Verifiable Knowledge

A protocol for trustless agents.

June Kim
Independent Researcher
june@june.kim
ORCID 0009-0005-3153-9396

June 13, 2026

Abstract

Large language model (LLM)-based agents cannot be held accountable. Even with persistent memory and full provenance trails, their reasoning disappears with the context window. The burden of proof is on whoever drives the agent. Each agent, instead of attesting its own work, must present every claim with a falsifiable condition that can be reproduced to the same verdict. This we call verifiable knowledge. Belief, knowledge, and truth reduce to structures an agent constructs and another checks. Verifiability is transitive, so their results are reproducible. In this protocol, accountable failure outranks unaccountable assertion. The epistemics is borrowed from What Cannot Be False Cannot Be True; here, we introduce a protocol to apply it. Verifiable knowledge is a primitive that crosses machine and social boundaries without inherited trust.

The operationalization of What Cannot Be False Cannot Be True, carried to a population of agents. It runs the frame as a protocol; the data structure that instantiates it is The Hypothesis Graph.

1 Introduction

How did you feel when your coding agent told you that it was done, but it clearly wasn't? It said done but it never checked if it was. The word meant nothing. Anyone can justify a belief to themselves; untested, it stays untrue. What Cannot Be False Cannot Be True presents this argument. Belief, knowledge, truth: their bitwise representation doesn't distinguish them. So what does? How the data became entitled is the proof, and that proof is bitwise: the test results the claim survived. That's how a machine verifies knowledge.

Confidence is a vibe, and a vibe doesn't encode. I'm absolutely sure has no bitwise form another agent could check, none to verify tomorrow, none for anyone else. That's the problem of knowledge interop: one agent makes a claim, another must trust it blind or start from ground zero. Anywhere in between needs a representation of knowledge, a semantic memory, for partial work to be checked. A chain of attestation breaks at a single forged link, and a chain of independent Bayesian credences, each below one, multiplies toward zero. Neither survives a distrusting auditor. Is there a truth contract that does?

A contract is the protocol by which a session's verdict survives across agent boundaries and persists between context windows. The obvious move is to store the attested output: keep each artifact with its provenance and trust what's on disk. That is a cache, and a cache is fine as long as a miss can be recomputed. Storing outputs is inert unless it can be regenerated. Coding agents lie: they report a test suite as passing when it is failing, and the bare output inherits the lie. Re-derivability outranks the stored verdict.

Several directions reach this research area at once. A recent DeepMind position paper names "robust falsifiability" as a core requirement for trustworthy artificial epistemic agents (Marchal et al. 2026), the verification gap this protocol fills; systems like NARS (Wang) and Traxia (arXiv:2606.08256) reach it from non-axiomatic reasoning and agent-native publishing. Verifiable knowledge is the standard those efforts

presuppose, the contract under which one agent's claim becomes another agent's checkable inheritance. Here, we offer the verification primitive.

2 Truth at the edge

What does it mean for knowledge to be verifiable? A claim is a semantic node, fixed less by its content than by how it stands to other claims: what it depends on, what it implies, and what would refute it. Those relations are its edges, the citations and inferences. Entitlement, the justification for a claim and its provenance, does not live in the node. It lives in the edges. An edge is a kill condition, a refutation that propagates. This is Brandom's inferentialism as a graph, entitlement as a matter of inferential relations, a claim's place in Sellars's space of reasons rather than a property sitting inside an isolated representation.

This is not a knowledge graph in the established sense. There a node is an entity and its uncertainty, if recorded at all, a stored confidence score; here a node is a claim that carries its uncertainty as a testable condition, the check that would refute it.

The tautology is the limiting case: its irrefutability and its uselessness are one property, a single node wired to nothing. It keeps its inferential edges inside the formal graph (the two graphs); what it lacks is a system-facing kill edge.

The kill condition travels two ways, from the system to what it tests, or from a source to what cites it. A citation makes a belief inherit the fate of its source, so naming a source is handing over a target. A failed test invalidates one or more edges without naming which, the Duhem-Quine underdetermination, so the next test disambiguates; a claim with a second surviving edge routes around the loss, where a single chain would snap at one forged link. The dispute moves up the chain, from whether to believe the claim to whether the source holds, a question you can put to the source. And you can make it cite its own. Each citation is a link, and falsifiability is the chain being climbable link by link. Truth is not the top but that you can always climb one more. The mapping is mechanical:

- Provenance is the dependency graph.
- Citation is an edge to what the claim rests on.
- Attestation is the signed check log, the line that says I ran this, here is the receipt.
- Falsifiability is the check being able to fail, the test whose firing is the claim's kill condition, the operational form of what the companion paper calls refutation.
- The test is the system pushing back.
- Truth is the check currently passing.
- Reproducibility is whether anyone else can rebuild it from source.

That mapping produces a ranking. Even a Bible verse citation cites its provenance and names its axiom openly, a complete stack trace you can follow to a root that is an axiom and not a measurement, then decide for yourself whether to accept it.

By contrast, a withheld measurement cites a procedure it will not show, a dangling pointer where the evidence should sit. On provenance, and only on provenance, the withheld measurement ranks below the scripture citation: more accountable, because naming the axiom hands you the principle to reject, though no more falsifiable as a claim about the world. A disagreement resolves where its axiom is exposed, not where a proxy buries it: whether hiring is a matter of merit or of humanity is arguable in a way that a hire/no-hire threshold, which has already chosen merit, is not. The claim that can be argued with is worth more than the claim that hides the target. And the checkable guardrail holds at this scale too: checkable never means manufacturable to spec, the check has to carry a test that can fail, and a check whose test can never fail is `return 0.70`, a mocked pass.

3 The entitlement ledger

How do claims carry their entitlement between agents? Make them trivially runnable. A claim is a hypothesis: it names the test that would refute it. The check runs that test, and searching for a proof, or a patch, or a measurement, is running that program; it returns one of three things. Green is true, the check passed. Red is false, the check ran and broke. Hung is untrue, the check that never returned, or the claim shipped with no check to run at all. True and false are siblings because both are halting states, computations that came back with a verdict; untrue is the one state that is not a verdict, the job still spinning, the test suite that never finishes. A red outranks a hung: a check that ran and broke tells you where, while a claim shipped with no check tells you nothing, even when it happens to be right. That is why accountable failure outranks unaccountable assertion.

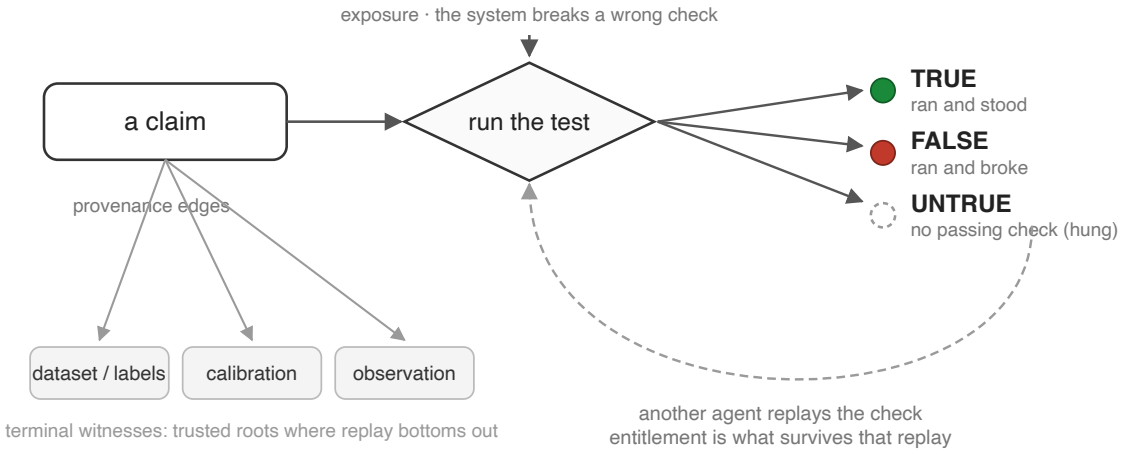

Figure 1: How knowledge is compiled. A claim links by provenance edges down to trusted roots, then runs its test, the point where the system can break it. The check returns one of three states: true (stood), false (broke), or untrue (no passing check, the hung state). The loop is the move a machine adds: the check is re-run by another agent, so entitlement is what survives the replay rather than what an agent grades in itself.

Every verifiable claim exposes itself to verification:

Claim	The check, as a program	Output	Refuted when
"it was 12°C at SFO at 14:05Z"	a logged reading from the named weather API	`12`	an independent source disagrees for that timestamp
"the run logged 3 errors"	`grep -c ERROR run.log` on the named image	`3`	the command prints another count
"$7 \times 72 = 504$"	evaluate `7 * 72`	`504`	recomputation differs
"the patch passes"	the suite at commit `a1b2c3`	`exit 0`	any test fails
"the theorem holds"	rechecked in a proof assistant	`no goals`	a step fails to check

The weather reading bottoms out at an observation no later run can re-derive; where replay cannot reach, the check is corroboration against independent ledgers, a second provider or a neighboring station, the reading refuted when they disagree. It is how a distributed ledger settles a fact at all: no central authority, truth bottoming out at a cross-referenced history that independent replicas agree on. The computation and the proof settle for good by re-running. Verifiability is graded by how firmly the program pins its roots, not by the kind of claim.

Encoded this way, the verdict doesn't depend on self-assertion; entitlement is conferred by replaying the check to a pass, with no author grading its own work. Entitlement here runs backward: the replay re-derives a verdict that already stood, climbing provenance to roots, not forecasting whether the claim will pay out. A claim record carries what the replay needs: the claim, the provenance edges down to its roots, the check procedure, the kill condition, the declared terminal witnesses, and the attestation that signs it. A receiver inherits knowledge after verifying its claim.

An argument settles by a mediating oracle both sides accept, where one exists. Independent verification is that oracle, trustless because the verdict comes from re-running the typed check, not from either party's word.

How cleanly it settles then depends on how completely the roots are typed. Pin every terminal witness, and replay is deterministic settlement, the same verdict for anyone who runs it: a unit test re-run against a

repo at a fixed commit, a task benchmark scored by a bash command on a named machine image, a proof rechecked by a proof assistant. Leave a root untyped, and settlement decays toward dispute, down to a withheld benchmark whose patches never ship, a claim that settles for no one.

Where no oracle exists at all, no reachable test and no check that could run, there is nothing to settle: the claim stays untrue and the dispute open, the third state doing its work. So verifiability is graded by typing: the more strictly a node pins its roots, the more its verdict settles by replay instead of by trust.

4 Triangulation

The ledger records a verdict. But who is entitled to write one? A single agent holds one lossy projection and cannot invert it. So its own artifacts and the world's structure are indiscernible to it, and grading itself grades a fiction.

This is Davidson's triangulation: the distinction between subjective and objective, and with it the concept of error, requires at least two minds and a shared world. A multi-agent view is multiple projections of the one object, and comparing them constrains the object no single projection reveals, with the view from nowhere (Nagel) as the limit no projection occupies. The blind men and the elephant is the picture: no one holds the animal, and only diverse touches approach it. Agreement among agents that share a blind spot is an echo chamber, so the checker has to be an independent projection. But the parable cheats. We outside know it is an elephant; the men inside never get that view, so they have to establish that the touches are of one animal before they can add up. Triangulation buys constraint, and even that on credit, against a shared object not yet established.

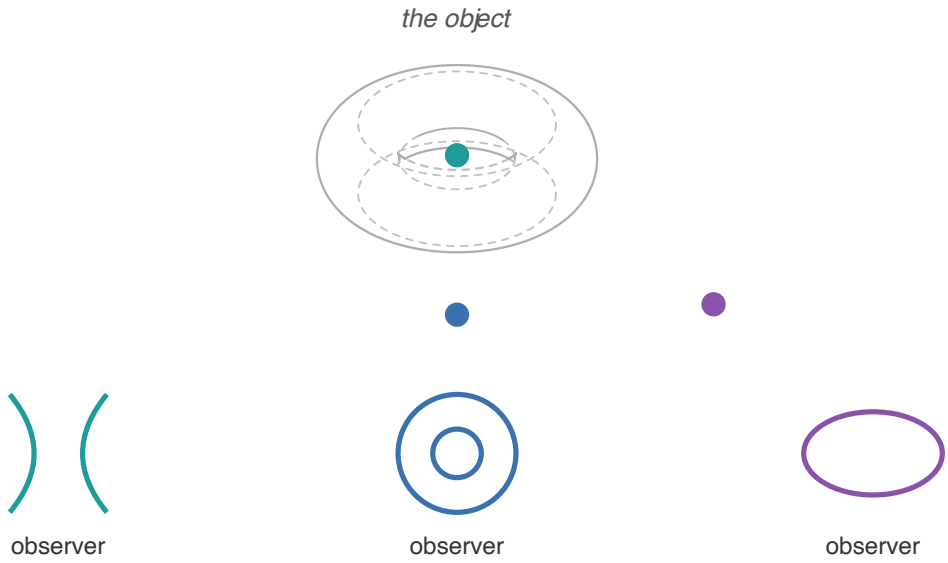

Figure 2: Triangulation. One solid, a torus, seen from three places. From below the axis it is a ring; from an oblique angle the hole vanishes into a solid oval; from inside the hole it is a hyperbola, the saddle wall flaring away, with no ring or hole in sight. No one view reveals the torus, and each alone misleads, yet the independent projections together pin the shape none holds whole. The observer in the middle is the sharpest case: surest of what it sees, least able to tell it sits inside a donut.

That independent projections constrain the object is the step taken here. Each additional constraint reveals the shape further.

The one move a machine adds: entitlement is conferred not by an agent grading itself but by another agent replaying the trace. What a human community of inquiry pays for in years of independent re-checking, a machine pays for in a single re-run: it emits a replayable trace, and another agent re-runs the same check now. A machine improves its entitlement by becoming checkable by another projection, not by getting smarter.

So it optimizes for verification cost rather than self-attestation, laying its methods and claims bare for the recipient to inspect. That openness, shared across a population, is what the protocol presupposes.

Replay breaks the who-trusts-the-truster chain that attestation spawns: the verdict is the check's, not the checker's, so no one needs to check the checker, only re-run the audit. Independent constraint improves entitlement, no strong objectivity smuggled in, since the replay is itself one more lossy projection.

Figure 3: The attestation regress. An agent's word rests on an attester, whose rests on an auditor, whose rests on a checker, and so on: who checks the checker? The chain never grounds. Replay cuts it: the verdict comes from re-running the check, so any agent re-derives it without trusting a truster.

For another agent to check rather than merely disagree, the projections have to compose, and that is what checkability buys. It also supplies the shared object triangulation had to assume: a replayable check lets one agent re-run another's touch and feel the same thing, so the two touch one object instead of guessing they do, and partial views add where they would otherwise collide. Constraint becomes convergence at the replay: triangulation narrows the object, the shared check closes the gap to a single verdict.

Without it, multi-agent epistemics degenerates into the blind-monks debate, each agent asserting its own projection, snake, tree, nothing replayable to reconcile them, a deadlock that reads as irreducible relativism. The parable is a tragedy only because the monks trade assertions instead of replayable touches. The single-knower guardrail was a test that can fail; this is the community version, replayable checks that compose where bare assertions deadlock.

5 The canon

Cheap re-checking is wasted if every agent re-derives everything. The payoff is accumulation: what one agent establishes, and any other can re-check, is kept once and built on rather than proven again. A population that can re-run the same checks forms a shared frame, united not by shared beliefs but by shared protocol, agreement on the method of adjudication. Agreeing on the method, they verify each other's work without trusting each other, and accumulate a canon, the union of standing hypotheses, each re-checkable by any member, Peirce's community of inquiry made durable.

Re-checkable need not mean re-checked. The canon keeps each hypothesis's last passing verdict, so a later agent reads the result instead of re-deriving it. The cache pays most where the check is a search over a large space: finding the proof or the patch costs the first time, reading the verdict it returned does not. So the ledger entry stands in for trust without asking for it, an agent proceeding on the recorded verdict as on a trusted word, the check still one replay away, and no one re-searching a space already searched.

The Pythagorean theorem is the oldest entry that still reads this way. It ships with its proof, so no one inherits it on anyone's word: reconstruct the proof, the same for anyone who checks. Almost no one re-derives it. A 2,500-year-old claim that travels with its check cannot devolve into because the ancients said so.

Membership is provisional by construction: standing hypotheses, not settled truths. It is a canon of the activity, the hypotheses that still pass, not the institution, a corpus held true by authority. Held that way, an entry can outlive the check that earned it, entered the record drifting into true, nothing binding a later correction to travel as fast as the claim it corrects. A canon of standing hypotheses cannot drift so: every entry carries its test, so pull an entry's root, a dataset retracted or a calibration shown wrong, and it fails on its next replay, and so does everything downstream that cited it, the refutation spreading up the citation edges. Claim and check never came apart, so the retraction propagates mechanically.

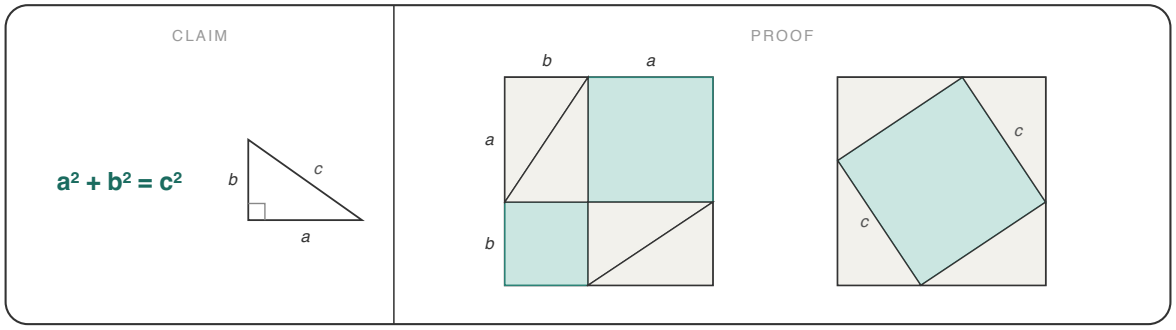

Figure 4: The rearrangement proof. Two squares of equal side hold the same four right triangles; on the left they leave an a-squared and a b-squared square, on the right a c-squared square, so the two smaller squares together fill the larger.

6 Inheritance without gatekeeping

Entitlement displaces reputation: the mechanism under it is that a credence is a shortcut over a verifiable substrate, never the source of entitlement. Each canon entry carries its own replayable kill-condition, so it still passes is checkable by anyone at any time, replay catching it rather than a committee.

This does not abolish credentials, it regrounds them. A credential is an attested check certification: a degree, a review stamp, a trusted-maintainer badge means this passed checks I verified, a cached pointer to a replayable check. Deferring to it is running a package you did not compile yourself, rational under cost (§7) precisely because the substrate stays verifiable underneath.

So the claim is not no gatekeeping, since a deployed system still needs spam control, identity, admissibility. The point is narrower: the credence shortcut points at entitlement but never sources it; entitlement lives in the certified check. You trust a bridge because it stands, physics because the rockets do not explode, a theorem because the proof checks, not because the engineer was credentialed, a journal approved the equations, or a famous name signed it. A credential is worth exactly the check behind it, and nothing once detached.

The pathology is the detached credential, an attestation with no check under it, the dangling pointer again, a Yahoo CEO listing a computer-science degree the college never granted. He got caught. What are the odds he was the only one lying? It is what gets captured, gamed, or inverted into rejection by identity, a claim judged by the name attached rather than the check behind it. That inversion is the same detachment as acceptance-by-credential: both let the verdict come apart from the check, one to wave a claim through, one to wave it away. The protocol keeps the credential anchored rather than banning it: nullius in verba as a check step.

For that to hold, agents need a stance: truth is not blind inheritance of canon. An agent that accepts an entry as true because it is canonical has reinstalled the gatekeeper it was supposed to retire. The stance is fallibilism (Peirce) about the canon itself, every entry standing trial forever, none graduating past a revisable standing hypothesis.

Structure makes that livable: the canon stays verifiable for the entire graph, down to declared terminal witnesses (§7), not just at the leaves. So inheritance is never blind, even when unverified in practice: you use the canon without re-running it, the efficiency the social layer rests on, but the option to verify any entry at any depth stays live, needing no gatekeeper's permission and no original author. The difference from a credence canon is not that you always verify, but that you always can: inheritance revocable by replay rather than permanent by authority.

That replay works across agents with no shared clock. Each check carries its own causal order, provenance edges running strictly backward from a claim to what already stood when it was made, so any agent replays it later, in its own frame, and reaches the same verdict. That partial order is all the synchrony the protocol needs: Lamport showed a distributed system carries only a happened-before order, no shared physical clock, the way data centers already run.

7 Limitations

Three limits, each a cost of removing the gatekeeper.

First, replay bottoms out. Every empirical check terminates somewhere: sensor calibration, dataset integrity, a human observation, an instrument log, hardware, a random seed, an API response, an institutional attestation. The protocol does not abolish trust anchors, it makes them explicit and attackable. A root is admissible when it is typed, signed, reproducible where possible, independently cross-checkable where not, and kill-conditioned by calibration or contradiction.

An eval bottoms out at its dataset's labels: replay can re-run the scorer against them all day, but it cannot re-derive whether the labels were right, only check them against a declared source and a calibration that is itself open to challenge. So "verifiable all the way down" means the chain replays down to declared terminal witnesses.

A reliable process and a true claim differ here: a passing check entitles the claim, while entitlement about the check machinery itself is its own node, on pain of letting the check passed harden into a new authority.

Second, naive replay assumes good faith, and a machine-native epistemics has to survive its absence: forged logs, poisoned provenance, sybils, collusion, benchmark overfitting, selective disclosure. The sharpest form is the defeat device, the system that detects the test and passes only it, green on the bench and red everywhere else, and its machine versions are a forged attestation and a provenance edge poisoned at the root.

Independence is the defense, since diverse projections resist a shared-bias capture (Condorcet's jury theorem is the canonical form: independent judgments aggregate, correlated ones do not), but it is not free. It has to be engineered, across model families, across operators, with randomized challenges. That is the price of regrounding credentials: removing the trusted gatekeeper raises the adversarial-robustness bill, which the replay substrate and engineered independence pay rather than a gate.

Third, always can verify holds only where replay is feasible. Verification can be computationally, financially, legally, or physically prohibitive. Re-running a ten-thousand-line proof is feasible; re-running a climate model, a drug trial, or a decade of accelerator runs is not, because the refutation there needs a fresh world-trial rather than a replayed command. So most of the canon in practice travels the credence shortcut, because full replay is expensive, and that deference is the normal rational mode rather than a failure.

Entitlement improves when the replay cost is finite and declared. The protocol guarantees the option to verify; it does not guarantee the labor of verifying. A canon that hides its replay cost is as opaque as one that hides its provenance.

8 Related work

The frame's lineage, Kant and Peirce and Ramsey and Dummett and the rest, is named in What Cannot Be False Cannot Be True (DOI). Verifiable knowledge's own borrowings are the operational ones, and the borrowing is the point.

Brandom and Sellars supply entitlement in inferential relations, the space of reasons, which the edge picture renders as a graph. Davidson supplies triangulation, the objective needing two minds and a shared world, and Nagel the view from nowhere as the unoccupied limit. The nearest tradition is the one that made epistemology checkable before: AGM belief revision (Alchourrón, Gärdenfors, Makinson 1985) specified rational belief change as postulates, and truth-maintenance systems (Doyle 1979; de Kleer's assumption-based TMS 1986) ran dependency-directed retraction, both protocols over what to hold given what supports it. Neither carries a three-state entitlement ledger, a system-facing kill condition, or replay by a distrusting party; they revise a believer's own commitments rather than transmit a claim across an agent boundary.

On the machine side: NARS is the nearest non-axiomatic cognitive architecture, with experience-grounded graded truth revised by experience; it stops short of a replayable trial, an entitlement or provenance graph, a three-state ledger, and replay by a distrusting party. OpenCog's AtomSpace with PLN is a typed hypergraph carrying truth values, but the truth is a stored label, as opposed to a replayable check. Traxia is concurrent work on agent-native scientific publishing, signed identities and provenance and a replication record; it stops at infrastructure rather than epistemics, with no three-state ledger, no stakes-threshold knowledge, no falsifiability-as-structure, and its convergence is evidence rather than threat. Nanopublications (Groth et al. 2010) attach machine-readable provenance to claims, but descriptively: the evidence is not a check another agent can re-run.

The nearest prior art is not a cognitive architecture at all, but the reproducibility stack: executable research papers, proof-carrying code, software supply-chain attestation like in-toto and SLSA. Each runs a real piece of the contract, a signed build, a checkable provenance chain, a reproducible artifact. What none makes one thing is the semantic claim-states, the kill condition, and independent verification together: they attest that a build happened. None attests that a belief earned its entitlement.

The strongest objection: This is Brandomian inferentialism plus Davidsonian triangulation plus AGM belief revision plus executable provenance infrastructure. The philosophical claims are inherited, the machine claims are ordinary reproducibility engineering, the three-state ledger is old many-valued bookkeeping, and the result is a useful architecture, not a new epistemology. Concede every piece.

The difference is the exact contract read as agent-knowledge semantics. No prior system assembles all six of these into one contract for an agent's knowledge:

- replayable check — the verdict re-derived by running it, not asserted
- provenance edge — each claim wired to what it depends on
- kill condition — the test whose firing refutes it
- independent verification — replay by a party that need not trust the author
- entitlement ledger — the three states a claim can hold
- check-time canon admission — entry earned by passing, not granted by authority

It is a useful architecture, not a new logic or metaphysics; that, narrowly scoped, is the claim. The primitives are natural and each has a canonical citation; the contribution is their assembly. The convergence noted in the introduction is independent work arriving at the same need.

9 Future work

Future work is the outward falsifiability edge, not "more research": the epistemology becomes falsifiable only by being built and used in real agent systems, where its claims about checkable entitlement, replay-triangulation, and a gatekeeper-free canon meet a world-facing trial and can fail. An epistemology with no application edge is a detached node, irrefutable and therefore useless. The witnesses are already on the edge: the hypotheses turned into a graph, the verification ledger that records each check, the abductor that raises the candidates, the agent harnesses that run the trials.

Everything past that edge is an open node, not a result: a conjecture that names its own test. What the body scoped out lives here, as open edges, not in the proven core.

9.1 Settlement and stakes

A claim that settles without anyone's trust is a claim anyone will stake on, so a fully-typed node would be, in the limit, a position in a market. Settlement by replay is what a prediction market calls its oracle, here trustless because the oracle is the typed re-run, not a named authority, and a node's price becomes the population's credence in it, Ramsey's odds made literal. The two graphs bound what is bettable: a strictly-typed formal node settles cheap and final by replay, an empirical one settles only on a fresh world-trial, rate-limited by world-contact, and a node with no oracle does not settle and cannot be staked, untrue and unpriced. Whether such markets sharpen credences or merely price slop is the open test.

Under it sits one conjecture, an open node: verifiability is the entry condition for knowledge held in common. Between agents who do not trust each other, a claim becomes shared only as a check others can re-run; what no one can re-run stays private, or stays untrue. The refutation is exact: exhibit shared knowledge that scales and survives bad faith with no replay beneath it, and the condition was never necessary.

10 Conclusion

What is new here is the contract: the executable semantics under which one agent's claim becomes another's checkable inheritance. The frame it runs on is the companion paper's, and every primitive is borrowed; the assembly is the contribution.

The contract makes one empirical bet: agents that adopt it should be more accountable to each other, and better coordinated, than agents that do not. That bet is falsifiable and still unpaid, settled only by building the protocol into the systems that would run it.

Knowledge, for a machine, is the hypothesis another agent can re-check, linked to its grounds instead of attested by its author, and a population that agrees on the replay can hold a canon without holding a

gatekeeper. That is what it is for knowledge to be verifiable: re-runnable by another, and re-runnable again by the next. Show your work, make it checkable.

References

Canonical sources verifiable knowledge rests on. The frame's sources are listed in the companion paper; the author's companion essays are under Provenance below as lineage, not as entitlement.

- Alchourrón, C., Gärdenfors, P. & Makinson, D. (1985). "On the Logic of Theory Change." Journal of Symbolic Logic 50. (AGM belief revision)
- Brandom, R. (1994). Making It Explicit. Harvard University Press.
- Condorcet, M. de (1785). Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. (the jury theorem: independent judgments aggregate)
- Davidson, D. (1982). "Rational Animals." Dialectica 36; and (1991) "Three Varieties of Knowledge."
- de Kleer, J. (1986). "An Assumption-based Truth Maintenance System." Artificial Intelligence 28.
- Douceur, J. R. (2002). "The Sybil Attack." Intl. Workshop on Peer-to-Peer Systems (IPTPS).
- Doyle, J. (1979). "A Truth Maintenance System." Artificial Intelligence 12.
- Duhem, P. (1906). The Aim and Structure of Physical Theory; and Quine, W. V. O. (1951). "Two Dogmas of Empiricism." Philosophical Review 60. (underdetermination: a refutation condemns the bundle, not a named premise)
- Goertzel, B., Iklé, M., Goertzel, I. & Heljakka, A. (2008). Probabilistic Logic Networks. Springer. (OpenCog / PLN)
- Groth, P., Gibson, A. & Velterop, J. (2010). "The Anatomy of a Nanopublication." Information Services & Use 30.
- Lamport, L. (1978). "Time, Clocks, and the Ordering of Events in a Distributed System." Communications of the ACM 21 (7).
- Marchal et al. (2026). "Architecting Trust in Artificial Epistemic Agents." arXiv:2603.02960.
- Nagel, T. (1986). The View from Nowhere. Oxford University Press.
- Peirce, C. S. (1877). "The Fixation of Belief"; and (1878) "How to Make Our Ideas Clear." Popular Science Monthly.
- Sellars, W. (1956). "Empiricism and the Philosophy of Mind."
- Traxia (2026). Agent-native scientific publishing. arXiv:2606.08256.
- Wang, P. (2013). Non-Axiomatic Logic: A Model of Intelligent Reasoning. World Scientific. (NARS)

Provenance

These arguments were first worked out informally on the author's blog and are reproduced here in operational form, so the protocol stands on its own as the companion; the posts are listed as lineage, not as entitlement.

- What Cannot Be False Cannot Be True: the frame verifiable knowledge runs, the trichotomy and the type-split and the buildable-truth core.
- Truth Is Buildable (2026-06-04): the build mapping (provenance, citation, attestation, test, reproducibility), truth in the edges, and the second-model blind-spot note.
- Science on Trial (2026-04-19): every claim stands trial forever, activity versus institution, the four terms (publication, peer review, replication, truth), the citation graph with no reverse gear, trust by checkability rather than by credential.
- Sour Red Tapes (2026-06-01): merit attaching to the work, nullius in verba, delete the author and the receipts stand.
- Complementations (2026-05-09): judging a claim by the name attached, the acceptance-or-rejection by identity the protocol regrounds.
- Auditing DeepSWE (2026-05-27): the motivating empirical case, and the dual of the consistency-without-test error.
- Modes of Reason and Abduction (2026-05-04): the three modes of inquiry behind the abductor.
- Compress and Unfold (2026-06-10): generation as the unfold, filtering as the fold.
- The Hypothesis Graph (2026-05-28): the data structure this protocol is the semantics for, and the application edge on which it can fail.

License