GOVERNED AUTONOMY

Engineering Methodology

A paper in the Governed Autonomy collection

Andrew Durden
Independent research
Publication edition 2026.10.9

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

Content identifier: 7150dfae1470e29ab5d0fbcc

All rights reserved · Research exposition

Contents

Engineering Methodology **3**

Responsibility and the subject of evidence . . . 3

A test can change the claim . . . 4

Working with substantive AI proposals . . . 4

Integration, operations and continuing responsibility . . . 5

Credit and novelty . . . 5

Acceptance and review . . . 5

All rights reserved · Research exposition

Engineering Methodology

I am the sole human engineer and architect behind Cuttlefish and Freehold. My responsibility includes technical conception, architecture, implementation, integration, production infrastructure and operation. For Cuttlefish, that includes the enterprise backend, GitOps, production deployments, Azure and AWS engineering and orchestration, and the Azure-to-AWS migration. This is my account of the work I am responsible for. It does not mean that every mechanism discussed in the research is deployed.

I worked with AI systems throughout exploration, research, design, implementation, debugging, testing, critique, validation and engineering execution. Their contribution was substantial. Some architectural concepts began with me; other discoveries and refinements emerged through that collaboration. I directed the work and retained final technical authority, integration decisions, engineering acceptance and production responsibility. AI contributed well beyond document preparation. I remained responsible for accepting and operating the result.

My Author biography describes my background and how I came to this work. Here I explain how to evaluate the engineering evidence and what the available record establishes. Preparing this publication is a separate activity from building the systems.

Responsibility and the subject of evidence

A useful architecture assigns an owner to consequential state, a consumer to each authorization decision and an observer to each claimed outcome. That assignment is initially a specification. To make an engineering claim, one must identify what executable component actually performs the responsibility, where a failure can occur and how the result was observed.

Three records therefore remain separate. Research records explain a proposition, assumptions and counterexample. Reference records identify an executable model, inputs and observed behavior. Production records identify a dated operational observation and its provenance. The records can support one another, but they are not substitutes. A compiled carrier does not validate all semantic values. A finite checker does not establish the behavior of every product route. A ready workload does not show that a particular customer objective was fulfilled.

The dated implementation report retains the strongest provenance I have published. I directed that examination on 10 October 2026 with AI assistance, combining authenticated read-only metadata with retained workload observations and source inspection. That examination did not execute a customer action to prove fulfillment. The AD8 research pass added in edition 2026.10.7 had no access to product repositories or live infrastructure, so it adds no fresh production observation or independent recertification of that examination. I attest that both products operate end to end for real customers. That statement does not

All rights reserved · Research exposition

supply payment evidence, customer identities or independent validation of individual feature deployments.

A test can change the claim

The research record contains corrections, failures and withdrawals as well as successful checks. In the recorded September 2026 reference work, boundary tests challenged text normalization, nested structures, container counts and total encoded size. Later passing tests support the corrected bounded codec behavior. They do not establish complete object authorization or a full runtime. The AD8 investigation keeps those limits explicit.

A particularly instructive failure mode is a test that obtains its supposed observation from the fixture's expected answer. Such a runner can produce a polished success report without executing the behavior under examination. The appropriate engineering response is to withdraw the assurance, retain the scenario as an authored outline and build an independent observation path before claiming execution. A large inventory of scenarios cannot repair that missing path.

Another example is a valid final state reached through an invalid intermediate observation. The atomic-observation dossier deliberately publishes a partial two-field update. The resulting total is wrong at the first observation and correct at the last. A final-state-only check would pass. This original public example makes the testing lesson inspectable without exposing the product's implementation.

These examples show why a failing case can expose a claim or implementation that needs to change. A failed test should prompt a question about which claim, assumption or implementation must change. I do not present a reconstructed conversation history about how particular AI proposals were debated. The available evidence supports the corrections and test outcomes; it does not establish every private decision that preceded them.

Working with substantive AI proposals

AI-generated designs and code need the same acceptance boundaries as other candidate work. A proposal can be coherent and still omit an important observation route, broaden a value language, misclassify an uncertain effect or assume progress that an environment does not provide. Agreement among several outputs does not supply the missing experiment.

The research illustrates a disciplined evaluation pattern: state the obligation, expose the assumptions, construct a case that should succeed, construct a case that should fail, inspect the actual implementation and retain the observed result. In my collaboration with AI, that gives me a concrete basis for accepting or rejecting a proposal. The examined corrections and reference artifacts support this method; I do not claim that an identical checklist governed every product change.

The new projection case illustrates one such boundary. Four boxes and four items share a numeral but fail the declared preservation relation. A model or compiler that preserves only the number has not preserved the meaning. A generated program may also compile while

All rights reserved · Research exposition

silently narrowing numeric values. Those are questions for explicit validation, not confidence inferred from fluent explanation.

Integration, operations and continuing responsibility

My responsibility extends to the seams between components: deployment configuration, authority boundaries, infrastructure transitions, recovery and acceptance of the integrated behavior. AI contribution can be substantial at each stage, and I remain responsible for the system I direct and accept. I make no claim about a staffing equivalent, productivity percentage, development duration or autonomous staffing history.

The operating environment connects persistent institutional work to external action. An attempted action, an acknowledged request and a completed obligation have different acceptance conditions. The uncertain-effect dossier shows why a missing acknowledgement cannot prove that nothing happened. The existing continuity exhibit supplies a more concrete public reference with a controlled witness and restart reconciliation. Neither is substituted for a production customer experiment.

Recovery also does not automatically restore permission. A persistent institution may retain its name and history while its execution incarnation changes. The epoch case makes that distinction explicit. Likewise, a backup observation does not establish a successful restoration with matching history and continued obligations. These are separate engineering tests to demand before broadening an operational claim.

Credit and novelty

Protection principles, causal ordering, provenance, durable execution, formal methods and proper scoring rules are foundational contributions by other researchers. The prior-art comparison and bibliography identify those works and their limits. My contribution is an institutional synthesis and scoped engineering investigations. I do not claim to have invented every ingredient or proved superiority over other approaches.

My account of Cuttlefish and Freehold is specific to those systems. Capstone contributed to the separate Moguls platform. They were not my engineering team for Cuttlefish or Freehold, and I do not use their work on Moguls as evidence for these systems.

Acceptance and review

Edition 2026.10.7 added twelve synthetic investigations. I ran their public teaching cases with AI assistance and disclosed the material needed for others to reproduce them. I do not claim independent reproduction. The public evidence schema separates engineering maturity from assurance and rejects missing provenance and unsupported escalation. Build success establishes successful assembly and checks within their scope, not the validity of every research claim or permission to publish.

Previously issued PDFs, archives, manifests and source companions retain their exact bytes and URLs. New artifacts use edition-specific paths. The correction ledger explains how current qualifications relate to older records without rewriting them. Before publication, I

All rights reserved · Research exposition

require a public-only build, a separately executed private semantic disclosure review and my approval of the exact release bundle.

For a specialist review, start with Research Depth & Proof, select a claim, run its positive and deliberately faulty case, and compare the observation to the stated nonclaims. Use the review guide to submit a counterexample or a missing foundational reference. The strongest feedback identifies exactly which inference fails and what evidence would settle it.

All rights reserved · Research exposition