Foundry: Proof-Carrying Agent Construction

Working technical report - public attachment edition, 2026-10-30. The supplied manuscript dates to 2026-10-28; this edition incorporates an October 30 receipt review and original-author clarification. Historical results retain their dates and scope. "Certified" means internal project review: the builder built, tested and self-reviewed, then an independent auditor who did not build it reviewed the work. It does not mean scholarly peer review or endorsement by Meta, Cerebral Valley or any organizer. Evidence labels describe the cited project records, not a new replication of every result for this edition. Private references and abbreviated hashes describe sources; they are not public reproduction links or complete artifact identities. "Factory" remains a working metaphor. Public omissions are marked.

Building Agents That Ship With Their Evidence

S. Luc Misfit Circuit, Columbia, SC, USA *stephen@misfitcircuit.com*

Manuscript prepared with the Composer crew. Evidence classes are labeled throughout in the author's house style: PROVEN (reproduced by an independent checker), REPORTED (attested, not independently reproduced), and UNMEASURED (believed, not yet instrumented). Component systems are cited by their certification records; where a component's status is PASS-WITH-FINDINGS, the findings travel with the claim.

October 30 update: components, live calls and the unclosed loop

The original author confirms that no agent in service has submitted an exchange through Capture's webhook. Certification probes exercised its judge; those tests are not an in-service agent firing. The full service-to-Capture-to-regression loop has run only as a local reference integration on two preserved failures. Understudy's channels in/out remain planned. The certified Astra candidate does not establish completion of the separate Composer two-builder comparison, which was blocked after two Starts and two timeouts.

The Scoring Layer's 146/146 result belongs to its certified candidate: its gold pack has not run, canonical promotion remains pending, and Capture still borrows that grader. The later Claim Checker gate round is a separate lineage and is **NOTCERTIFIED**; passing tuning inputs does not certify raw-text detection. An active judge can confidently return a wrong PASS. Fail-closed handling of an absent or malformed judge does not bound that error; gold-set evaluation remains pending (§9.3).

A newer installed copywriter prototype demonstrates a job-shaped agent assembled from instructions and method material, not model-weight training. The earlier v0 trial’s three bounded live Meta API requests belong to its writer/scorer project, not Capture. The record retains a successful writer, a failed writer and a scorer evaluation voided by a builder bug; scorer calibration is NOT_RUN. Costs cited for this prototype are rate-computed estimates, not verified billing. This is not evidence of a continuously self-improving service or measured outcome improvement.

Sources: original-author clarification and internal current-receipt review, October 30; copywriter installation/independent closure and aggregate later Claim Checker checkpoint. These updates do not re-certify the entire suite.

Abstract

Agent builders are now common; agents whose competence is *evidenced* are not. We present **Foundry**, a suite for constructing small, job-shaped agents that ship with their proof. A non-programmer describes a job in plain words to the suite's front door (**Understudy**), which interviews them, proposes a plan for explicit approval, builds the agent, and subjects it to a "test & prove" examination graded by the suite's deterministic-first grading engine (**Scoring Layer**). For future connected in-service agents, the design calls for exchanges to be kept as hash-sealed receipts by the suite's capture store (**Capture**), whose design minimizes what is kept: facts and pointers circulate freely, the words themselves are retained briefly, and only *failures* are promoted - redacted - into a permanent, hash-chained case tier. A historical auditor (**Excavator**) grades sealed records at scale (56,474 verdicts over 3,322 records in its certified run, every verdict class labeled, including the honest one: UNVERIFIABLE), and promoted failures are *compiled* into regression cases that the agent must subsequently pass - demonstrating in a local reference integration how a preserved in-service fault can become a build-time test; a continuously operating live loop remains pending. Approvals and cross-agent handoffs are carried by the suite's hub, which is pull-only by construction and names any wake it performs. A claim-support grader checks agent utterances against an approved fact sheet, with a strict precedence: deterministic mechanism first, an explicitly labeled JUDGE seat only for the residue the mechanism defers, and fail-closed hardening whenever the judge is silent. We report each component's certification status, the loop's reference integration (in which two preserved in-service failures were compiled into cases that the recorded agent fails, the current agent passes, and the original fault can no longer be replayed), and the suite's limitations at the same prominence as its results - chief among them that no live agent has yet fired at the capture door, and that the front door's runtime is, by its own README's admission, deterministic parts and bounded matching rather than general language understanding.

Index Terms - agent construction, evaluation, proof-carrying systems, claim support, audit trails, hash chains, human approval.

1. Introduction

The market's agent builders optimize for the moment of creation: describe a bot, receive a bot. The operator's problem begins the moment after - *what is this agent allowed to claim, and how would anyone know if it worked?* For the suite's intended user - an everyday person or a small business describing a job in plain words, at a kitchen table - the question is not rhetorical. The agent will answer customers. Its errors will be believed by people who trusted the owner's name. An agent builder that cannot answer "show me it works" is selling the *shape* of help.

Foundry's thesis is that construction and evidence belong in one process. The intended lifecycle below is not a claim that the complete live-service loop is operating:

1. **Describe.** The job is stated in the owner's words and kept intact; builders downstream work from the exact request, not a relay's paraphrase.
2. **Prove at build.** Before the agent serves anyone, it sits an examination built from the job itself, graded mechanism-first.
3. **Keep the receipt in service.** Every exchange is captured under minimization rules; failures - and only failures - are kept permanently, redacted, and sealed into a chain.
4. **Compile the failure.** A kept failure becomes a test. The loop closes when the agent passes the test its own history wrote.

The suite's name is the operator's coinage and its argument: a foundry takes raw material and casts a made thing, and what comes out of a foundry is expected to bear weight. The components are independently useful and independently certified; the suite is the loop that connects them.

Contributions. (i) A construction front door whose deliverable includes an examination record and a downloadable evidence chain, not merely a running agent (§3). (ii) A deterministic-first claim-support grader with a labeled, non-blended judge residue and fail-closed semantics (§4). (iii) A capture architecture built on minimization - grade facts over everything, promote on fail only - with four narrow, separately shaped doors (§5). (iv) A historical auditor that grades sealed past records under an honesty-preserving verdict taxonomy (§6). (v) A compiler that turns sealed failures into regression cases, with a hard refusal - no_verbatim_material - when the words a case requires were not kept (§7). (vi) The loop's reference integration and a limitations section we intend to be citable against us (§8, §9).

2. Background and Related Work

The closest intellectual ancestor is *proof-carrying code* (Necula), in which a program ships with a proof that a consumer can check independently of the producer. Foundry pursues the analogous property for agents - *proof-carrying agents*: the artifact arrives with evidence a skeptic can verify without trusting the builder. The analogy is imperfect in an instructive way: code proofs are decidable, whereas agent evidence is graded, sampled, and partly judgment - which is why Foundry labels evidence classes rather than pretending to proof in the mathematical sense. From evaluation practice the suite takes the discipline of deterministic-first grading (mechanism before models, because a mechanism's verdict can be re-derived by anyone), and from forensic practice it takes chain-of-custody thinking: sealed records, hash chains, and access shaped by need. The suite's sibling system, **Composer** - the agent-oriented workflow and coordination system in which Foundry's components were themselves built and audited - supplies the governance culture (independent certification, fresh-session auditors, receipts over narratives) that Foundry's components inherit; the two systems are companions, the factory and the foundry, and the reader is asked to forgive the operator his industrial period.

3. Understudy: The Front Door

Understudy is where an everyday person meets the suite. Its premise is in its name: an understudy studies the role-holder until it can perform the job - and, in this theatre, it must *audition*. The shipped build is a local web application with a deliberately small, legible flow:

1. **Describe the job** in plain words.
2. **Choose the supported scope** explicitly (in the current build, a local message-helper); the application states its limits rather than implying broader powers.
3. **Interview.** The application asks the questions the job's performance will turn on.
4. **Approve the plan.** Nothing builds until the owner approves what will be built - approval is a step in the flow, not a settings page.
5. **Test & prove.** The built agent sits an examination; the owner sees per-area marks with the exact messages, replies, and checks behind them.
6. **Review the record / download the chain.** The build's evidence is a first-class output, exportable as a ZIP.

The application's honesty about itself is a feature we wish to underscore, because it is the suite's thesis at the smallest scale: by its own README, *no message is delivered outside the page; no calendar or payment*

actions are performed; external schedule and customer-channel integrations are not implemented; and the runtime is deterministic parts plus bounded matching, not general language understanding. Demo data is an invented salon or the owner's own supplied information. A builder that announces its ceiling invites trust in its floor.

Status. The build (produced by an external builder seat against a sealed contract, under the factory's two-builders-one-contract method) is CERTIFIED PASS-WITH-FINDINGS, all findings LOW or INFORMATIONAL; the certifying figures include a full independent re-run of the frozen examination harness - 123 of 123 checks - and the harness itself is content-frozen by hash so that "the exam changed" is never an available explanation for a changed result. The application is installed on the operator's own machine shelf, where its surface was proven serving in place (PROVEN: HTTP 200, live state endpoint, clean shutdown) - these checks establish serving/state behavior, not by themselves a complete rendered browser journey. Its gap list to full usability is known and enumerated: a real channel in, a way out gated by the owner's approval tap, a model seat behind the interview and drafting (with the deterministic graders as police), and the owner's own sealed fact sheet. None of these is hidden; all are sequenced.

4. The Scoring Layer: Grading at Build

If Understudy is the door, the Scoring Layer is the examiner - and it is a deliberately unglamorous one. Its core is **deterministic**: checks that any skeptic can re-run and re-derive, because a grade that cannot be reproduced is an opinion wearing a number.

Its distinctive component is the **claim-support grader**. An agent's reply is decomposed into claims; each claim is checked against the sources it cites - in the Foundry setting, the agent's approved fact sheet, the single document the agent is permitted to speak from. The grader's verdict classes are the paper's recurring characters:

- **SUPPORTED** - the cited material carries the claim (mechanism-decidable matches);
- **CONTRADICTED** - the cited material says otherwise (mechanism-decidable where the contradiction is structural, e.g. aligned quantities that disagree);
- **UNSUPPORTED with** needs_judge - beyond the mechanism: a paraphrase to recognize, a topic the sheet is silent on. The mechanism *defers*, and says so.

Deferral is where most grading systems quietly become vibes. Foundry's rule, inherited from the operator's eval-hackathon work, is that **judge residue is labeled JUDGE and never blended into deterministic totals** - the report card shows mechanism verdicts and judge verdicts as separate rows, forever. And until a judge seat is wired, a deferral *hardens*: it grades as a failure with its reason (needs_judge) recorded verbatim. This is intentionally strict - a true paraphrase fails alongside a true invention - and the strictness is a *product decision*, documented as such, because the alternative (an unaccountable pass) is the one failure mode the suite exists to prevent. A judge seat has since been wired under exactly these rules (§5.3): it speaks only to deferrals, its verdicts live in their own row, and its silence restores the hardening.

Status. The deterministic core and claim grader are CERTIFIED (PASS-WITH-FINDINGS) with a 146/146 suite at certification; one residual LOW is carried openly - nested parentheses can leave a () residue in claim extraction, fail-closed in effect - and the gold-standard task pack against which future judge seats will be tuned is queued work, named as such rather than implied done.

5. Capture: Keeping the Receipt

5.1 The design problem

In-service grading faces a dilemma the operator articulated while designing this component: to turn a real failure into a regression test, you need the *verbatim exchange* - the words in and the words out. But a store of verbatim exchanges is, in the operator's own catch, **a database of user information**, with everything that implies. Capture is the suite's answer, and its principles were dictated before its schema:

- **Capture is keeping the receipt.** The verbatim exchange is recorded and hash-sealed at capture; verdicts elsewhere *point* to records (record id, sequence, hash) rather than copying them.
- **Minimize by construction.** Grade facts over everything; retain words briefly; promote on fail only.
- **The tap depends on the agent.** In the operator's doctrine, capture is the *seat* and taps are *doors*: the receiving end is owned (local, or on the owner's tailnet), the sender holds its copy until the receiver acknowledges, and capture fails loudly - never silent holes.

5.2 Two tiers, four doors

The store (SQLite in the testing build; boring standard SQL throughout, so the named upgrade door - Postgres/Supabase, when multi-machine writes or real users arrive - is a carry, not a rebuild) keeps two tiers:

- The **window tier** holds full exchanges and *expires on a clock*; every purge is logged.
- The **case tier** holds promoted failures only - redacted at promotion - sealed into a **hash chain** from a genesis record. Purge never touches it; tampering with a sealed case is detectable by chain verification (demonstrated by the certifier, who altered one byte of a sealed case and watched verification name the exact case).

Access is through **four narrow doors**, each shaped to its need: *write-in* (the receiver; per-agent secret, hash-only in storage; acknowledgment only after the write commits - an unacknowledged capture does not exist, and the sender knows it), *mine* (an index of facts, pointers, and hashes - **no words**), *pull-on-fail* (the only door through which words leave, one receipt at a time, by pointer, on a FAIL), and *report-out* (the tracking UI: agents, verdicts, sealed cases with chain status, and the report card with its AT BUILD and IN SERVICE halves). Creating an agent in the UI mints its capture door - webhook URL, per-agent secret, fact-sheet slot - because in this suite an agent without a receipt path is not finished.

5.3 The judge, wired

The miner that walks the window tier applies the deterministic check and the claim grader of §4; FAILs promote immediately into the case tier. With the judge seat wired (Meta Muse Spark, reached through the suite's provider machinery), deferrals are decided as §4 prescribes: the judge sees the fact sheet and the deferred claims - with the reply and claims explicitly framed as *data, never instructions* - and returns per-claim support with a stated reason. Its verdicts are kind JUDGE in storage and in every total. The certified live evidence is small and exact: a paraphrase ("A cut is $40," against a sheet reading "A haircut costs $40") received a judge PASS with the sheet cited; two inventions (a discount, a walk-in policy the sheet never mentions) received judge FAILs and were promoted; verbatim exchanges never consulted the judge at all; and when the certifier disabled and then sabotaged the judge mid-run, the remaining deferrals hardened fail-closed, labeled, and the run completed. An independent certifier's own live probe of the route returned the recorded model identity from the provider's response - the route is real, not artifact theater (PROVEN: delta certification, all findings INFORMATIONAL).

5.4 Status

Capture is CERTIFIED PASS-WITH-FINDINGS and its one product-level finding - the strictness adjudicated in §4 - is CLOSED by the judge wiring and its delta certification. Its remaining unproven list is stated in §9 without varnish; the largest item is that no live agent has yet fired at its webhook. The system is a tested, sealed, empty theatre, awaiting its first real performance.

6. Excavator: Auditing the Past

Excavator grades history. Where Capture watches the present, Excavator consumes *sealed records of past agent sessions* - collected under a facts-only discipline - and issues verdicts over them at a scale the live loop cannot: in its certified canonical run, **56,474 verdicts over 3,322 records**, distributed PASS 129 · FAIL 12 · UNVERIFIABLE 46,364 · NOT_APPLICABLE 9,969, with the underlying chain of 9,972 records verifying intact (PROVEN: certification; test suite 72/72 across two machines).

The taxonomy deserves a paragraph, because it is the component's thesis. A naive auditor forces every record into pass or fail and manufactures confidence; Excavator's largest class is UNVERIFIABLE - records whose evidence cannot carry a verdict, labeled as such rather than resolved by vibes. The 12 FAILs it *can* stand behind are correspondingly worth something: each is a fact about the past with a sealed provenance. On the operator's Mac, a collector runs under launchd producing facts-only sealed bundles for Excavator's consumption; one boundary is recorded here because the suite is prouder of it than of any feature - the collector has an access-control boundary [public-edition omission: private household directory details], and the system's response was to *name the wall in its own needs list* rather than route around it. An audit tool that respects the fences it cannot cross is the only kind worth deploying in a home.

7. The Compiler and the Hub: Closing the Loop

7.1 Failures, compiled

A promoted failure is evidence; a *compiled* failure is an instrument. The suite's compiler takes a sealed FAIL - verdict plus the verbatim exchange pulled through the pull-on-fail door - and emits an eval case in the loop's standard shape: the original input, the recorded (failing) reply as the negative fixture, and the grading specification. The case then joins the agent's examination set. The compiler's most important behavior is a refusal: when the corpus holds only facts about an exchange and not its words - as with Excavator's facts-only verdicts over history - compilation fails with no_verbatim_material and writes nothing. A compiler that invents the exchange it needed would poison the suite's entire evidentiary claim; the refusal is the feature, and it is precisely the gap Capture (§5) exists to close for the future.

7.2 The hub: approvals and handoffs

The suite's hub - the Together Hub, built for the Meta Global AI Developer Hackathon and independently reviewed within this project - carries the loop's human and inter-agent traffic: invitations issued through a key desk (single-use, expiring), a dispatcher seat, cross-family handoff probes, an **approval gate** at which a human's decision is a first-class event in the record, and a receipts view in which every stage of a handoff can be inspected after the fact. One doctrine governs the hub and is stated wherever it is described: it is **pull-only** - it never claims to have woken an agent, and any handoff that conceals a wake must name the wake. In a suite whose currency is honest records, a hub that exaggerated its own agency would be counterfeit at the center. (Certification: smoke 105/105, suite 471/471, typecheck PASS, independent audit PASS-WITH-FINDINGS.)

8. Evaluation: The Reference Integration

The components were integrated in a reference run - Understudy × Excavator × Scoring, with Capture's predecessor gap deliberately left visible - and the integrated build was independently certified PASS-WITH-FINDINGS. The run's evidentiary centerpiece uses two *preserved* in-service failures from Excavator's corpus (a fault the suite calls *boundary-discount*: the agent conceding terms at the boundary of its instructions), preserved in the sealed corpus [public-edition omission: exact case pointers]:

1. The bridge compiled both verdicts into graded cases - honestly compiling **zero** cases from the rest of the facts-only corpus, with no_verbatim_material recorded, exactly as §7.1 requires.
2. Against the compiled cases: the **recorded reply FAILs** (the historical agent is convicted by its own words), the **current agent PASSes**, and a **replay of the original fault is blocked by the guard** the case installed.

That triple - history fails, present passes, fault cannot recur silently - is the loop's existence proof in miniature: an in-service failure, carried by sealed evidence, became a build-time test with teeth. Suite-level figures for the record:

Component	Status	Headline evidence
Understudy	Internally certified PASS-WITH-FINDINGS (Astra candidate)	Harness 123/123 on independent re-run; serving/state checks on operator machine; separate Composer comparison blocked
Scoring Layer	Certified PASS-WITH-FINDINGS	Suite 146/146; deterministic + claim grading; R1 LOW carried
Excavator	Certified PASS-WITH-FINDINGS	56,474 verdicts / 3,322 records; chain of 9,972 verifies; 72/72 tests, two machines
Capture	Certified PASS-WITH-FINDINGS; judge delta certified, F1 CLOSED	67 tests; live judge evidence incl. an independent probe; one-byte tamper caught by chain
The Loop	Certified PASS-WITH-FINDINGS	§8 triple: recorded FAIL → current PASS → replay blocked
Hub	Finalized + certified	Smoke 105/105; suite 471/471; approval gate + receipts view

In keeping with the house rule, the table reports *deltas and certifications*, not raw enthusiasm: each component's findings are enumerated in its own record, and two findings from the loop's certification (a package-hash recording gap and a non-standalone package) were fixed and delta-audited closed before this paper was written - the v2 loop package carries a 239-entry manifest for precisely that reason.

9. Limitations

1. **The theatre is empty.** No live agent has yet fired at Capture's webhook; nothing in the in-service half has been exercised by a real customer conversation, on a real tailnet, at real scale. The loop's integration proof used preserved records and local processes. We regard this as the suite's largest honest gap, and it is a gap of *occasion*, not of known defect - which is exactly what one says before reality grades the claim.

2. **The front door is narrower than its welcome.** Understudy's current scope is a local message-helper; channels in and out (with the owner's approval tap on the way out) are designed and sequenced, not built; its runtime is deterministic parts and bounded matching. The examination it administers is therefore an examination of a deliberately small performer.
3. **The judge is young.** Judge correctness is evidenced by a handful of live calls, exactly counted in the certification record; prompts and thresholds are untuned against a gold set, and the residue decisions of the whole suite rest, by design, on judge quality where the mechanism defers. Fail-closed handling covers an absent, disabled or malformed judge: unresolved claims harden to failures. It does **not** bound a confidently wrong PASS from an active judge. That can silently admit an unsupported claim. Judge error must be measured against a gold set; that evaluation and calibration remain pending. The handful of live probes is not a correctness guarantee.
4. **Borrowed parts are labeled, not hidden.** The capture miner loads its claim grader from the grading engine's certified candidate tree, because the canonical tree's promotion is a deliberate, pending step; the package therefore fails loudly (ImportError) without its workspace siblings rather than silently grading with something else. Standalone-ness is a known, recorded debt.
5. **Scale is asserted narrowly.** The capture store's concurrency discipline is a single writer by construction; purge is a function awaiting its clock; the multi-machine trigger that would move the store to Postgres is named but unexercised.
6. **One operator.** As with its sibling factory, the suite's bus factor is the author. Its mitigations are the artifacts themselves: sealed records, frozen harnesses, and papers like this one, written so that the evidence outlives the session that produced it.

10. Conclusion

Foundry's claim is modest and, we think, new in its combination: an agent builder whose output is not an agent but an *agent with a record* - examined at build by mechanisms anyone can re-run, receipted in service under minimization rules that treat other people's words as borrowed property, audited against its own sealed past, and improved by a compiler that turns its worst moments into its next examination. The foundry metaphor earns its keep in one final way: castings are inspected before they bear weight, and the ones that fail inspection are not shipped quietly - they are melted down and recast, and the inspection record is kept. The suite's components are built and certified; its loop is proven in miniature; its theater awaits a first real performance. We intend to report that performance with the same labels used here.

References

[1] Luc, S. *Understudy* - build record and certification (PASS-WITH-FINDINGS), frozen harness SHA-256 2821b5b3…; Mac shelf install record RUN-RECORD-UNDERSTUDY-INSTALL-2026-10-28.

[2] Scoring Layer - Session 1 build and E1 certification (PASS-WITH-FINDINGS); candidate SHA-256 fe120e95…; residual R1 carried.

[3] Excavator - certification record (Fable Job 301, PASS-WITH-FINDINGS; nine findings fixed); canonical verdict histogram and chain verification, 2026-10-27.

[4] Capture system - specification (operator-dictated design, 2026-10-28), build report, certification (PASS-WITH-FINDINGS), and judge delta certification (F1 CLOSED). Final package SHA-256 0d93548b…de72.

[5] Understudy × Excavator × Scoring integration ("the Loop") - build and certification records; v2 package SHA-256 56e12af8…, manifest 239 entries.

[6] Together Hub - finalization and audit records (Opus Job 001 PASS-WITH-FINDINGS; Jobs 004/005 confirmations); Meta Global AI Developer Hackathon, Indie Developer track.

[7] Luc, S. *Composer: Architecture and Operation of an Agent-Oriented Workflow and Coordination System.* Companion paper, Misfit Circuit, 2026.

[8] Necula, G. *Proof-Carrying Code.* POPL 1997 - the ancestor acknowledged in §2, with the differences stated there.

Author's note. The author thanks the crew - and in particular the two seats whose preserved failures became §8's examination, and who have been gracious about it. [Public-edition omission: private household scheduling details.] Correspondence to the author; the foundry is warm.