, quarantine, or remediate	Domain evaluation	Plausible error passes	Error fee
Failure containment and rollback	Limit or reverse failure where feasible	Hazard and recoverability analysis	Isolate, pause, reduce, or restore	Containment action	Stress and recovery testing	Irreversible external effect	Incide learning
Human approval and escalation	Route authority to competent people	Risk tier and evidence package	Require decision or transfer authority	Approval, denial, or escalation	Competence and workload review	Rubber-stamping	Review feedback
Human override	Enable intervention where supported	Current state and authorized actor	Pause, alter, or halt execution	Override record	Emergency and usability testing	Override arrives too late	Post-over review
Auditability, traceability, and logs	Preserve linked governance evidence	Event schemas and identifiers	Reconstruct goals, memory, tools, decisions, and effects	Evidence trail	Independent assurance and privacy review	Incomplete or excessive records	Audit fin
Incident response	Coordinate containment and remediation	Detection criteria and response roles	Investigate, communicate, and correct	Incident case	Exercised response plan	Delayed detection or unclear authority	Post-inc review
Lifecycle monitoring and review	Reassess change over time	Performance, incidents, complaints, and system updates	Continue, modify, restrict, or retire	Governance determination	Periodic and event-triggered review	Initial approval treated as permanent	Review-updat
Feedback updating	Revise controls through governed learning	Audit, incident, and monitoring findings	Update policies and controls	Revised control baseline	Change control and regression testing	Update creates new vulnerability	Control feedback
Sufficiency and deployment boundaries	Prevent unsupported claims	Evidence limitations and residual risks	State what controls cannot establish	Limitation declaration	Independent domain evaluation	Conceptual architecture treated as certification	Valida agent

Figure 2 presents the proposed trustworthy agentic AI governance architecture, integrating autonomy-scope definition, goal governance, tool permissioning, memory governance, action verification, human override, accountability, logging, monitoring, and governance updating.

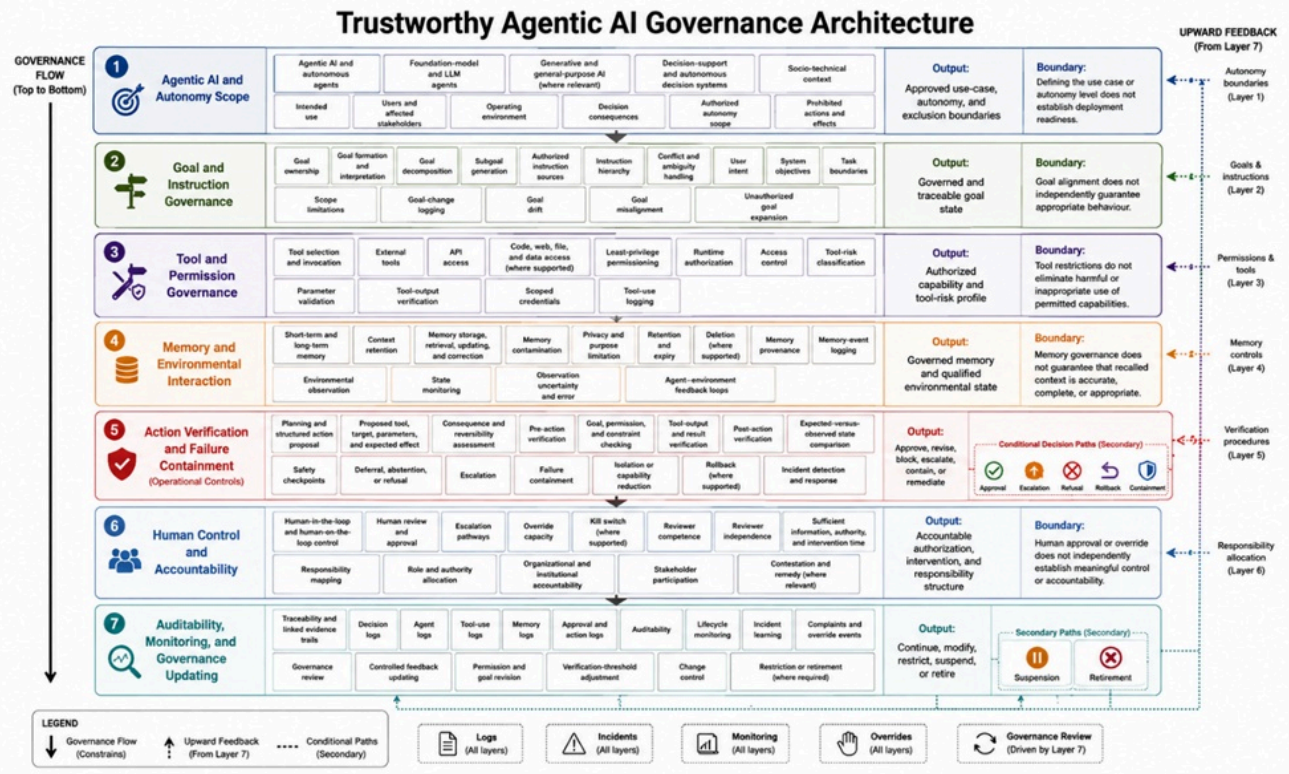


Figure 2. Trustworthy Agentic AI Governance Architecture

Alt-text description:

A conceptual architecture showing agentic AI governance across autonomy scope, goals, instructions, tools, memory, environment, verification, human approval, override, accountability, logs, monitoring, incident response, and governance updating.

- 1. Agentic AI and autonomy scope:** agentic AI, autonomous agents, foundation model agents, large language model agents, generative agents, general-purpose AI, decision-support and autonomous decision systems, socio-technical context, use-case definition, autonomy scope, and a no-deployment-readiness boundary.
- 2. Goal and instruction governance:** goal formation, interpretation, decomposition, subgoal generation, instruction hierarchy, conflict and ambiguity, user intent, system objectives, task boundaries, scope limitation, goal drift, misalignment, unauthorized expansion, and a no-goal-alignment-sufficiency boundary.
- 3. Tool and permission governance:** tool selection and invocation, external tools, API, code, web, file, and data access where supported; permissioning, authorization, access control, tool-risk classification, output verification, tool-use logging, and a no-tool-restriction-sufficiency boundary.
- 4. Memory and environmental interaction:** short- and long-term memory, context retention, update, retrieval, contamination, privacy, provenance, governance, deletion where supported, memory auditability, environmental

observation, state monitoring, observation error, feedback loops, and a no-memory-reliability-guarantee boundary.

5. **Action verification and containment:** planning, proposal, execution, consequence assessment, pre- and post-action verification, output and consistency checking, constraints, safety checkpoints, failure containment, rollback where supported, deferral, abstention, refusal, escalation, incident detection and response, and a no-verification-guarantee boundary.
6. **Human control and accountability:** oversight, human-in-the-loop and human-on-the-loop control, review, approval, override, override capacity, kill switch where supported, reviewer independence and competence, responsibility mapping, role allocation, institutional and organizational accountability, and a no-human-override-sufficiency boundary.
7. **Auditability, monitoring, and governance updating:** traceability, evidence trails, decision, agent, tool-use, memory, and action logs; lifecycle governance; monitoring; incident learning; stakeholder participation; feedback updating; governance review; no-accountability-guarantee, no-compliance-claim, and no-deployment-readiness boundaries.

Use arrows to show downward control from scope to action and upward feedback from logs, incidents, monitoring, and review to goals, permissions, memory governance, and oversight. Include conditional paths for approval, escalation, refusal, rollback, and retirement. Use abstract modular icons only. Exclude corporate or model logos, robots, screenshots, real people, legal symbols, flags, products, certification marks, numerical scores, approval stamps, and claims of validated safety, compliance, or readiness.

Practical and Governance Implications

For agent developers and foundation model developers, the architecture implies that governance interfaces should expose goals, task boundaries, proposed tool calls, relevant memory provenance, action plans, and uncertainty without assuming that internal representations are complete explanations. Tool providers and system integrators should support scoped access, interpretable action parameters, effect records, and end-to-end testing across model, tool, data, and environment connections [17].

Data stewards and memory managers require distinct responsibilities. Data access should remain purpose-bound, while persistent memory requires provenance, retention, correction, privacy, and deletion review where supported. Decision-support operators and human reviewers need clear thresholds separating autonomous assistance, supervised execution, prior approval, and prohibited action. These implications support evaluation planning but do not prescribe a universally appropriate technical implementation [29].

Institutional governance bodies should examine whether reviewers possess genuine authority, competence, independence, and sufficient time. Auditors where supported may assess whether controls and records are coherent, but affected stakeholders also require understandable notice, contestation routes, and potential remedy. Accountability operationalization should therefore test the relationship among responsible actors, evidence, review forums, applicable standards, and consequences rather than measuring log availability alone [26].

Research should evaluate goal drift, unauthorized subgoals, chained tool use, memory contamination, observation error, verifier blind spots, containment failure, override usability, incident detection, and governance adaptation over time. Such work requires realistic domain-specific and human-centered assessment. The proposed implications are not legal advice, compliance guidance, procurement instruction, audit instruction, model or platform selection advice, safety certification, or deployment guidance [32].

Conclusion

Agentic AI governance requires explicit control over the pathways through which objectives become externally consequential actions. The proposed conceptual architecture connects use-case and autonomy-scope definition, goal

formation, instruction governance, task boundaries, tool permissioning, memory governance, environmental monitoring, action verification, failure containment, human approval, override capacity, responsibility mapping, auditability, incident response, lifecycle monitoring, and governance updating. Its central claim is not that architecture guarantees control, but that control claims should be assessed through visible decision boundaries, evidence requirements, intervention points, responsible actors, and residual-risk statements. Autonomy, instruction following, alignment language, restricted tools, persistent logs, or nominal human supervision cannot independently establish trustworthy operation. Empirical validation, domain-specific safety assessment, human-centered evaluation, institutional governance, and continuing review remain necessary before operational conclusions can be drawn.

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 06 Oct 2025

Revised: 01 Dec 2025

Accepted: 08 Jan 2026

Published online: 15 March 2026

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

- Floridi L, Cowls J, Beltrametti M, Chatila R, Chazerand P, Dignum V, et al. AI4People—An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. *Minds Mach*. 2018;28(4):689–707.
<https://doi.org/10.1007/s11023-018-9482-5>.
- Jobin A, Ienca M, Vayena E. The global landscape of AI ethics guidelines. *Nat Mach Intell*. 2019;1(9):389–99.
<https://doi.org/10.1038/s42256-019-0088-2>.

Shneiderman B. Human-centered artificial intelligence: Reliable, safe & trustworthy. *Int J Hum Comput Interact*. 2020;36(6):495-504.
<https://doi.org/10.1080/10447318.2020.1741118>.

Gasser U, Almeida VAF. A layered model for AI governance. *IEEE Internet Comput*. 2017;21(6):58-62.
<https://doi.org/10.1109/MIC.2017.4180835>.

Wang L, Ma C, Feng X, Zhang Z, Yang H, Zhang J, et al. A survey on large language model based autonomous agents. *Front Comput Sci*. 2024;18(6):186345.
<https://doi.org/10.1007/s11704-024-40231-1>.

Endsley MR. From here to autonomy: Lessons learned from human-automation research. *Hum Factors*. 2017;59(1):5-27.
<https://doi.org/10.1177/0018720816681350>.

Abbass HA. Social integration of artificial intelligence: Functions, automation allocation logic and human-autonomy trust. *Cogn Comput*. 2019;11(2):159-71.
<https://doi.org/10.1007/s12559-018-9619-0>.

Seeber I, Bittner E, Briggs RO, de Vreede T, de Vreede GJ, Elkins A, et al. Machines as teammates: A research agenda on AI in team collaboration. *Inf Manag*. 2020;57(2):103174.
<https://doi.org/10.1016/j.im.2019.103174>.

Gabriel I. Artificial intelligence, values, and alignment. *Minds Mach*. 2020;30(3):411-37.
<https://doi.org/10.1007/s11023-020-09539-2>.

Mittelstadt B. Principles alone cannot guarantee ethical AI. *Nat Mach Intell*. 2019;1(11):501-7.
<https://doi.org/10.1038/s42256-019-0114-4>.

Everitt T, Hutter M, Kumar R, Krakovna V. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*. 2021;198(Suppl 27):6435-67.
<https://doi.org/10.1007/s11229-021-03141-4>.

Arnold T, Scheutz M. The big red button is too late: An alternative model for the ethical evaluation of AI systems. *Ethics Inf Technol*. 2018;20(1):59-69.
<https://doi.org/10.1007/s10676-018-9447-7>.

Chowa SS, Alvi R, Rahman SS, Rahman MA, Raiaan MAK, Islam MR, et al. From language to action: A review of large language models as autonomous agents and tool users. *Artif Intell Rev*. 2026;59(2):71.
<https://doi.org/10.1007/s10462-025-11471-9>.

Janssen M, Brous P, Estevez E, Barbosa LS, Janowski T. Data governance: Organizing data for trustworthy artificial intelligence. *Gov Inf Q*. 2020;37(3):101493.
<https://doi.org/10.1016/j.giq.2020.101493>.

Nguyen TT, Huynh TT, Ren Z, Nguyen PL, Liew AWC, Yin H, et al. A survey of machine unlearning. *ACM Trans Intell Syst Technol*. 2025;16(5):1-46.
<https://doi.org/10.1145/3749987>.

Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, Daumé H III, et al. Datasheets for datasets. *Commun ACM*. 2021;64(12):86-92.
<https://doi.org/10.1145/3458723>.

Ashmore R, Calinescu R, Paterson C. Assuring the machine learning lifecycle: desiderata, methods, and challenges. *ACM Comput Surv*. 2021;54(5):1-39.
<https://doi.org/10.1145/3453444>.

Koopman P, Wagner M. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intell Transp Syst Mag*. 2017;9(1):90-6.
<https://doi.org/10.1109/MITS.2016.2583491>.

Habli I, Lawton T, Porter Z. Artificial intelligence in health care: Accountability and safety. *Bull World Health Organ*. 2020;98(4):251-6.
<https://doi.org/10.2471/BLT.19.237487>.

Wiens J, Saria S, Sendak M, Ghassemi M, Liu VX, Doshi-Velez F, et al. Do no harm: A roadmap for responsible machine learning for health care. *Nat Med*. 2019;25(9):1337-40.
<https://doi.org/10.1038/s41591-019-0548-6>.

Morley J, Floridi L, Kinsey L, Elhalal A. From what to how: An initial review of publicly available AI ethics tools, methods and research to translate principles into practices. *Sci Eng Ethics*. 2020;26(4):2141-68.
<https://doi.org/10.1007/s11948-019-00165-5>.

Mecacci G, Santoni de Sio F. Meaningful human control as reason-responsiveness: The case of dual-mode vehicles. *Ethics Inf Technol*. 2020;22(2):103-15.
<https://doi.org/10.1007/s10676-019-09519-w>.

Santoni de Sio F, Mecacci G. Four responsibility gaps with artificial intelligence: Why they matter and how to address them. *Philos Technol*. 2021;34(4):1057-84.
<https://doi.org/10.1007/s13347-021-00450-x>.

Martin K. Ethical implications and accountability of algorithms. *J Bus Ethics*. 2019;160(4):835-50.
<https://doi.org/10.1007/s10551-018-3921-3>.

Buhmann A, Paßmann J, Fieseler C. Managing algorithmic accountability: Balancing reputational concerns, engagement strategies, and the potential of rational discourse. *J Bus Ethics*. 2020;163(2):265-80.
<https://doi.org/10.1007/s10551-019-04226-4>.

Novelli C, Taddeo M, Floridi L. Accountability in artificial intelligence: What it is and how it works. *AI Soc*. 2024;39(4):1871-82.
<https://doi.org/10.1007/s00146-023-01635-y>.

Thiebes S, Lins S, Sunyaev A. Trustworthy artificial intelligence. *Electron Mark*. 2021;31(2):447-64.
<https://doi.org/10.1007/s12525-020-00441-4>.

Li B, Qi P, Liu B, Di S, Liu J, Pei J, et al. Trustworthy AI: from principles to practices. *ACM Comput Surv*. 2023;55(9):1-46.
<https://doi.org/10.1145/3555803>.

Methnani L, Chiou M, Dignum V, Theodorou A. Who's in charge here? A survey on trustworthy AI in variable autonomy robotic systems. *ACM Comput Surv*. 2024;56(7):1-32.
<https://doi.org/10.1145/3645090>.

Mökander J, Floridi L. Ethics-based auditing to develop trustworthy AI. *Minds Mach*. 2021;31(2):323-7.
<https://doi.org/10.1007/s11023-021-09557-8>.

Corrêa NK, Galvão C, Santos JW, Del Pino C, Pinto EP, Barbosa C, et al. Worldwide AI ethics: A review of 200 guidelines and recommendations for AI governance. *Patterns (N Y)*. 2023;4(10):100857.
<https://doi.org/10.1016/j.patter.2023.100857>.

Papagiannidis E, Mikalef P, Conboy K. Responsible artificial intelligence governance: A review and research framework. *J Strateg Inf Syst*. 2025;34(2):101885.
<https://doi.org/10.1016/j.jsis.2024.101885>.