NINCHI

Verified Human Understanding as Cognitive Infrastructure

A shared accountability primitive for AI-assisted work and learning

Version 1.0 — June 2026

Scope and status. This paper defines the category Ninchi builds in and the model beneath it. Throughout, it separates what is implemented today from what is in development. The proposed model in §8 and the validation program in §10 describe the next generation of the system; the production system today uses the transparent score of §7. Claims of calibration and prediction are targets until validated.

SHIPPED TODAY

Generated, artifact-specific, timed challenges over GitHub, GitLab, and Bitbucket Cloud; hidden prose rubrics; pass/fail scoring against a configurable threshold; the difficulty-weighted Ninchi Score; policy modes gating on that threshold; opt-in enforcement; an attributable scored event record per event; org analytics over those records (overview, trends, per-member and per-repo views); Pro customization — custom verification focus, Teach Me micro-lessons, copy-paste lockdown; re-push challenges that bias questions toward the new commits.

IN DEVELOPMENT

Tamper-evident hash-chained records; confidence-based routing; criticality policy; human review, disputes, overrides; a calibration program against human adjudication; the latent/identifiability estimation layer; org-level onboarding-curriculum generation.

Contents

1 The Missing Layer in AI-Assisted Systems

Writing software used to be the hard part. If your name was on the code, you'd almost certainly understood it along the way — the work vouched for you. AI broke that link. A model can now turn out a working diff, a clean brief, or a convincing analysis faster than the person who signs off on it can read it closely enough to explain it. Producing the artifact became cheap. Understanding it stayed expensive. And that was always the part that mattered.

So whether AI helped is rarely in question anymore; it almost always did. What matters is whether the accountable person can explain the work before it moves forward. When a reviewer merges the change or a manager signs off and neither can, they take on a risk no one can cleanly assign, inspect, or defend later — working code in production that no one ever fully traced.

> "PR merged" no longer means "code understood." "Assignment submitted" no longer means "learning occurred." *Submitted* is no longer evidence of *understood*, and the two now have to be evidenced separately.

Ninchi defines a shared primitive for that point of accountability, one loop that runs the same way every time: an artifact, a generated comprehension challenge, a human explanation, a rubric-scored event, a record, and the decision that follows. The trail those steps leave is what this paper means by verified human understanding: an auditable layer showing that a named person could explain the AI-assisted work they were accountable for, at the point of decision.

None of this is unique to software. AI-assisted output lands in a system of record under a person's name in plenty of forms: a merged change that ships to the codebase, a graded submission that goes on a transcript, a signed-off decision that commits real money. Each is a moment where an institution stands behind a person who may not be able to explain what they put their name to. Ninchi exists to make the evidence for that accountability reviewable instead of assumed. Over time, those events compound into organizational memory — an inspectable record of where demonstrated understanding exists and where it is thin.

2 The Verified-Understanding Event

The atomic object is the *verified understanding event*: an attributable, time-stamped observation that connects a human actor, an artifact, a generated question, a hidden rubric, a free-text answer, and an evaluated outcome.

For an event i we record: actor identity, target file / diff snapshot, question, hidden prose rubric, answer, score, pass/fail, difficulty, tags, threshold, feedback, key points hit and missed, time taken, and timestamps.

The event is deliberately narrower than a performance review. It does not assert that a person is generally capable. It asserts that, at a specific time, for a specific artifact, under a specific policy, the accountable human produced evidence of understanding. Three properties make the event useful where ordinary review metadata is not: *attribution* (it binds a named person to an artifact and an answer), *temporal specificity* (it captures understanding at submission, not in hindsight), and *auditability* (the recorded event — question, hidden rubric, answer, score — is available for later inspection).

3 Artifact to Decision: The Challenge Loop

Every event is produced by the same loop, regardless of surface:

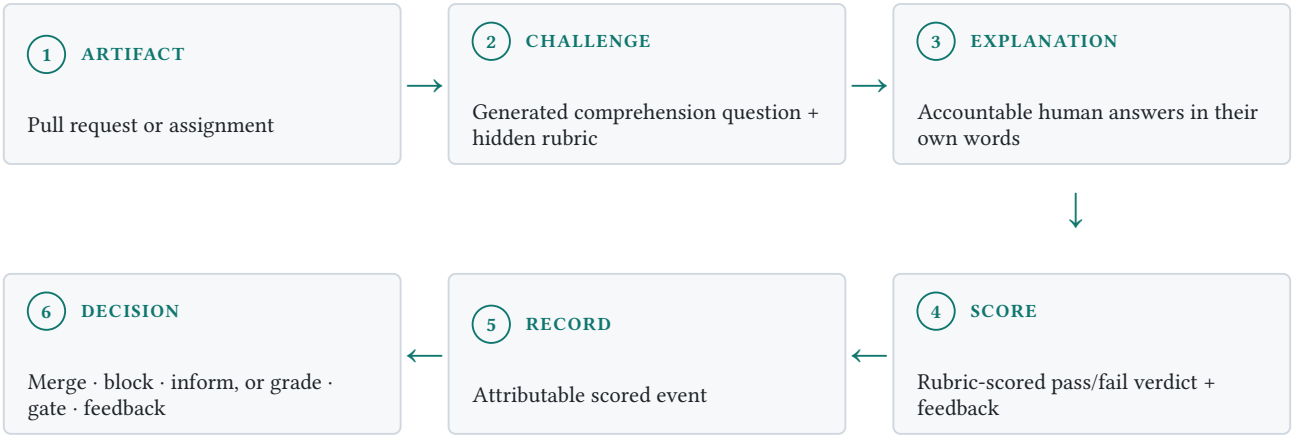

Figure 1. *The verified-understanding primitive runs identically on every surface: a pull request in engineering, a project or assignment in education. The surfaces differ only in the artifact and in what the final decision means.*

Ninchi analyzes the artifact, generates a small number of focused questions about this specific work, asks the accountable human to answer in their own words, scores the answer against a hidden rubric, records the scored event, and returns a verdict that a downstream policy can act on. The loop is intentionally boring and identical everywhere; the surfaces differ only in what the artifact is and what the downstream decision means.

The design rests on a practical learning-science distinction: recognition is cheap, explanation is harder. People can feel fluent with a mechanism until they have to explain its steps, assumptions, and failure modes in their own words; that gap is the *illusion of explanatory depth* [1]. Ninchi uses explanation because it exposes missing understanding more reliably than recognition does, while staying narrow enough to audit against a rubric. Generating the explanation is also one of the better-attested ways to consolidate the understanding it probes [2].

4 Can You Trust the Grader?

A system in which an AI grades a human's understanding has to answer one question before any other: *why should anyone trust the grader?* Using an AI to grade human understanding of AI-assisted work raises an obvious regress. We mitigate it by making the grader the most bounded and auditable component in the loop.

Two design commitments do the work today. First, the rubric must adjudicate whether an explanation is *correct*, not merely fluent: an articulate, confident, wrong answer is the human analogue of a model hallucination, and a grader that rewarded it would be worse than useless. Second, the verdict's authority is bounded by the policy mode and the score threshold: a verdict only does what the configured mode lets it do, and teams watch results in Tracking before enabling Blocking. The trust design is bounded: Ninchi records evidence of demonstrated understanding under a defined challenge — an operational signal tied to a specific artifact, not a general mental trait. And every event is recorded for later inspection.

IN DEVELOPMENT

Pro already computes and stores an evaluator confidence score on each verdict; calibrated, confidence-based routing is the next control built on top of it — low-confidence or high-stakes events route to human review rather than being decided automatically. The full exchange opens to dispute, sampling, and adjudication, and those disputes feed back as calibration data for the generator and evaluator.

There is also a structural reason the grader's task is more tractable than it first appears: it is *asymmetric*. Generating correct code from scratch is open-ended and error-prone; checking a short human explanation of a specific diff against a constrained, hidden rubric is far more bounded. The evaluator is not asked to be an oracle about the codebase. It acts as a strict rubric-checker, confirming that a given answer contains the specific claims the rubric requires for this change. We hold this as an intuition about relative difficulty, not a proof. A skeptic can fairly object that open-ended free-text grading is itself hard, and the LLM-as-judge research supports the caution: agreement studies validate judges against human chat preference, not comprehension grading [3]; reliability is task-dependent and must be checked per task [4]; and judges carry known artifacts such as position bias [5]. Whether the grader is reliable enough to gate on is an empirical question we treat as a validation target (§10), not an assumption.

> The grader is not assumed to be right. It is made *auditable*: every verdict has a question, a hidden rubric, an answer, a score, and a recorded event a reviewer can inspect.

A verified-understanding event can also become performative: a person may learn to satisfy the rubric without durable understanding. We treat that as a measurement risk, not a solved problem, which is why questions vary and challenges are timed and artifact-specific. When work is revised and re-pushed, the regenerated questions are biased toward the new commits, so each challenge stays tied to *this* change rather than rewarding a once-memorized answer. Because every scored event is recorded, repeated shallow-but-passing answers on related artifacts become a visible pattern that reviewers and policy can act on. Gaming any test is possible; here, durable patterns of it become part of the evidence.

5 One Engine, Many Artifact Types

One engine handles both production and training work. A pull request and a training assignment are different kinds of artifact, but Ninchi asks whoever submitted each the same question: can you explain what you submitted? Only the artifact and the stakes of the decision change.

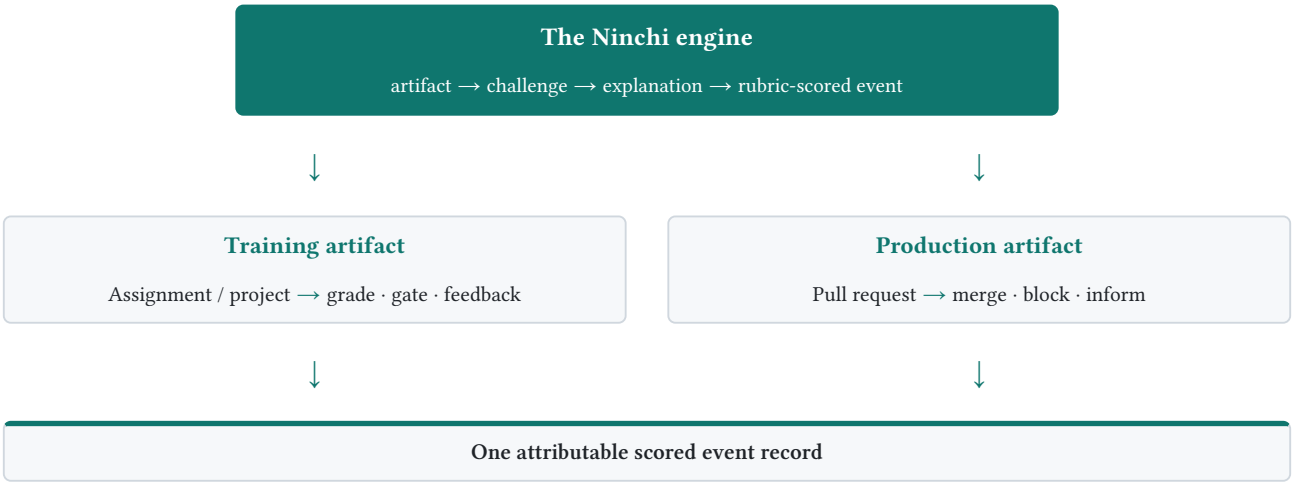

Figure 2. *One engine over multiple artifact types. The same artifact → challenge → explanation → scored-event loop produces one attributable scored event record; only the artifact and the meaning of the decision change.*

	Production artifact	Training artifact

Artifact	Pull request	Uploaded snippet or document (Direct)
Accountable human	Engineer who owns the merge	Learner who submits the work
Downstream decision	Merge / block / inform	Feedback and score; grade-gating (in development)
Concern	The codebase	The developing engineer

The two concerns differ — one protects a codebase, the other develops a person — but that is a difference of artifact and stakes, not of mechanism. Both ask the same thing: can you explain the work before someone trusts it? The engineer an organization wants to trust on a pull request is the same person a technical program was forming a year earlier. The artifact changes; the primitive does not.

6 Safe Velocity as Operating Principle

Verified understanding lets institutions move faster without losing accountability. Most AI tooling in the review loop is *passive*: a reviewer bot summarizes the diff, a generator explains what the code appears to do, and the human is left to nod along. That is the posture automation research warns against: when a system is fluent and usually right, operators stop scrutinizing it and miss the cases where it is wrong [6]. Ninchi inverts the posture. It is *active*: it makes the accountable human explain the change in their own words, scores that explanation, and records the result as an attributable scored event. Being shown an explanation and having to produce one are not the same act, and only the second leaves evidence that someone understood.

The practical effect is *safe velocity*. Because the gate fires only where evidence of demonstrated understanding is missing or weak, a team can pause the small fraction of work that lacks it and let the rest move. Engineers ship more confidently when edge cases and failure modes are explained before merge, and learners take on harder AI-assisted projects when a coach can trust demonstrated comprehension rather than a green build. Incident reduction and recovery-time impact are validation targets, not yet measured results; today Ninchi provides operational visibility into understanding coverage, making the *absence* of demonstrated understanding visible soon enough to act on.

The deeper warrant is older than AI. Bainbridge's "Ironies of Automation" observed that the skills which atrophy while automation runs smoothly are the ones needed when it fails [7], and the modern automation-bias literature sharpens the point: human oversight degrades exactly when the automation is good enough to trust [6]. Making missing understanding visible at the moment of decision therefore matters more, not less, as AI improves. Understanding is built in at the moment of production; it cannot be inspected back in after the work has shipped.

7 Scores, Thresholds, and Modes

The production system today uses a deliberately transparent score. Each scored challenge i yields a pass/fail outcome $y_i \in \{0, 1\}$ above a stored threshold τ_i, and carries a difficulty weight w_i (easy 1, medium 2, hard 3). For any scope A (actor, repository, organization, or a developer's cross-org public profile), the Ninchi Score is the difficulty-weighted ratio of passed evidence:

$$S_A = \left\lfloor 100 \cdot \frac{\sum_{i \in A} w_i y_i}{\sum_{i \in A} w_i} \right\rfloor$$

This is honest about its own meaning: a transparent, difficulty-weighted ratio of passed verified-understanding events. It is easy to explain, monotone in successful evidence, and cheap to roll up. It is not yet a calibrated probability, a validated measure of ability, or a substitute for review. The policy modes are different uses of the same evidence: `Casual` (practice; the attempt is kept in personal history but does not count toward the score and sets no commit status), `Tracking` (record, no gate), `Blocking` (a failure sets a failed Ninchi status that gates merge

where branch protection requires the check), and `Strict` (Pro; mechanically identical to Blocking today, with low-confidence routing to human review — in development — reserved for it). Anti-cheat is a separate set of Pro toggles, not part of Strict: a copy-paste lockdown, and a suspicious-answer flag that runs as a soft signal. Difficulty varies per challenge, and documentation-only changes are auto-passed. §8 describes how this transparent ratio becomes a calibrated, uncertainty-aware model without losing the transparency that makes it trustworthy.

8 The Mathematical Trust Model (Proposed)

The Ninchi Score of §7 is a difficulty-weighted coverage ratio over a declared aggregation scope — an actor, a repository, an organization, or the cross-org rollup that backs a developer's public profile. It is always that: a coverage ratio over passed evidence within a named scope. It is never a latent global rating of a person's ability. That limit is not modesty; it follows from the structure of the evidence, and it sets the shape of everything the model is allowed to claim.

8.1 What the evidence graph can and cannot identify

Write the data as a bipartite *response graph* G over developers U and items I, with an edge whenever a developer answered an item. Competitors imply they sell a single latent comprehension ability per developer, comparable across people, recovered from their pull-request answers. Model that as a generalized linear fit with one ability parameter per developer and one difficulty parameter per item: for an answered edge, logit $P(Y_{ui} = 1) = \theta_u - \beta_i$. The design matrix X then has

$$\operatorname{rank}(X) = |U| + |I| - c(G),$$

where $c(G)$ is the number of connected components of G, leaving a $c(G)$-dimensional family of indistinguishable solutions. Each component carries one free additive constant: shift every ability and every difficulty inside it by the same amount and not a single predicted outcome changes.

Pull-request items make this obstruction maximal. Each item is generated from a unique diff and answered by exactly one developer, so G is a union of near-disconnected *stars* — one developer at the center of their own private items, sharing nothing with anyone else. With $c(G)$ near the number of developers, there is one free ability scale *per developer, anchored to nothing*. A single pass is explained equally well by "this developer has high ability" and "this item was easy"; the two are not separable from the answer alone. A cross-developer comprehension ranking is therefore *not identified* from unanchored pull-request items. No estimator recovers it, because the information needed to separate ability from difficulty was never in the graph.

This is why Ninchi's headline metric is a coverage functional of observed evidence and not a latent ranking of people. The functional is identified by construction: it reads off the graph that exists. A latent ability layer is not foreclosed, but it has to be *earned* by changing the graph: shared items that several developers answer, periodic anchor items on a common scale, and validation against human adjudication. The defensibility is in that discipline, not in any claim that Ninchi computes latent ability from today's pull requests.

8.2 From counts to evidence

The proposed extension treats each verified-understanding event as probabilistic evidence about a scoped pass rate rather than as one weighted entry in a ratio. It preserves the atom-level judgments behind each verdict — whether an answer carried the specific claims a rubric required — so that every estimate remains explainable, disputable, and reviewable. Every estimate it produces is scoped and uncertainty-bearing, defined against an explicitly named challenge distribution: a score of 84 is "84 on this scoped distribution, under this model version, with this interval," never a context-free rating of a person.

The point of the probabilistic framing is uncertainty. Two passed challenges and one hundred passed challenges can yield the same ratio while carrying very different confidence. Model the passes in a scope as exchangeable draws and put a uniform $\mathrm{Beta}(1, 1)$ prior on the underlying pass rate p; the posterior is again Beta, and a 95%

equal-tailed credible interval is read straight off its quantiles. Because the interval lives in $[0, 1]$, it stays honest at the perfect records ($a = n$) that dominate early use, where a normal approximation would run off the end of the scale.

A Beta posterior over a scoped pass rate. With a passes out of n qualified challenges in a scope and a uniform $\mathrm{Beta}(1, 1)$ prior, the posterior is $\mathrm{Beta}(1 + a, 1 + n - a)$, with a 95% credible interval from its quantiles.

- *2 of 2* $\to \mathrm{Beta}(3, 1)$: posterior mean 0.750, 95% interval $[0.292, 0.992]$ — width ≈ 0.70.
- *95 of 100* $\to \mathrm{Beta}(96, 6)$: posterior mean 0.941, 95% interval $[0.888, 0.978]$ — width ≈ 0.09.
- *8 of 10* $\to \mathrm{Beta}(9, 3)$: posterior mean 0.750, 95% interval $[0.482, 0.940]$ — the same point estimate as 2 of 2, with far less spread.

Both perfect records score high; only the dense one has earned the confidence. A short history widens the interval without lowering the score. The uncertainty is epistemic: it reports thin evidence, and does not penalize the developer.

This makes the rule precise: a score from two passed challenges does not carry the confidence of a score from one hundred, and the point estimate alone discards exactly that distinction. The shipped Ninchi Score remains the difficulty-weighted coverage ratio of §7; the posterior uncertainty intervals, and the confidence-based routing they would inform, are in development as a layer on top of that ratio, reporting each score together with how much evidence stands behind it.

Three disciplines keep the extension honest. Its difficulty and discrimination parameters are not assumed by fiat; they require anchor items, feature-based estimation, and empirical checks against human adjudication. Criticality drives policy — escalation, review, gating thresholds — never the inference that a person understands more or less. And the model earns trust only to the extent that §10′s validation program shows calibration, acceptable false-pass and false-fail rates, stable performance across slices, and predictive value. The full development — the identifiability theorem and its proof, the Bayesian machinery, and the conditions under which each estimand is identified — lives in Ninchi's technical companion, available on request.

9 The Recorded Evidence

Each event is persisted as an attributable scored event record: artifact diff snapshot, question, hidden rubric, answer, evaluation, threshold, and timestamp. The integrity model is a familiar one. Append-only Merkle-tree logs underpin Certificate Transparency for TLS certificates [8], and provenance frameworks attest how a build artifact was produced [9]; here the same construction attests human *understanding* rather than a certificate or a build. Tamper-evident hash-chaining, now in development, will let a later auditor verify that no record was altered.

Documentation shows what was *submitted*. The recorded event shows what was *understood*.

The recorded history is the institutional memory that ordinary systems of record lack. One event can serve different audiences through different projections: a private learning signal, a manager's aggregate, a compliance audit payload. Over time the accumulation becomes a practical asset: a persisted history of artifact, question, rubric, answer, and evaluation, available for inspection today, that no single model can natively reproduce. That record does not prove cognition, and it does not guarantee the work was good. What it gives a reviewer is something concrete to inspect later: who was challenged, what they were asked, how they answered, and how the stored rubric, difficulty, and threshold produced the score. That is the account a post-incident review needs.

10 Calibration and Validation

A model that emits probabilities owes calibration. This section describes the validation program as a target, not as something running today. The standard is simple to state: a grader is well-calibrated if humans pass about $X\%$ of the challenges it scores with about $X\%$ confidence — the proper-scoring discipline that goes back to Brier [10] and that the modern literature on confidence calibration formalizes [11]. Here calibration would mean agreement with blinded human adjudication under a stated rubric, not access to a person's private mental state.

IN DEVELOPMENT

The intended program would compare model verdicts against trained human reviewers on a gold-standard corpus and report proper, asymmetric metrics, not a single accuracy number:

- *Brier score* and *expected calibration error*: does a predicted 0.8 correspond to an observed 80% under human adjudication?
- *False-pass rate*, model passes humans would fail, the key risk metric for blocking and compliance use. *False-fail rate*, model fails humans would pass, the key trust metric for user acceptance.
- *Slice metrics*: every figure reported by difficulty, tag, language, and answer length, because aggregate accuracy hides systematic failure, including the fairness failures (penalizing non-native speakers, say) that discredited earlier tools.

Sampled events would be reviewed for question validity, rubric validity, answer adjudication, and evaluator-confidence calibration; events that failed any check would be excluded from automated gating and used to improve the generator and evaluator. In Blocking and Strict modes, automated verdicts would be used only on slices where the current evaluator version had met stated false-pass and false-fail thresholds; otherwise the event would route to human review.

Once it runs, calibration lets the grader of §4 be trusted in proportion to its measured reliability, no more and no less. Evaluator-version tracking, in development, will keep historical scores interpretable as the model improves.

11 The Claim Ladder

Ninchi's credibility depends on never making a claim its evidence cannot support; we state the ladder openly and climb it only as evidence accumulates.

Level	Claim	Evidence required
Operational	A person answered a generated challenge tied to an artifact.	Event logs + identity binding
Scoring	Ninchi computes a difficulty-weighted evidence ratio.	Implemented formula + stored events
Calibration	Model verdicts approximate human adjudication.	Human-reviewed validation corpus
Prediction	Scores predict future demonstrated understanding.	Longitudinal validation
Governance	Ninchi reduces unmanaged AI-work risk.	Customer outcomes + audit studies

Today Ninchi stands on the first two rungs; the model of §8 and the program of §10 are how it earns the rest. The shape of the ladder is deliberate: it mirrors an argument-based approach to validity, in which a claim is warranted only by an explicit chain of inferences (operational, then scoring, then calibration, then prediction), each separately

supported rather than assumed [12]. We borrow the discipline, not the verdict: Ninchi is not, today, a validated psychometric instrument, and nothing in this ladder asserts that it is.

> Ninchi records evidence of demonstrated understanding under a defined challenge — an operational evidence layer that supports review, instruction, governance, and audit without replacing human judgment, testing, grading, or compliance review. The claim boundary is precise: Ninchi produces accountable evidence of demonstrated understanding, tied to a real artifact and a recorded challenge; stricter modes raise the cost and auditability of gaming a challenge rather than eliminating it.

12 Organizational Memory

Accumulated over time, verified-understanding events become an organization's memory: a record of demonstrated understanding organized by people, repositories, time, and the concepts attached to evaluated work. A leader can open an org overview, follow trends over time, and read score rollups by member and by repository. Drilling into individual challenge events and tag-level pass rates across evaluated attempts shows where understanding has built up, which repositories are thin on evaluated coverage, and what concepts keep recurring in passes and failures.

Undocumented understanding is an operational risk. When knowledge lives only in habit, tenure, or reputation, an organization cannot inspect, govern, or account for it after a critical decision. The current views expose concentration signals rather than a named concentration model — uneven member evidence, sparse repository evidence, and thin tag evidence a leader still has to read and weigh. What remains, and what is in development now, is to turn those signals into a named concentration and bus-factor model with key-person alerts, a queryable expertise index ("who understands Billing?"), and departure and expertise-decay analysis.

The claim is deliberately narrow. Ninchi records where accountable humans have demonstrated understanding of specific work, at a specific time, through a specific evaluated challenge, with stored difficulty, threshold, score, and pass/fail result. That record is a memory the organization can inspect, rather than a reputation it has to take on faith.

13 Cognitive Infrastructure

Regulatory frameworks increasingly demand documented human oversight of AI systems. The EU AI Act's Article 14 obliges the humans overseeing an AI system to "properly understand" and "correctly interpret" its output and remain aware of automation bias, NIST's AI RMF asks (GOVERN 2.1, 3.2) that roles and oversight of human–AI configurations be documented, and ISO/IEC 42001 makes such a management system certifiable [13], [14], [15]. Each asks, at bottom, whether an accountable human understood what the AI helped produce. Ninchi aligns with that demand by producing evidence; it does not, by itself, satisfy any regime. Practitioners reach the same place from the other direction, naming the liability of unreviewed AI output *comprehension debt* (a counterpart to the *cognitive debt* named in a small, not-yet-peer-reviewed 2025 MIT Media Lab preprint [16]) and warning that "the AI wrote it and we didn't fully review it" will not survive a post-incident review [17].

The word *infrastructure* is meant precisely, by analogy to layers engineering already takes for granted. Git became the system of record for *what* the code is; observability platforms became the telemetry layer for *how* it behaves; each settled from a tool into a foundational layer other tools assume. Verified-understanding events occupy a layer neither covers: a record of *who understood what, and when*, at the moment a change entered the system. Ninchi documents that layer over its org-analytics substrate; the work in development is to make those records tamper-evident, calibrated against human adjudication, and predictive of future demonstrated understanding. The market needs an infrastructure layer for verified human understanding [18]. Ninchi could compete inside an

existing category, as a better code reviewer or another detector, and lose: those tools improve a workflow, but none leaves behind a record other tools can build on. So Ninchi names and defines the category, "verified human understanding," and builds the system of record meant to become its default infrastructure. A feature can be copied next quarter; a company that defines a category and holds its record is harder to displace.

As AI drives the cost of production down, the bottleneck shifts from producing work to accounting for it. The work gets cheaper; what stays scarce is verified human understanding of it, at the moment it enters a codebase, a transcript, or a system of record. Recording and preserving the evidence of that understanding, honestly and auditably, is the category Ninchi is building.

14 Appendix: References

[1] L. Rozenblit and F. Keil, "The misunderstood limits of folk science: an illusion of explanatory depth," *Cognitive Science (vol. 26, no. 5)*, 2002, [Online]. Available: https://doi.org/10.1207/s15516709cog2605_1

[2] K. Bisra, Q. Liu, J. C. Nesbit, F. Salimi, and P. H. Winne, "Inducing Self-Explanation: a Meta-Analysis," *Educational Psychology Review (vol. 30, no. 3)*, 2018, [Online]. Available: https://doi.org/10.1007/s10648-018-9434-x

[3] L. Zheng, W.-L. Chiang, Y. Sheng, and et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*, 2023. [Online]. Available: https://arxiv.org/abs/2306.05685

[4] A. Bavaresco, R. Bernardi, and et al., "LLMs instead of Human Judges? A Large-Scale Empirical Study across 20 NLP Evaluation Tasks," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Vol. 2: Short Papers)*, 2025. [Online]. Available: https://aclanthology.org/2025.acl-short.20/

[5] P. Wang, L. Li, and et al., "Large Language Models are not Fair Evaluators," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*, 2024. [Online]. Available: https://aclanthology.org/2024.acl-long.511/

[6] R. Parasuraman and D. H. Manzey, "Complacency and Bias in Human Use of Automation: An Attentional Integration," *Human Factors (vol. 52, no. 3)*, 2010, [Online]. Available: https://doi.org/10.1177/0018720810376055

[7] L. Bainbridge, "Ironies of Automation," *Automatica (vol. 19, no. 6)*, 1983.

[8] B. Laurie, E. Messeri, and R. Stradling, "Certificate Transparency Version 2.0 (RFC 9162)," IETF, 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9162

[9] Open Source Security Foundation (OpenSSF), "Supply-chain Levels for Software Artifacts (SLSA)." [Online]. Available: https://slsa.dev/

[10] G. W. Brier, "Verification of Forecasts Expressed in Terms of Probability," *Monthly Weather Review (vol. 78, no. 1)*, 1950, [Online]. Available: https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml

[11] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML 2017), PMLR 70*, 2017. [Online]. Available: https://proceedings.mlr.press/v70/guo17a.html

[12] M. T. Kane, "Validating the Interpretations and Uses of Test Scores," *Journal of Educational Measurement (vol. 50, no. 1)*, 2013, [Online]. Available: https://doi.org/10.1111/jedm.12000

[13] European Union, *Regulation (EU) 2024/1689 (EU AI Act), Article 14 — Human oversight*. 2024. [Online]. Available: https://artificialintelligenceact.eu/article/14/

[14] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1," 2023. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-1

[15] ISO/IEC, "ISO/IEC 42001:2023 — Information technology, Artificial intelligence, Management system," 2023. [Online]. Available: https://www.iso.org/standard/42001

[16] N. Kosmyna and et al., "Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task," 2025, *MIT Media Lab (arXiv:2506.08872)*. [Online]. Available: https://arxiv.org/abs/2506.08872

[17] A. Osmani, "Comprehension Debt: The Hidden Cost of AI-Generated Code." [Online]. Available: https://www.oreilly.com/radar/comprehension-debt-the-hidden-cost-of-ai-generated-code/

[18] A. Ramadan, D. Peterson, C. Lochhead, and K. Maney, *Play Bigger: How Pirates, Dreamers, and Innovators Create and Dominate Markets*. HarperBusiness, 2016.