GOVERNED AUTONOMY

AD8: Accountable change across software, action, and learning

A paper in the Governed Autonomy collection

Andrew Durden
Independent research
Publication edition 2026.10.6

Research on authority, evidence, and continuity in autonomous systems.

Content identifier: 058a680e5b3eee8e57a28486

All rights reserved · Research exposition

Contents

AD8: Accountable change across software, action, and learning **3**

Abstract . . . 3

1. The problem of connected responsibility . . . 3

2. Identity, meaning, and authority . . . 4

3. From intended behavior to an operating system . . . 5

4. Concurrent work and the validity of premises . . . 5

5. Effects, observation, and uncertain outcomes . . . 6

6. Generated agreement and bounded assurance . . . 7

7. Learning, model release, and continuing eligibility . . . 7

8. Four illustrative cases . . . 8

9. Evidence boundary and open questions . . . 9

Source guide . . . 12

All rights reserved · Research exposition

AD8: Accountable change across software, action, and learning

Governed Autonomy · Publication 2026.10.6
Research study: conceptual exposition of the AD8 specification
Source: NTVBCD/AD8 2.0.0, 19 July 2026; Master, Annexes A–I, and reference scope statement

Abstract

An autonomous system can change the software it runs, take actions through that software, and use the resulting experience to change its future behavior. Each activity has a different evidential and authorization burden. Combining them without preserving those differences allows a successful build to stand in for a successful deployment, an apparently useful action to stand in for an authorized action, or accumulated usage to stand in for permission to train. AD8 investigates a common account of identity, authority, evidence, effects, and continuity across these activities. Its contribution is an architectural proposal for keeping their responsibilities connected without allowing one responsibility to confer the powers of another.

This study examines intended meaning and observed behavior, the continuing validity of decisions, agreement among generated artifacts, and the authorities involved in learning and model release. Four illustrative cases connect these questions to ordinary work and recovery. The available evidence includes reported reference checks and bounded formal results. Complete conformance remains unestablished, and a construction affecting a foundational evidence relationship remains unresolved. The paper identifies what further evaluation would need to demonstrate.

1. The problem of connected responsibility

Automation is often organized around individual transactions: accept a request, produce a result, and report success. Long-lived autonomous work is harder to describe this way. A request can cause a new artifact to be generated; that artifact can depend on other software; the software can act in an external system; the resulting observations can influence later plans or become proposed training material. Responsibility follows this chain even when the organizations, technologies, and times involved change.

A plausible result at one point does not settle the questions at the next. A generated document might satisfy a formatting test while containing an unsupported factual claim. A tested program might be unsuitable for the environment in which it is deployed. A payment request might be authorized but have an uncertain outcome after a connection loss. A useful transcript might contain material whose owner permitted operational use but prohibited shared learning. These are different conditions, requiring different evidence and different authority.

All rights reserved · Research exposition

AD8's master specification proposes that those distinctions remain part of the system's enduring account of its work. The central question is not simply whether a task can be completed. It is whether a particular change or action can be justified by the intended meaning, current authority, applicable evidence, and known external conditions that bear on it. Source guide: scope.

The proposal spans software realization, concurrent work, external effects, stochastic generation, learning, model release, and recovery. Evaluating it requires both evidence for those responsibilities and evidence that their boundaries survive composition. The architectural overview compares AD8 with related works, each of which retains its own scope and requirements.

2. Identity, meaning, and authority

The first conceptual distinction concerns what a thing is, which version is being discussed, what information supports it, and what it is presently allowed to do. These questions are related, but a single label cannot safely answer them all.

A continuing business object can change while remaining the same object for organizational purposes. A particular version can remain available as historical evidence after it is no longer eligible for current use. A signature can establish a relationship to a signing identity without establishing that the signer had the relevant power, that the intended audience matches, or that the permission remains valid. AD8 consequently treats identity and current authority as separate subjects of reasoning. Source guide: identity.

An intention is also distinct from evidence about the world. An organization may intend to restrict an action to a class of recipients; observations may show that a candidate action includes someone outside that class. The observation does not silently amend the intention. Conversely, the existence of an intention does not establish that the world already conforms to it. This separation enables disagreement and uncertainty to remain visible instead of being resolved by whichever representation is most convenient to the next component.

Authority is contextual. It concerns an actor, a purpose, an affected resource or party, and relevant conditions. Delegation is meaningful only within those limits. Passing a request through another component should not enlarge it, and a technical ability to perform an operation is not sufficient justification for doing so. Recovery likewise cannot create a permission merely because the ordinary permission service is unavailable.

The practical consequence is that historical validity and current usability need separate answers. A reviewer may legitimately inspect a withdrawn decision in order to understand prior events. That historical access does not make the decision suitable for a new action. This distinction is particularly important when a task lasts longer than the permissions, environmental assumptions, or evidence on which its initial plan depended.

All rights reserved · Research exposition

3. From intended behavior to an operating system

AD8 treats software realization as an accountable interpretation of intended behavior. The intended result includes more than the most visible feature. It can include state ownership, access restrictions, data preservation, recovery, accessibility, operational behavior, and the evidence needed to determine whether the result is acceptable. A plan that accounts for the happy path while omitting these obligations is incomplete in the proposal's terms. Source guide: realization.

This conception of completeness does not imply that a compiler can discover every relevant fact or settle every human judgment. Some obligations depend on research, legal interpretation, or a decision among materially different alternatives. Those dependencies must remain identifiable. A favorable score on convenient criteria cannot compensate for a prohibited action or an unresolved requirement that is essential to the intended result.

The proposal also separates the artifact from its operation. Preserving the bytes of a generated artifact establishes which artifact is under discussion. It does not establish its semantics, suitability, or successful activation. Testing contributes evidence within its stated scope. Deployment introduces additional questions about the environment, state transitions, compatibility, and observed behavior. A partially activated system cannot be described as fully operating simply because some of its components are healthy.

The same distinction governs repair. A system may be able to reconstruct a resource that it owns and whose intended state is established. It cannot infer that every absent external effect should be recreated. An absent local record of a message does not establish that the message was never sent. Nor can repair overwrite a person's work merely because that work differs from a desired configuration.

Experience is part of this account. If a promised behavior requires a person to perceive a status, use a control, recover focus, or understand a necessary approval, then producing backend machinery alone does not satisfy the promise. AD8 includes experiential obligations while retaining a separate authority boundary: an interface can communicate an authorization decision, but its appearance cannot establish the underlying permission.

4. Concurrent work and the validity of premises

An autonomous worker reasons from a view of the world. Other workers may change that world while the reasoning continues. Correctness therefore depends on more than whether a view was once accurate. It also depends on whether the worker consulted the information necessary for its decision and whether relevant changes have occurred since then. Source guide: concurrent work.

Consider a task that prepares a change for a group of accounts. Reading every account already known to the worker does not necessarily establish that the group is complete. A newly eligible account or a change in the rule defining the group can matter just as much as an edit to a known account. Adequacy of the premises and freshness of those premises are distinct questions.

AD8's proposal gives both questions architectural standing. The conceptual requirement is that a consequential change remain related to a sufficiently complete and sufficiently

All rights reserved · Research exposition

current basis. When that relationship no longer holds, the system must reconsider affected work. Merely recording that a worker read something earlier cannot justify a later decision under changed conditions.

This does not require every concurrent change to invalidate all work. Some changes are independent under a declared interpretation. The research difficulty lies in establishing that independence rather than assuming it from superficial separation. Two edits in different places can still conflict through a common business rule; two edits to the same object may concern independent properties. Any stronger claim about compatible concurrency depends on the semantics and evidence available for that domain.

The proposal thus treats collaboration as a problem of preserving justified decisions over time. Scheduling mechanisms may reduce wasted work, but scheduling priority does not create authority or settle a semantic conflict. The CSSL study develops the related distinction between changing stored content and changing justified semantic state.

5. Effects, observation, and uncertain outcomes

An effect is a consequence that matters beyond an isolated computation. It can involve data, communication, money, physical activity, or a change in another service. AD8 argues that effects should be considered both before execution and through observation afterward. Intended scope, authorized scope, and observed consequences answer different questions. Source guide: effects.

Different powers cannot always be reduced to one scale. Permission to read a file does not imply permission to send its contents elsewhere. Permission to spend within a budget does not imply permission to contact any recipient. A system that reports only a broad confidence or risk score can conceal those distinctions. AD8 instead investigates a structured account in which the relevant dimensions remain separately meaningful.

Attribution is equally important. An action can pass through several components before reaching an external service. Its consequences do not lose relevance because a supporting component produced them. Operational measurements may themselves become significant if they influence later decisions, expose information, or alter consequential state. The classification must follow actual use and consequence, rather than the component's convenient description of itself.

Observation also has limits. If a connection ends after an external request may have been dispatched, the system cannot conclude either that the action succeeded or that nothing happened. Repeating it may duplicate a payment, message, reservation, or physical instruction. Recovery must establish what happened and which further actions remain authorized. A compensating action creates a new consequence; it does not make the original event disappear.

This is a demanding form of continuity. The objective remains relevant after a failed attempt, but continuing requires a valid account of current conditions. The system may resume, refresh evidence, choose another eligible route, or reconcile an external result. It must not confuse persistence with permission to repeat an action whose outcome is unknown.

All rights reserved · Research exposition

6. Generated agreement and bounded assurance

Stochastic systems can produce several plausible artifacts for the same intended task. Their agreement is informative only in relation to what was held constant, what was varied, how results were compared, and which failures remained visible. AD8's certification proposal is motivated by the risk of converting repeated plausibility into an unwarranted claim of correctness. Source guide: generation.

Two generated results can appear equivalent under the observations available to an evaluator while differing in an untested condition. Evidence that has not distinguished them is weaker than a proof that they are equivalent within a defined model. A newly discovered distinguishing case can therefore matter more than a large collection of earlier similarities. The public claim must remain bounded by the interpretation and instruments that supported it.

Agreement also differs from compliance. Many candidates may converge on the same prohibited behavior. Conversely, several distinct realizations may all meet the requirements. Combining agreement and compliance into an undifferentiated success measure would obscure both possibilities. The proposal keeps their evidential roles distinct and requires variation across conditions to remain visible rather than disappearing into an average.

This is not a claim that deterministic checking eliminates uncertainty. A checker has a scope, assumptions, and an implementation of its own. The choice of observations may miss a relevant consequence; a test population may inadequately represent future use. The research question is how to state the resulting guarantee precisely enough that another party can tell where it applies and where further evidence is needed.

The comparison with SEPION concerns assurance for generated artifacts. It identifies related evaluation questions; transferring a guarantee between realizations would require an explicit argument about the relevant models and assumptions.

7. Learning, model release, and continuing eligibility

Operational experience can support improvement, but use of a system is not itself permission to contribute to shared learning. AD8 distinguishes ordinary use, private memory, organization-specific adaptation, and shared model development. Each raises different questions of purpose, ownership, privacy, and future influence. Source guide: learning.

The proposal treats an experience as more than its successful ending. Failed branches, corrective actions, uncertainty, dissent, and missing observations can all change what may reasonably be learned from it. Removing those elements can turn a conditional success into an apparently general rule. An account of experience should preserve the limits of its causal interpretation, including whether the claimed outcome was independently observed.

Even eligible material need not belong in shared model parameters. Some information changes frequently, applies only to a particular organization, remains private, or requires prompt withdrawal. Its appropriate use may remain external to the model. More persistent learning raises a stronger burden because the resulting influence can be harder to isolate

All rights reserved · Research exposition

and remove. This is a question of the proposed destination and future use, rather than an assumption that more collected material is always better.

AD8 further distinguishes producing a model candidate, evaluating it, making it eligible for release, deploying it, and selecting it for a particular request. A candidate's capability gain does not settle privacy, authority, security, or lineage concerns. The ability to train a model cannot confer the right to approve its own use. Nor does availability of a model imply that it is suitable for every organizational or jurisdictional context.

Continuity of model assets introduces another separation. Preserving or reconstructing model bytes can be necessary for recovery, but possession does not itself authorize deployment or behavior. Similarly, withdrawing an eligible contribution does not prove that every past influence has been erased from a trained model. Future-use restrictions, replacement, validated remediation, and residual influence require distinct statements. Claims of complete removal must be supported by the evidence actually available.

8. Four illustrative cases

Four hypothetical cases illustrate the responsibilities AD8 seeks to connect. Their outcomes are examples for analysis rather than recorded executions.

8.1 An authorized service change

An organization requests that customers receive a clearer appointment confirmation. The intended outcome includes the message's meaning, permitted recipients, accessibility, and preservation of existing appointments. A proposed realization accounts for these obligations, and review evidence addresses the behavior in the intended environment. The organization authorizes the change within a defined scope.

Success in this illustration requires the relevant service to operate as intended and the result to be observed. A stored artifact or an accepted deployment request is an intermediate fact. If only some routes use the new behavior, the account remains partial. The example demonstrates why semantic intent, artifact identity, operating state, and observed experience need connected but distinct descriptions.

8.2 A useful proposal outside its authority

A worker preparing the confirmation improvement proposes adding a promotional message to every customer record. The suggestion might be commercially attractive, and its implementation might pass technical checks. Neither fact establishes permission to use the broader recipient group or to change the message's purpose.

The expansion is denied within the existing authority. Work on the authorized confirmation improvement can continue. If the organization later chooses the broader change, that choice needs its own adequate basis and authorization. The denial concerns the particular proposed expansion; it does not redefine the original objective as failed or require abandoning work that remains achievable within scope.

All rights reserved · Research exposition

8.3 A payment response is lost

An authorized payment is submitted, but the connection ends before its result is observed. The worker retains the objective of settling the invoice. It does not infer from the missing response that the payment never occurred.

An authorized inquiry establishes that the external service completed the payment. The local account is reconciled to that observed result, and the invoice can be reported as settled. If the inquiry had instead established that no payment occurred, a permitted continuation could be evaluated. If the result remained unresolved, the account would retain that uncertainty. The example illustrates recovery through knowledge of the external state rather than through blind repetition.

8.4 A learning contribution is withdrawn

A participant explicitly allows a bounded body of experience to be considered for a learning purpose. Later, the relevant permission is withdrawn. The system must determine which proposed or active uses depend on that contribution and prevent uses that no longer satisfy the applicable conditions.

The claim that can be made depends on the affected destination. Removing an external reference, retiring a candidate, and addressing influence in an already trained model are different accomplishments. The system records what it has actually restricted or remediated and which residual questions remain. Recovery aims to restore an eligible service or model route without silently preserving a use whose justification has been withdrawn.

9. Evidence boundary and open questions

AD8 2.0.0 defines a proposed architecture and its conformance requirements. Its disclosed evidence includes reference encoding checks, generated-representation checks, and three bounded formal models executed in TLC. Authored traces remain illustrations. The results below establish their stated finite propositions and are attributed to the recorded source runs. Source guide: evidence.

9.1 The three finite model results

The model scope guide and the recorded 22 September 2026 verification identify TLC 2.19, the selected model and configuration revisions, and four deliberately faulty controls. The table reports those records, rather than inferring a result from the presence of formal source. Every successful base model has four distinct reachable states and complete search depth three. Their small size is part of the scope, not evidence of a complete AD8 implementation.

All rights reserved · Research exposition

Model and proposition	Domain and premises	Reported checking result and control	Claim supported
Approval payload binding	One approval, two possible presented payload identities, and one consumption; execution must consume the approved payload	Base model: no error, 10 generated states and four distinct states. Removing the payload-equality guard produces an invariant counterexample.	The executed payload equals the approved payload in this model. A later presented input cannot redefine what was consumed. This does not establish the real signing, approval, or execution path.
Receipt coverage and freshness	Three Boolean properties represent coverage, freshness, and present eligibility; invalidation must remove eligibility	Base model: no error, 10 generated states and four distinct states. Retaining eligibility through invalidation produces a coverage/freshness counterexample.	Current eligibility implies covered, fresh supporting reads in this model. Invalidation affects present reliance; it does not erase historical publication. Completeness of real observations is assumed rather than proved.
One-time initialization and progress	Four abstract phases and one initialization exception; ordinary authority follows initialization. Eventual resolution assumes weak fairness of preparation and finalization-or-abort. Terminal phases permit stuttering.	Base model: no invariant or temporal error, six generated states and four distinct states. Retaining the exception produces an invariant counterexample. Removing fairness produces a temporal counterexample.	The exception cannot remain usable after finalization; initialization precedes ordinary authority; eventual resolution holds under the declared fairness assumption. Real people, services, custody, and recovery are not shown to satisfy that assumption.

The successful base runs are identified by the following SHA-256 fingerprints. These identify the actual model and configuration bytes used in the recorded evaluation; the deliberately modified controls have their own model fingerprints. The checker archive is tla2tools 1.7.4, SHA-256 `936a262061c914694dfd669a543be24573c45d5aa0ff20a8b96b23d01e050e88`. The source result record has SHA-256 `f707399e88ac25f90ff8b53c3abfb689033629f1c28cc983547d11301e3e857f`. Fingerprints establish artifact identity, while the table above states the limited propositions supported by the results.

Model	SHA-256 of the selected inputs
AD8Approval	Model: `7cf8c5b32445429298828af1e3e31d9071e8fee314c61eba679f1420b1a93c06`; configuration: `61e4defcac549bcf45a633fcf3000a2531ce1cdaab852153724f0290434e1446`
AD8ReceiptFreshness	Model: `afeb3493a4bd6ec4e558c5f51b44e74f55c4809569ebb1e59b033fe07ed00a9e`; configuration: `6b5f06a7fe6b34557e4d21dfeff4f58d60e74d8f0cebbb7aa7e6ddd72269a746`
AD8Genesis	Model: `0db5bca9363cac399f863c1047adb5853c2896240e55c75501557cccd3aa80b9`; configuration: `52a624c6fd413f0bcadc5d115f48d4ce22986d1a9a8fd99dba368030574ef515`

All rights reserved · Research exposition

The deliberate faults are successful controls because the checker produces the relevant counterexamples. A parsing failure or timeout would not supply that evidence. The remaining TLA+, Alloy, Lean, and SMT artifacts are candidate studies without qualification for the broad claims previously attached to them. Neither these three results nor the source's aggregate status counts establish its 215 system-level assurance claims. Concrete behavior still requires a faithful implementation, a stated fault model, and observations at the real decision and effect boundary.

9.2 Foundational dependencies and their consequences

Two disclosed dependencies restrict the transition from reference arithmetic to object conformance.

Foundation	Missing step and why it matters	Dependent claims	What remains supportable
Reciprocal binding between a candidate and its required evidence	The proposed candidate identity depends on a commitment to required evidence, while that evidence identifies the same candidate. The source has not supplied an acyclic construction for a nonempty evidence set. Computing each commitment correctly does not determine how to construct both mutually dependent identities.	Conformance of nonempty evidence-bearing candidates and the authorization or transitions relying on their binding	Bounded encoding and evidence-tree arithmetic; the requirement to bind evidence to the exact candidate. Neither establishes a constructible complete signed object.
Agreement between primitive meanings and generated carriers	The source records a 128-bit decimal description paired with 64-bit carriers and differences in some domain and identifier ranges. A generated representation has not thereby preserved the complete declared value domain.	Full primitive closure and semantic equivalence of every generated representation	Checks over their explicitly supported values and representations. A result for that subset cannot qualify values outside it.

For the first dependency, consider an abstract artifact A whose identity must bind its required review evidence E. If E must already name A's final identity, neither final commitment follows simply from having the other's placeholder. Leaving out the required evidence would change the requirement. For the second, a value accepted in a wider declared range can be unrepresentable in a narrower carrier; successful small-value round trips cannot settle that case. These examples explain construction and compatibility gaps, not newly observed product incidents.

The source also distinguishes historical integrity fixtures from current conformance. Older signature fixtures bind opaque descriptive bytes, and a retained evidence-tree fixture uses legacy text identifiers. Their integrity or arithmetic can still be checked within that historical scope. They do not qualify the current semantic producer, current authority, or typed-object binding. Resolution requires a defined construction and agreement among its producers and

All rights reserved · Research exposition

verifiers, followed by checks of the affected objects. The three finite models above do not supply that missing construction.

Several research questions follow. How can an observation model be shown to cover the consequences that matter in a new domain? Which concurrent changes can be proved irrelevant to an ongoing decision? How can independent evaluation avoid common sources of error or institutional capture? What evidence justifies a claim about residual influence after learning permissions change? What recovery guarantees remain achievable when the external service or authority required for reconciliation disappears?

There is also an integration question: the more responsibilities a common architecture connects, the greater the need to show that their boundaries survive composition. Local checks can all pass while an end-to-end path silently bypasses an intended owner. Demonstrating the intended separation therefore requires observed system behavior across success, denial, interruption, and recovery, together with explicit assumptions about the people and services outside the system.

AD8 contributes an account of responsibility across software change, action, learning, and time. Its evaluation would test whether those connections make continuing work more accountable without losing useful progress. The evidence guide, institutional thesis, and glossary provide the wider context for that investigation.

Source guide

All references below identify sections of AD8 2.0.0. The paper develops their conceptual argument while excluding proprietary construction and acceptance procedures.

Scope: *NTVBCD/AD8 Master*, §§0–2, “Architecture decision record” (§9), and migration and supersession scope (§14).

Identity and authority: *Annex A — Canonical Object, Identity, Time, and Authority Algebra*, A.1–A.15; *Master*, §§3–5.

Software realization: *Annex D — Intent Compiler, IR, Projection, Materialization, and Activation*, D.1–D.20; *Annex I — Implementation Topology, Migration, Retirement, and Acceptance*, I.1–I.16.

Concurrent work: *Annex B — Semantic Ledger, Concurrency, Agent ABI, and Replay*, B.1–B.17.

All rights reserved · Research exposition

Effects and recovery: *Annex C — Effect Lattice, Containment, Attribution, and Receipts*, C.1–C.18; *Annex H — Security, Threat, Failure, Recovery, and Formal Verification*, H.4, H.8, and H.11.

Generated artifacts: *Annex E — Stochastic Artifact Certification*, E.1–E.17.

Learning and models: *Annex F — Native Intelligence Accretion, Learning, Evaluation, Release, and Custody*, F.1–F.22.

Evidence limits: *Master*, §§0, 11–13; *Reference Implementation Scope*; *TLA Model Scope* and its recorded evaluation of 22 September 2026; *Annex A*, A.2 and A.5; *Annex G — Schemas, Protocols, Algorithms, and State Machines*, G.0 and G.12; *Annex H*, H.12–H.15; *Annex I*, I.12 and I.14–I.18. The model, configuration, checker, and result-record fingerprints are given in §9.1.

All rights reserved · Research exposition