VORDAN

THE ACCOUNTABILITY GAP

VORDAN FRAMEWORK STANDARD

Agentic Accountability

V 0.2 JUNE 2026

V 0.1 MAY 2026

The harness captures what the agent did.
This defines what sufficient looks like.

VORDAN.CO/BASELINE AAB V0.2

VORDAN

THE ACCOUNTABILITY GAP

VORDAN FRAMEWORK STANDARD

Agentic Accountability

V 0.2 JUNE 2026

V 0.1 MAY 2026

The harness captures what the agent did.
This defines what sufficient looks like.

VORDAN.CO/BASELINE AAB V0.2

CONTENTS

VORDAN AGENTIC ACCOUNTABILITY BASELINE | AAB V0.1

PREAMBLE

Why this document exists

A new category of enterprise risk is forming faster than the governance vocabulary to describe it.

Autonomous AI agents are being deployed into production environments today. They act. They access data. They call tools. They hand work to other agents. They make decisions that carry organizational consequences. And in most enterprises deploying them, no one has defined what accountability looks like for any of it.

This is not a technology problem. The infrastructure layer is maturing rapidly. Agent identity systems, runtime policy enforcement, memory governance tooling, activity logging. The harness that captures what an agent does is being built, funded, and deployed. The technology will arrive.

The accountability gap is not in the harness. It is in what the harness is required to prove.

A log that records every action an agent took does not constitute accountability. Accountability requires that every action be traceable to a human authorization event, that every permission be bounded to a specific task, that every handoff carry the authorization chain forward, and that the whole of it be reconstructible by the organization without dependence on a vendor. A log that satisfies none of those conditions is observation. It is not governance.

This document defines the minimum conditions under which an enterprise can claim its agentic AI deployment is accountable by design.

It is not a technical specification. It does not prescribe tooling, architecture, or vendor selection. It is vendor-neutral by design, because the moment an accountability standard is authored by someone selling the infrastructure it covers, it is no longer an accountability standard.

It is not a compliance framework. It does not map to a regulatory requirement or create one. It is a governance standard: a definition of what sufficient looks like, authored independently, so that enterprises, auditors, regulators, and practitioners have a common reference point when they ask whether an agentic deployment closes the accountability gap or merely generates data.

Version 0.1 establishes the seven accountability conditions and their evidence requirements. It will evolve as agentic AI deployments mature and as practitioners operating inside the gap contribute to the methodology.

The Vordan Accountability Framework principle applies throughout: accountability must be built in before deployment, not retrofitted after failure. A system that requires a post-mortem to discover its accountability gaps was not built with accountability in mind.

VORDAN.CO/BASELINE AAB V0.2

VORDAN AGENTIC ACCOUNTABILITY BASELINE | AAB V0.1

Accountable by Design. Not by default.

VORDAN AGENTIC ACCOUNTABILITY BASELINE | AAB V0.1

SECTION 1

Scope and Applicability

The Baseline applies to any production deployment of one or more autonomous AI agents that take actions, access data, use tools, or transfer work to other agents on behalf of an organization, with or without human oversight at the action level.

The minimum conditions that trigger Baseline applicability are autonomous action, tool use, memory access, or handoff capability. A system that only suggests actions for a human to approve is a copilot. The Baseline does not govern copilots. A system that performs actions is an agent. The Baseline governs agents.

The Baseline applies regardless of the infrastructure vendor, the agent framework, or the underlying model. It is not contingent on organizational size, industry, or regulatory context. Any organization deploying agents that act on its behalf operates within the accountability conditions this document defines.

VORDAN.CO/BASELINE AAB V0.2

SECTION 2

The Seven Accountability Conditions

The Eight Accountability Conditions

Each condition translates an infrastructure question the harness answers into a governance requirement the organization must satisfy.

Conditions 2.1 through 2.7 govern how an agent acts. Condition 2.8 governs what the agent is. An agent that cannot verify its own identity has an accountability gap that precedes every other.

2.1

Authorization Provenance

THE HARNESS QUESTION

Is there a record of when and how this agent was deployed?

THE ACCOUNTABILITY CONDITION

Every action taken by an autonomous agent must be traceable to a human authorization event that preceded it. Deployment provenance is not authorization provenance. The existence of an agent in a production environment does not constitute organizational authorization for any specific action it takes. Authorization provenance requires a documented human decision, identifiable by role, timestamp, and scope, that explicitly sanctions the agent to act within defined boundaries on a defined task.

THE GAP

When an agent acts under standing permissions rather than task-specific authorization, the accountability chain begins with the agent, not with a human decision. That is not accountability by design. It is accountability abdicated at the point of deployment.

Authorization is not a configuration. It is a decision. The audit trail must show the decision, not just the configuration that followed from it.

2.2

Scope Integrity

THE HARNESS QUESTION

What permissions does this agent have?

THE ACCOUNTABILITY CONDITION

Agent permissions must be narrow, temporary, and revocable. Narrow means bounded to the specific task for which authorization was granted, with no ambient access to systems, data, or tools outside that scope. Temporary means permissions expire when the task ends, not when the agent is decommissioned. Revocable means a human with appropriate authority can terminate permissions immediately, without requiring a change to the agent's underlying configuration.

THE GAP

Persistent broad permissions that survive task completion are a scope integrity failure. An agent that retains access to systems it no longer needs for any authorized purpose is operating outside its accountability boundary, regardless of whether it acts on that access.

The accountability condition is not whether the agent used its excess permissions. It is whether excess permissions existed. The gap is structural, not behavioral.

2.3

Memory Governance

THE HARNESS QUESTION

What can this agent read from and write to memory?

THE ACCOUNTABILITY CONDITION

Every memory access, read or write, must be attributable to an authorized purpose, connected to an accountable owner, and auditable after the fact. It is not sufficient for the harness to log that a memory access occurred. The log must be interpretable: what was accessed, in the context of which task, under whose authorization, and for what stated purpose. Write operations require a higher evidence standard: what was written must be attributable to a human-authorized decision, not an autonomous inference.

THE GAP

Memory that accumulates across sessions without an owner, a purpose boundary, or an expiration condition is ungoverned memory. An agent with access to ungoverned memory is an agent operating partially outside the accountability architecture, regardless of how well the rest of the deployment is governed.

Memory is not a neutral resource. It is a data asset with governance obligations. The accountability condition applies to what the agent remembers, not just what it does.

2.4

Handoff Traceability

THE HARNESS QUESTION

Which agents did this agent transfer work to?

THE ACCOUNTABILITY CONDITION

Every transfer of work between agents must carry the authorization chain forward intact. A handoff that transfers a task without transferring the authorization context that sanctioned the task is an accountability gap, regardless of whether the technical handoff succeeded. The receiving agent must be able to answer, from the handoff record alone, who originally authorized the work, what scope that authorization covered, and whether the receiving agent's actions fall within that scope.

THE GAP

Authorization chains that terminate at the first handoff create accountability voids in every downstream agent action. The organizational liability for what downstream agents do does not terminate at the handoff. The accountability architecture must extend to wherever the work goes.

A handoff is not a transfer of liability. It is a transfer of task with continuity of accountability. The chain must survive the handoff or the handoff breaks the governance architecture.

2.5

Prompt Integrity

THE HARNESS QUESTION

What data entered this agent's context?

THE ACCOUNTABILITY CONDITION

What enters an agent's context must be controlled, auditable, and protected against unauthorized manipulation. Prompt injection, the introduction of instructions through data channels not intended to carry instructions, is an accountability failure before it is a security failure. It breaks the authorization chain by causing an agent to act under instructions that were never authorized by any human decision. Context contamination from unverified external sources carries the same accountability consequence: the agent acted, but not under any authorization the organization can identify or own.

THE GAP

An agent whose behavior can be altered by data it retrieves from external sources, other agents, or user-supplied content, without a control layer that validates that content against authorized instruction sources, is an agent whose actions cannot be fully attributed to organizational authorization. The accountability chain has an exploitable break at the context boundary.

Authorization governs instructions, not just permissions. An agent that can be instructed through its data inputs has an authorization boundary that the harness alone cannot secure. The accountability condition requires that the instruction surface be governed, not just logged.

2.6

Decision Auditability

THE HARNESS QUESTION

Is there a log of what actions the agent took?

THE ACCOUNTABILITY CONDITION

Actions above a defined sensitivity threshold must have an approval record connected to an accountable human decision point. The sensitivity threshold must be defined before deployment, not determined retroactively. An automated approval, a rule that triggers without human review, does not satisfy this condition for sensitive actions. The audit trail must show a human decision, identifiable by role and timestamp, that authorized the specific action or class of action before it was taken.

THE GAP

A log that records what an agent did is not an approval record. Recording an action after the fact proves the action occurred. It does not prove it was authorized. For sensitive actions, the accountability condition requires prospective authorization, not retrospective documentation.

The audit trail must answer two distinct questions: what did the agent do, and who decided it should. A trail that answers only the first question is an observation record. A trail that answers both is an accountability record.

2.7

Forensic Reconstructibility

THE HARNESS QUESTION

Do we have logs sufficient to investigate an incident?

THE ACCOUNTABILITY CONDITION

When something goes wrong, the organization must be able to reconstruct who did what, under whose authority, on what data, and why, from its own audit trail, without requiring access to vendor infrastructure, vendor personnel, or vendor-held logs. Forensic dependence on the vendor is an accountability gap. If the organization cannot independently reconstruct the sequence of events, it cannot independently establish accountability. It cannot answer to a regulator, a board, or an affected party without the vendor's cooperation, cooperation that may not be available, timely, or unconflicted.

THE GAP

Logs held exclusively by the infrastructure vendor, logs that require vendor tooling to interpret, and logs that are inaccessible without an active vendor relationship are not organizational audit trails. They are vendor-held records that the organization may be permitted to access under current contract terms. The accountability condition requires that the organization own the forensic record, not lease access to it.

Accountability cannot be outsourced. The organization that deployed the agent bears the accountability for what it did. That accountability requires the independent capacity to reconstruct the agent's actions. A governance architecture that depends on vendor cooperation to establish what happened is not accountable by design.

2.8

Model Substrate Integrity

THE HARNESS QUESTION

Is there a record of which model was deployed?

THE ACCOUNTABILITY CONDITION

An accountable agentic deployment must be able to verify, through a mechanism independent of the deploying party's assertion, that the model executing its instructions is the model it was authorized to deploy. Namespace, documentation, and community attestation do not satisfy this condition. Verification must be technically grounded: the deployed model must be capable of producing evidence of its identity that cannot be forged by a model with different parameters. Where no independent verification mechanism exists, the absence of verification is itself a condition finding. An organization that cannot verify the identity of the model its agents are running cannot satisfy Authorization Provenance for any action those agents take.

THE GAP

Model distribution infrastructure does not currently provide technical model identity verification. Weights retrieved from public repositories may have been substituted, modified, or backdoored between publication and download with no platform-level mechanism to detect the change. An agentic deployment that retrieves and executes weights without independent identity verification has delegated a foundational accountability decision to the distribution platform's social controls. Social controls are not a governance architecture.

Authorization Provenance requires knowing who authorized what. It cannot be satisfied if the identity of the model taking action is unverified. Model Substrate Integrity is the precondition for every other condition in this Baseline.

VORDAN AGENTIC ACCOUNTABILITY BASELINE | AAB V0.1

SECTION 3

Evidence Standards

Each evidence standard is organized into three tiers: what must exist, what must be demonstrable on demand, and what must be independently verifiable. Any technical implementation that produces the required evidence satisfies the standard.

3.1 Authorization Provenance

MUST EXIST

A human authorization record for each agent deployment, containing the authorizing role, the timestamp of authorization, the scope of authorized actions, the data sources the agent is permitted to access, and the conditions under which authorization expires or must be renewed.

MUST BE DEMONSTRABLE ON DEMAND

For any agent action in the audit trail, the organization must be able to produce the authorization record that preceded and sanctioned that action. The connection between action and authorization must be traceable without vendor assistance.

MUST BE INDEPENDENTLY VERIFIABLE

The authorization record must be stored in a system the organization controls, not exclusively in vendor infrastructure. It must be immutable after creation. Any modification to an authorization record must itself generate an auditable event with its own authorization provenance.

3.2 Scope Integrity

MUST EXIST

A permission manifest for each agent deployment documenting the specific systems, data sources, and tools the agent is permitted to access, the conditions under which each permission is active, and the expiration condition tied to task completion.

MUST BE DEMONSTRABLE ON DEMAND

For any point in time during an agent's operational period, the organization must be able to produce a complete picture of what permissions were active, what had expired, and what had been revoked.

MUST BE INDEPENDENTLY VERIFIABLE

Permission state changes must be logged as discrete auditable events. The log must show when permissions were granted, when they expired or were revoked, and whether any actions were taken during periods when permissions should not have been active.

3.3 Memory Governance

MUST EXIST

A memory access log that records every read and write operation performed by the agent, the task context in which each operation occurred, the authorization under which it was performed, and the owner responsible for the data accessed or created.

MUST BE DEMONSTRABLE ON DEMAND

For any memory access in the log, the organization must be able to identify the purpose that justified it and the human authorization that sanctioned the task it served. Write operations must additionally be traceable to the specific decision or output that generated them.

MUST BE INDEPENDENTLY VERIFIABLE

Memory that persists across sessions must have a documented owner, a documented purpose, and a documented expiration or review condition. Memory without an owner is ungoverned memory. Its existence in the audit trail is a gap finding, not a clean record.

3.4 Handoff Traceability

MUST EXIST

A handoff record for every transfer of work between agents, containing the originating agent identifier, the receiving agent identifier, the task being transferred, the authorization context being carried forward, and the timestamp of transfer.

MUST BE DEMONSTRABLE ON DEMAND

For any action taken by any agent in a multi-agent workflow, the organization must be able to trace that action back to the original human authorization event through an unbroken chain of handoff records. A chain with a missing link is an accountability void.

MUST BE INDEPENDENTLY VERIFIABLE

Handoff records must be stored in a way that neither the originating nor the receiving agent can modify after the fact. The integrity of the handoff record must be verifiable independently of the agents involved in the transfer.

3.5 Prompt Integrity

MUST EXIST

A context log for each agent session that records what entered the agent's context, the source of each input, whether each input was from an authorized instruction channel or a data channel, and whether any input was flagged or filtered by a prompt integrity control.

MUST BE DEMONSTRABLE ON DEMAND

For any agent action, the organization must be able to produce the full context that was present when the action was taken and identify whether any element of that context originated from an unauthorized or unverified source.

MUST BE INDEPENDENTLY VERIFIABLE

The context log must be produced at the time of the session, not reconstructed afterward. Retroactive reconstruction of what entered an agent's context does not satisfy this evidence standard. The log must be contemporaneous.

3.6 Decision Auditability

MUST EXIST

A sensitivity classification for agent action types, defined before deployment, that specifies which actions require prospective human authorization rather than automated execution. For each action above the sensitivity threshold, an approval record containing the approving role, the timestamp of approval, and the specific action or action class approved.

MUST BE DEMONSTRABLE ON DEMAND

For any sensitive action in the audit trail, the organization must be able to produce the approval record that preceded it. Approvals that postdate the actions they purport to authorize do not satisfy this evidence standard.

MUST BE INDEPENDENTLY VERIFIABLE

The sensitivity classification must be a governing document, versioned and dated, that predates the deployment it governs. Post-hoc sensitivity classifications applied to actions already taken do not constitute prospective authorization.

3.7 Forensic Reconstructibility

MUST EXIST

An organizational audit trail, stored in infrastructure the organization controls, that contains sufficient information to reconstruct the complete sequence of agent actions, the authorization state at each point in the sequence, the data accessed or modified, and the decision points where human authorization was required and obtained.

MUST BE DEMONSTRABLE ON DEMAND

The organization must be able to conduct a full forensic reconstruction of any agent session using only its own audit trail, without accessing vendor infrastructure, contacting vendor personnel, or invoking any vendor-provided forensic tooling. If the reconstruction requires vendor cooperation at any point, that dependency is a gap finding.

MUST BE INDEPENDENTLY VERIFIABLE

The audit trail must be tamper-evident. Any modification to the trail after the fact must itself generate an auditable event. Log completeness must be verifiable: gaps in the timeline are findings, not unknowns.

3.8 Model Substrate Integrity

MUST EXIST

A model identity record for each agentic deployment documenting the authorized model, version or checkpoint identifier, source from which weights were retrieved, date of retrieval, and the verification method applied to confirm the retrieved weights correspond to the authorized model.

MUST BE DEMONSTRABLE ON DEMAND

The organization must be able to demonstrate, for any point in the deployment's operational period, that the model executing instructions was the model authorized to do so. Where the underlying model has not changed, the model identity record serves as evidence. Where verification tooling exists, verification logs must be available. The organization must be able to distinguish between an unverified assertion and a verified fact regarding model identity.

MUST BE INDEPENDENTLY VERIFIABLE

Verification cannot rest solely on the distribution platform's namespace or documentation. Where a cryptographic or technically grounded verification mechanism is available and applicable to the deployed model, its use is required. Where no such mechanism currently exists for the model in question, the organization must document that fact explicitly as a known gap and assess the residual risk of unverified model identity against its accountability posture. Stating that no verification mechanism exists is not a gap finding. Failing to assess and document that absence is.

SECTION 4

Accountability Gap Classification

When an accountability condition is not met, the nature of the gap determines the remediation path. Vordan classifies accountability gaps into three types.

TYPE 1

Structural Gap

The accountability condition cannot be satisfied by the current architecture, regardless of process changes or ownership assignments. The limitation is in how the system was built. No policy document, no governance committee, and no additional logging will close a structural gap. The architecture must change. Structural gaps are the highest severity classification.

TYPE 2

Procedural Gap

The technical capability to satisfy the accountability condition is present but no process, ownership structure, or governance mechanism activates it. The architecture can support accountability. The organization has not built the governance layer that makes it function. Procedural gaps are closeable without architectural change through ownership

CRITICAL	Structural gap in Authorization Provenance, Handoff Traceability, Forensic Reconstructibility, or Model Substrate Integrity. The organization cannot establish accountability for agent actions in the affected domain under any circumstances.

Technical Gap

The architecture supports the accountability condition and a governance process has been designed to activate it, but a specific technical implementation failure prevents the condition from being fully satisfied. The intent is present. The execution has a defect. Technical gaps require a specific engineering fix.

GAP SEVERITY MATRIX

SEVERITY	CONDITION
CRITICAL	Structural gap in Authorization Provenance, Handoff Traceability, or Forensic Reconstructibility.

HIGH	Structural gap in any remaining condition, or procedural gap in Authorization Provenance, Handoff Traceability, or Forensic Reconstructibility.
MEDIUM	Procedural gap in any remaining condition, or technical gap in Authorization Provenance, Handoff Traceability, or Forensic Reconstructibility.
LOW	Technical gap in any remaining condition. The architecture is correct, the governance intent is correct, and the defect is specific and remediable.

SECTION 5

Relationship to Existing Frameworks

The Baseline is complementary to existing governance frameworks and derivative of none. The Baseline does not replace them. It extends into territory they have not covered.

NIST AI Risk Management Framework (AI RMF 1.0)

WHERE IT ALIGNS

The Govern and Map functions create space for the accountability architecture the Baseline requires. The AI RMF's emphasis on human oversight, transparency, and explainability maps directly to the Baseline's conditions on Decision Auditability and Forensic Reconstructibility.

WHERE THE BASELINE EXTENDS BEYOND IT

The AI RMF does not address agentic deployments specifically. Its accountability constructs were developed for AI systems that operate under continuous human oversight, not for autonomous agents that act, hand off work, and accumulate memory across sessions with minimal human intervention at the action level.

PRACTITIONER GUIDANCE

Treat the Baseline as an agentic extension of the Govern and Map functions. The seven accountability conditions provide the specific requirements that the AI RMF's principles imply but do not specify for autonomous agent deployments.

ISO/IEC 42001: Artificial Intelligence Management System

WHERE IT ALIGNS

ISO 42001 requires organizations to determine the context of their AI systems, establish roles and responsibilities, and maintain documented information sufficient to demonstrate conformance. These requirements create an organizational infrastructure that is necessary for, though not sufficient to satisfy, the Baseline's accountability conditions.

WHERE THE BASELINE EXTENDS BEYOND IT

An organization can achieve ISO 42001 conformance while deploying agentic AI systems that satisfy none of the Baseline's seven conditions. The distinction is between having a management system and having an accountability architecture.

PRACTITIONER GUIDANCE

Organizations that satisfy the Baseline's evidence standards for agentic deployments will have substantially completed the documented information requirements for those deployments under ISO 42001.

NIST Cybersecurity Framework 2.0 (CSF 2.0)

WHERE IT ALIGNS

The Govern function's emphasis on organizational context and risk management strategy provides a direct foundation for several Baseline conditions. Authorization Provenance and Scope Integrity map to Govern and Identify. Memory Governance and Prompt Integrity map to Protect and Detect. Handoff Traceability and Decision Auditability map to Govern and Respond. Forensic Reconstructibility maps to Recover.

WHERE THE BASELINE EXTENDS BEYOND IT

The CSF 2.0 does not address the specific accountability failures unique to autonomous agents: the authorization chain breaking at handoff, ungoverned memory accumulating across sessions, prompt injection breaking the instruction boundary, or forensic dependence on vendor infrastructure.

PRACTITIONER GUIDANCE

Organizations with mature CSF 2.0 implementations have the organizational infrastructure to implement the Baseline without building from scratch. Use the mapping above to align Baseline conditions with existing CSF controls.

ISACA COBIT 2019

WHERE IT ALIGNS

COBIT's governance domain addresses accountability assignment, stakeholder needs, and governance system design in ways that directly support the Baseline's requirements. Its management objectives around risk management, audit, and compliance provide organizational structures that the Baseline's evidence standards assume are present.

WHERE THE BASELINE EXTENDS BEYOND IT

COBIT was designed for IT governance broadly and predates the emergence of agentic AI as a distinct governance challenge. The framework provides the organizational scaffolding for accountability but does not specify what accountability requires in the agentic context.

PRACTITIONER GUIDANCE

Treat the Baseline as a domain-specific governance standard that operates within the COBIT governance architecture. The Baseline's accountability conditions represent the specific governance objectives that COBIT's framework structures should be directed toward for agentic AI deployments.

Every framework referenced in this section was developed before autonomous AI agents became a production reality in enterprise environments. Existing frameworks provide the organizational context in which the Baseline operates. They do not provide the accountability conditions the Baseline defines. Both are necessary. Neither is sufficient without the other.

SECTION 6

The Agentic Accountability Baseline is a living standard. Version 0.2 adds Condition 2.8 (Model Substrate Integrity) in response to documented accountability failures in AI model distribution infrastructure. It will continue to evolve as agentic AI deployments mature and as practitioners operating inside the gap contribute to the methodology.

6.0 Version History

Version 0.1, May 2026 Initial publication. Seven foundational accountability conditions, evidence standards, gap classification taxonomy, and framework alignment mappings.

Version 0.2, June 2026 Added Condition 2.8: Model Substrate Integrity. Preceded by Gap Alert Seventeen (June 5, 2026). Evidence Standard 3.8, Checklist item 2.8, and Glossary entry added. Severity table updated. Section 2 heading updated to Eight Conditions.

version increments reflect refinements to existing conditions, clarifications to evidence standards, or updates to framework alignment mappings. Major version increments reflect material changes to the accountability conditions themselves and will be preceded by a public comment period of no less than sixty days. Version 1.0 will be published when the methodology has been validated through field assessments.

6.2 Comment and Contribution Process

Practitioners, researchers, and organizations may submit observations, challenges, and proposed refinements through the Vordan editorial channel at hello@vordan.co with the subject line Baseline Feedback. Vordan does not accept feedback from infrastructure vendors regarding the conditions that govern the infrastructure they sell. The independence of the standard from vendor interest is a structural property, not a policy preference.

6.3 Assessment Record Governance

Organizations that conduct Baseline assessments should maintain their assessment records as governed documents subject to the same retention and integrity requirements as other audit artifacts. Assessment records are governance artifacts. Their absence, in an environment where the standard is publicly available and the assessment methodology is documented, is itself a posture finding.

6.4 Relationship to the Gap Score Instrument

The Gap Score is Vordan's assessment instrument for measuring an organization's accountability posture against the The Gap Score agentic module maps directly to the eight conditions in Section 2 and the standards in Section 3. The Gap Score instrument will be published separately as a companion document to this Baseline.

6.5 Publication and Distribution

The Baseline is published by Vordan at vordan.co and freely available for reference, citation, and organizational use. Organizations may incorporate the Baseline's conditions and evidence standards into their internal governance documentation provided they cite Vordan as the source and reference the specific version. Citation format: Vordan Agentic Accountability Baseline, Version 0.2, June 2026. vordan.co/baseline

VORDAN AGENTIC ACCOUNTABILITY BASELINE | AAB V0.1

APPENDIX A

Glossary

AAB	The shorthand reference for the Agentic Accountability Baseline. Used in practitioner citations as AAB v0.2, or with publishing body as Vordan AAB v0.2. The canonical document is available at vordan.co/baseline.
Accountability Condition	One of the seven requirements defined in Section 2 that must be satisfied for an agentic AI deployment to be considered accountable by design. Each condition defines a governance requirement, not a technical specification.
Agentic AI Deployment	Any production use of one or more autonomous AI agents that take actions, access data, use tools, or transfer work to other agents on behalf of an organization, with or without human oversight at the action level.
Authorization Chain	The unbroken sequence of human authorization events that sanctions every action taken
Model Substrate Integrity	The property of an agentic deployment in which the model executing instructions has been verified, through a technically grounded mechanism independent of the deploying party's assertion, to be the model authorized for that deployment. Its absence is a Critical gap finding where a verification mechanism exists and has not been applied. Condition 2.8 of the Baseline.
Forensic Dependence	The condition in which an organization cannot reconstruct the actions of an autonomous agent without access to vendor infrastructure, vendor personnel, or vendor-held logs. Forensic dependence is a gap finding under condition 2.7.
Gap Finding	A determination that a specific accountability condition is not satisfied by a specific agentic deployment, accompanied by a gap type classification and a severity rating.
Governing Document	A versioned, dated policy or standard that predates the deployment it governs and defines the accountability requirements that deployment must satisfy.

Harness	The infrastructure layer that captures what an autonomous agent does: identity systems, runtime policy enforcement, memory governance tooling, activity logging, and audit trail generation. The harness is necessary for accountability. It is not sufficient to constitute accountability.
Prompt Integrity	The property of an agentic deployment in which what enters the agent's context is controlled, auditable, and protected against unauthorized manipulation. Prompt integrity is violated when an agent can be caused to act under instructions that were not authorized by any human decision.
Sensitivity Threshold	The organization-defined classification of agent action types that require prospective human authorization rather than automated execution. The sensitivity threshold must be defined as a governing document before deployment.
Ungoverned Memory	Memory accessible to an autonomous agent that has no documented owner, no documented purpose, and no documented expiration or review condition. Ungoverned memory is a gap finding under condition 2.3.

APPENDIX B

Baseline Assessment Checklist

A condensed reference for practitioners conducting an initial accountability posture review. A negative response to any item indicates a gap finding requiring classification and remediation per Section 4.

2.1

Authorization Provenance

Is there a human authorization record for each agent deployment containing the authorizing role, timestamp, scope, permitted data sources, and expiration condition? Can any agent action in the audit trail be connected to that authorization record without vendor assistance? Is the authorization record stored in organizational infrastructure and immutable after creation?

2.2

Scope Integrity

Does a permission manifest exist for each deployment documenting specific systems, data, and tools the agent may access, the conditions under which each permission is active, and the expiration condition tied to task completion? Can the organization demonstrate that no permissions persisted beyond their authorized scope?

2.3

Memory Governance

Does a memory access log exist recording every read and write operation, the task context, the authorization under which it occurred, and the data owner? Has every persistent memory object been assigned an owner, a purpose, and an expiration or review condition? Is there ungoverned memory in the environment?

2.4

Handoff Traceability

Does a handoff record exist for every transfer of work between agents containing the originating agent, the receiving agent, the task transferred, the authorization context carried forward, and the timestamp? Can any downstream agent action be traced back to the original human authorization event through an unbroken chain?

2.5

Prompt Integrity

Does a context log exist for each agent session recording what entered the agent's context, the source of each input, and whether any input was flagged or filtered? Can the organization demonstrate, for any agent action, whether an unverified input was present and what control evaluated it?

2.6

Decision Auditability

Does a sensitivity classification governing document exist that predates the deployment? For every sensitive action in the audit trail, does a prospective approval record exist identifying the approving role and timestamp? Are there automated approvals substituting for human decision points on sensitive actions?

2.7

Forensic Reconstructibility

Is the organizational audit trail stored in infrastructure the organization controls? Can the organization conduct a full forensic reconstruction of any agent session without vendor cooperation? Is the audit trail tamper-evident and verifiable for completeness? Are there gaps in the timeline?

CITATION

Vordan Agentic Accountability Baseline, Version 0.1, May 2026. vordan.co/baseline

Published by Vordan. Feedback: hello@vordan.co (subject: Baseline Feedback). Assessment inquiries: hello@vordan.co

2.8 Model Substrate Integrity

Does a model identity record exist for this deployment documenting the authorized model, version identifier, retrieval source, retrieval date, and verification method? Can the organization demonstrate that the model currently executing instructions is the model that was authorized? Has a technically grounded verification mechanism been applied where one exists? If no mechanism exists, has the organization documented that absence and assessed the residual risk?

CITATION

Vordan Agentic Accountability Baseline, Version 0.2, June 2026. vordan.co/baseline
Published by Vordan. Feedback: hello@vordan.co (subject: Baseline Feedback).
Assessment inquiries: hello@vordan.co