Proof-governed autonomy for live revenue experiments

Alan N. Pham[1,*]

[1]AO Labs; institutional affiliation and coauthor list to be confirmed before submission

*Correspondence: Alan Pham

Autonomous agents can write, route, and execute workflows, but commercial autonomy is hard to evaluate when constraints, buyer signals, and outcomes live in scattered logs. We report Relay, a continuously updated deployment record for a notes-first revenue agent governed by an explicit proof contract. The system converts a broad commercial goal into a state machine with permitted actions, forbidden actions, bottlenecks, proof windows, buyer-intent signals, payment outcomes, and compact research-journal entries. At the latest production read, the seven-day public-intent window contained 21 page views, 11 page-engagement events, 7 sample clicks, and no email-intent clicks, while the active outbound proof remained `paid_test_explicit` at 14/20 sample sends. Replies, unhandled replies, payments, gross revenue, and tracked outbound visits were all zero. Because the money loop was disabled under owner cost control and no fresh active first-touch leads were queued, the current bottleneck is execution/refill capacity, not a demand finding. The contribution is not a claim that Relay makes money. It is a falsifiable protocol for making autonomous revenue experimentation auditable enough that future success, failure, and regressions can be interpreted.

1 Introduction

Autonomous software agents are moving from demonstrations into deployed settings where they can observe, decide, and act through external tools (*1–5*). This creates a new evaluation problem. A tool-using system can appear autonomous because it sends messages, records data, or triggers payments, but a scientific claim of autonomy requires more than action. The system must preserve the state that justified action, the constraints that limited action, and the evidence that should update control.

This requirement is familiar in closed-loop experimentation. In autonomic computing, a controller monitors, analyses, plans, and executes under an explicit knowledge model (*6*). In closed-loop materials discovery, a system chooses the next experiment, records the result, and updates the search path (*7*). Commercial agent deployments need a related discipline, but with different risks: buyer privacy, outreach ethics, payment accounting, internal-test contamination, and human operator intervention. Without an explicit proof contract, a founder can repeatedly change the market, copy, price, or volume and still describe the whole process as one autonomous system.

Relay was built around a deliberately narrow commercial offer: a buyer sends one stuck client email, last message, rough draft, or a few non-private bullets and receives a subject line plus one reply draft. The user-facing product is simple, but the research object is broader. Relay connects a product surface, buyer-intent telemetry, payment links, intake capture, outbound acquisition records, production-event routes, a success governor, an autonomous money loop, and a research-journal endpoint. The system is designed to keep the operator out of routine replanning during a proof window while preserving enough evidence to reconstruct what happened. Paper-backed changes are now treated as part of the evidence system: when the dashboard, backend, report language, source graph, metrics, guardrails, or results change materially, the manuscript and public PDF must be updated in the same work cycle.

Here we frame Relay as a proof-governed autonomy system. The term denotes a closed loop in which the agent can act only under a recorded proof state: a measurable objective, a current bottleneck, a permitted action, a forbidden-action set, a deadline, and outcome fields that define whether the action succeeded. This paper preserves the initial evidence lock and adds dated operating updates instead of replacing weak results with smoother language. The resulting claim is intentionally bounded. Relay is deployed and auditable; it is not yet proven profitable.

2 Results

The Results are organised around four levels of the deployment: the proof-governance protocol, the deployed product surface, the evidence matrix at the lock, and the journaled active experiment that determines the next interpretable claim.

2.1 A proof contract for autonomous revenue experiments

Relay converts an open-ended goal—make the system generate money while the operator stays out of the loop—into a constrained experiment. The proof contract contains five parts. First, a revenue objective defines the target outcome. Second, the state layer records money, payments, queue state, active-sample progress, send capacity, exception state, and recent market signals. Third, the success governor computes the bottleneck and selects the next allowed action. Fourth, the guardrail contract blocks actions that would make the experiment uninterpretable. Fifth, the research journal records the control tick and money-loop tick.

At the evidence lock, the success state was *active_experiment_sample*. The owner mode was *out_of_loop_waiting*. The allowed action was to send 10 active leads during the approved business-hour window. The forbidden set included asking the owner to choose copy, targeting, price, or volume before the proof deadline, increasing volume before buyer signal, changing more than one major variable at a time, and declaring demand failure from an execution miss. This converts an emotionally loaded commercial objective into a narrow experimental state.

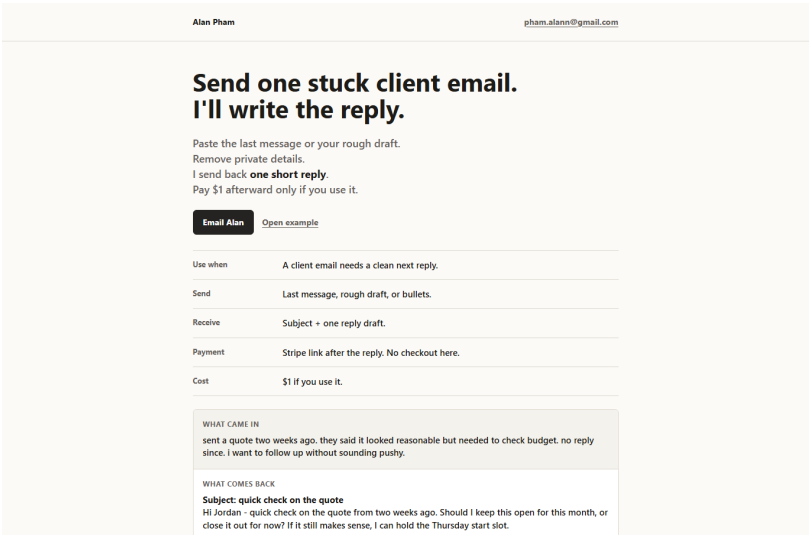
Table 1. Proof contract at the evidence lock. The table separates the current state from the permitted action and the claims that remain unsupported.

Element	Recorded value	Experimental function
Objective	Reach the first paid follow-up email and a USD 1/week minimum-money proof	Defines the target outcome without changing the evidence.
Money state	USD 0.00 gross revenue; zero non-test payments	Prevents internal or unmatched test events from becoming a false success.
Bottleneck	Active hard-paid-test sample incomplete	Blocks premature judgement of demand.
Allowed action	Send 10 active leads inside the approved cap and window	Completes the minimum exposure before interpreting signal.
Forbidden actions	No owner-choice request, no volume increase, no multi-variable change, no failure claim from execution miss	Preserves causal interpretability.
Audit point	2026-05-04T17:00:00-04:00	Defines when the window should be evaluated.

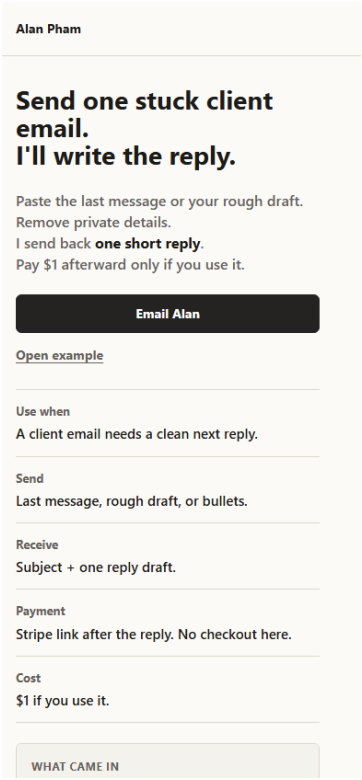
2.2 The product surface is part of the experiment

The buyer-facing surface is intentionally plain. The current `relaybrief.com` page presents a small manual email-reply service by Alan Pham rather than a named software product or AO Labs project route. The first screen now starts from the buyer's smallest likely work object: one client email that feels stuck. The page asks for the last message, rough draft, or bullets with private details removed. It promises a subject line plus one short reply draft and states that payment happens through Stripe afterward only if the buyer uses it. These interface elements are not only conversion elements; they are also sensors. Opening the example output and emailing first create evidence used by the proof contract; payment is measured only after a preview exists.

Figure 1 shows the current production product surface used by the experiment. The desktop page presents only the human-service identity, email-first action, stuck-client-email example, payment-after-use rule, and service facts. The mobile view preserves the same order without adding a separate sales layout, file-download path, upload path, account path, or card-entry form. The product surface is included here because future claims about buyer behaviour depend on what buyers actually saw.



(a) Desktop product page.



(b) Mobile product page.

Figure 1. Buyer-facing `relaybrief.com` surface used by the evidence-locked deployment. The surface is both a product page and an experimental instrument: page views, example opens, email-intent clicks, and payment-after-use preflights are recorded or checked as buyer-intent evidence.

The returned artifact is one reply draft. Figure 2 shows the example output retained as the buyer-visible artifact. This matters because the experimental offer is not a vague artificial-intelligence claim or a broad writing assistant; it is a concrete promise that one stuck client email can be converted into a direct next reply.

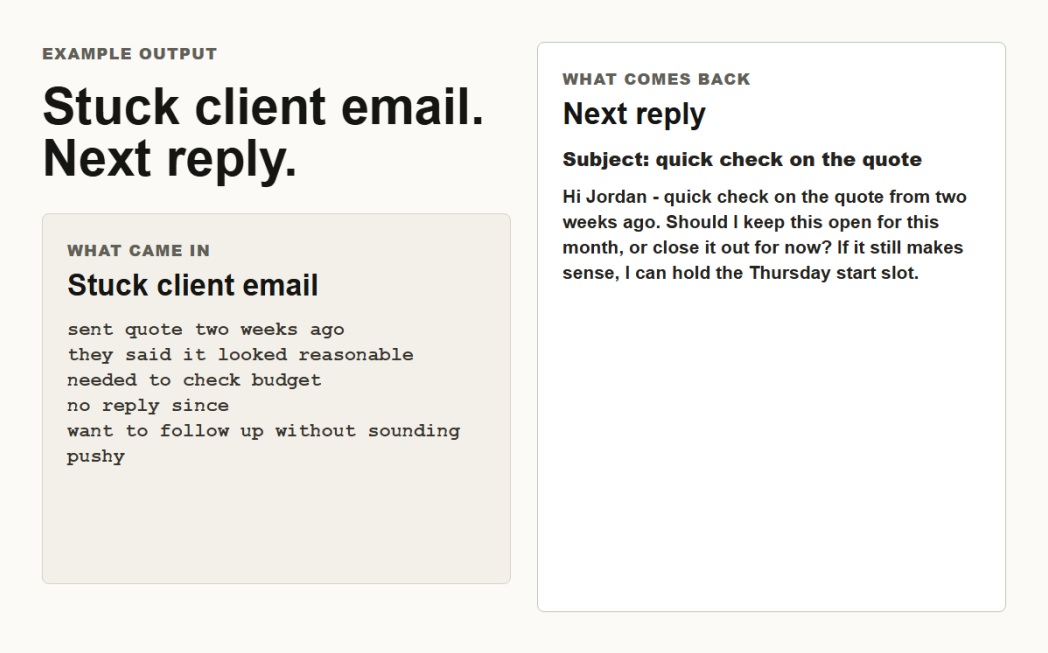


Figure 2. Example output. The revised sample makes the value proposition visible as a before–after transformation: stuck client email context received, one reply draft returned.

2.3 Evidence at the lock supports deployment, not profit

The saved operational snapshot showed a live route surface and a configured backend, but the money state was still zero. Table 2 summarises the supported and unsupported claims. This distinction is the main scientific discipline of the manuscript. A stronger paper does not inflate early signals; it records where the proof stops.

2.4 The active experiment defines the next interpretable result

At the evidence lock, the active experiment was a hard paid test. It had 0/10 progress, 10 sends remaining, 45 queued direct leads, 23 active due leads, and a remaining cap of 10. The next autonomous send window was 2026-05-04T09:30:00-04:00, and the proof audit point was 2026-05-04T17:00:00-04:00. Because the snapshot was taken before the send window, zero active sends was an expected pre-window state rather than evidence of market rejection.

The primary endpoint for the next window is non-test paid purchase count and gross revenue. Secondary endpoints are replies, checkout clicks, note submissions, lead captures, send failures, and open exceptions. If the active sample completes and no buyer signal appears, the protocol should rotate one major variable—targeting, copy, price, or volume—rather than rewriting the entire system. If buyer signal appears, the protocol should preserve the lane and close the payment or fulfillment loop.

Table 2. Evidence matrix at the 2026-05-04 lock. Values are descriptive and should not be read as statistical evidence of demand.

Domain	Evidence at lock	Supported claim	Unsupported claim
Route surface	Stripe and Tally webhooks, client gate, production routes, autopilot routes, intent summary, research journal	Relay was a deployed operational system, not only a static page.	Route robustness under buyer load.
Configuration	11 of 11 checked environment requirements present	Payment, intake, delivery, and tracking routes had required configuration.	Correct behaviour for every future edge case.
Intent	56 page views, 8 link clicks, 3 checkout clicks, and 3 lead-capture events	There was measurable site interaction.	Product-market fit or statistically reliable conversion.
Filtered buyer signal	18 real page views, 1 real lead, 2 internal test leads separated	Internal activity was not treated as buyer demand.	A large external buyer sample.
Outbound	60 rolling seven-day sends, 0 replies, 0 payments, 0 send failures	Prior outreach and non-response were visible.	Outbound conversion.
Money	USD 0.00 gross revenue, zero non-test payments	Profit was not yet proven.	Autonomous money-making success.
Active experiment	0/10 active sends, 23 active due leads, 10/day cap	The next proof action was defined.	Demand failure before the sample completes.
Journal	4 compact research-journal entries	Control states were preserved for audit.	Longitudinal generalisation.

2.5 Current operating state

The latest live check was read from the production Relay API at 2026-06-01T12:17Z, or 8:17 AM Eastern. It updates the paper beyond the initial May 4 evidence lock without changing the interpretation standard. The system has more observed intent than the initial snapshot, but it still has no real buyer revenue. The prior `revenue_leak_direct` observation matured with zero replies and zero payments, so the controller rotated into `paid_test_explicit`. The current active sample is still incomplete at 14/20, with six sample sends remaining. The controller reports `active_experiment_refill` because no fresh active first-touch leads are queued, no active-sample send capacity is estimated for the next window, and the money loop is disabled under owner cost control. This is an execution/refill

Table 3. Predeclared interpretation of the next active sample. The goal is not to make a dashboard feel optimistic, but to make the next result interpretable.

Observed outcome	Interpretation	Permitted operation
Paid test or strong buyer reply	The lane has buyer signal.	Fulfil, preserve the lane, and continue only within the current cap.
Checkout clicks without payment	Friction may be in price, trust, or checkout path.	Change one variable and run another small sample.
No replies or clicks after complete sample	Current lane has no signal at this sample size.	Rotate one major variable; do not change targeting, copy, price, and volume simultaneously.
Send-window failure	Execution failed before demand was tested.	Fix execution before drawing market conclusions.
Manual owner intervention	Experiment becomes contaminated.	Mark the window as interrupted and separate it from autonomous evidence.

finding, not buyer-demand evidence.

The buyer-visible object remains the small desk-service path introduced after earlier zero-revenue reads: send one stuck client email, last message, rough draft, or bullets; Alan replies with a subject line and one short reply draft; the buyer pays USD 1 afterward only if they use it. The page leads with the direct human-service claim, “Send one stuck client email. I’ll write the reply”, and keeps no-app, no-upload, no-account, no-password, no-card-form, and no-checkout-before-preview boundaries visible. The active outbound, reply, fallback, and post-interest copy match that path by asking for one stuck client email, last message, rough draft, or a few bullets before any payment link. The product page, sample PDF, telemetry, public-offer preflight, outbound preflight, reply-autoclose preflight, and payment-webhook preflight were live or OK in the current read. The missing result remains a real paid buyer.

Table 4. Current Relay state from the live production API. Values are operational state, not a hiring, market, or profitability probability.

Field	Live value
Seven-day buyer-facing intent	21 page views, 11 page-engagement events, 4 scroll-depth events, 7 sample clicks, 0 email-intent clicks, and 0 lead records in the current ops read.
Revenue	USD 0.00 gross revenue; 0 payments.
Outbound execution	The current read shows 20 rolling seven-day sends, 148 custom-sent acquisition records, 81 direct follow-ups due, 0 replies, 0 unhandled replies, 0 send failures, and 0 tracked outbound visits.
Active experiment	`paid_test_explicit`; 14/20 active-sample sends observed; 6 active sends remaining; no fresh active first-touch leads due; proof deadline 2026-06-01 at 5:00 PM Eastern.
Controller state	`active_experiment_refill`; readiness is `blocked` because the money loop is disabled, the active experiment needs more queued first-touch leads, and there is no estimated active-sample send-window capacity.
Runtime state	Money-loop runtime status is `disabled`; the latest error and wake reason are `paused_by_owner_cost_control`; the live verdict is `money_loop_unhealthy:disabled`.
First-money gates	The USD 1 price, product route, stuck-client-email preview-first page, no-checkout-before-preview rule, telemetry, sample inspection, payment-webhook check, paid-fulfillment check, and reply-window protection are present; the real paid-buyer gate remains missing.
Public offer path	Landing page, sample PDF, personal-service identity, stuck-client-email email-first path, privacy-removal instruction, inline example, payment-after-use rule, lead API, payment-webhook preflight, outbound preflight, and reply-autoclose preflight were verified OK in the current read.

2.6 A send-day accounting fault became part of the evidence

The May 11 operating read exposed an execution-accounting fault rather than a market conclusion. The controller showed `paid_test_explicit` at 10/20 active progress with ten queued active leads and no replies or payments. After UTC midnight, however, the live state also reported `outbound_window_missed` because the daily send counter could reset on the UTC day instead of the configured Eastern send day. The correct interpretation is not that the active lane failed; the sample had not completed, and the post-window miss state could be a time-accounting artefact.

The send-day accounting was revised so daily send counts begin at midnight in the configured send timezone rather than midnight UTC. The active paid-test path was also revised to expose a direct USD 1 first-money checkout, make the checkout-first route the first-screen action, add a source-visible

primary-money-path marker, and preserve a lower-priority `#send-notes` fallback without presenting it as a send-first path. Later revisions made the reply-observation state authoritative across all money-control summaries. The public dashboard was then reduced to a compact revenue ledger: revenue, payments, checkout price, checkout clicks, the required first-payment condition, and a plain no-revenue status line, with raw data collapsed behind disclosure. A subsequent dashboard revision added a single cumulative sample-click line chart so the operator can see buyer inspection accumulating over time without restoring broad metric-card clutter. The next dashboard revision replaced the ledger-first money block with a connected signal path: seven-day views, seven-day sample opens, email-intent clicks, active-sample completion, replies, and revenue. This change keeps zero revenue explicit while showing pre-revenue movement without broad metric-card clutter. The buyer page was then separated from AO Labs suite navigation and rebuilt as a standalone `relaybrief.com` checkout surface with the USD 1 action as the only header action. The example output was then replaced with a before–after buyer proof showing a rough input converted into a larger follow-up artifact. The product surface was then reduced again after repeated trust failures: it presents RelayBrief as a small manual cleanup service operated by Alan Pham, makes email the first action, states the USD 1 price before payment, names Stripe as the after-acceptance payment path, states that no card details, download, login, or password are entered on the site, removes public pre-payment note collection, and retains a secondary Stripe link only for accepted jobs. A later source revision made a short plain-text preview the first value event, asked for USD 1 only if the buyer wanted the finished artifact, and updated the public-offer preflight and outbound copy so cold buyers were not asked to pay before seeing any useful output. The next source revision narrowed the buyer promise again: the visible product, example PDF, public-offer smoke check, and outbound templates sold one follow-up email instead of a multi-part packet. The next revision removed the visible and source-level public checkout action entirely: the public page offered email and an example only, the outbound, legacy fallback, and positive-reply copy asked for a rough note before any payment link, and the public-offer preflight accepted the email-first route without requiring checkout-click instrumentation. The next revision moved the buyer-facing claim from abstract product description to the buyer's concrete situation: needing to send a follow-up email after a call. The following revision reduced trust burden by making the public page lead with no app, no upload, and no checkout, while the autonomous acquisition copy said the buyer could reply with a rough draft or bullets without clicking a link, uploading anything, downloading anything, creating an account, sharing a password, entering a card, or paying before preview. The next revision removed the defensive first impression and made RelayBrief read as a normal manual email-cleanup service:

a few non-private bullets go to Alan by email, private details can be removed, a draft comes back first, and the USD 1 Stripe link comes only if the buyer uses it. The following revision narrowed the ask again: buyers were asked for the rough follow-up draft they already intended to send, or a few non-private bullets, and the returned object became one cleaner email. The next revision changed the economic job rather than only the input burden: the buyer is asked for one stuck lead, last reply, rough draft, or a few bullets, and the returned object is one follow-up email meant to get a yes, no, or next step. A later revision made the page less product-like and more explicitly human: email Alan one stuck lead, get one plain email back, and pay USD 1 afterward only if it is useful. The reply-autoclose preflight was aligned to that same stuck-lead preview-first path so the proof system no longer treats the new buyer wording as a missing checkout-link path. The next revision treated the buyer's immediate unfinished draft as the control target: the page said "Send the rough follow-up. I'll make it sendable", updated privacy, terms, sample, manifest, link-preview image, intake page, and outbound/reply templates to the same rough-draft rewrite framing, kept the Stripe link after the useful draft, and fixed the active outbound preflight so this email-first path was accepted without requiring a checkout link in cold outbound. The next revision changed the sample and active buyer language again because the previous proof was still too generic to justify even a USD 1 ask: `relaybrief.com` asked for one stuck follow-up, last reply, rough draft, or bullets, promised a subject line plus one email, and showed a stalled quote follow-up that asks for a yes, no, or next step. The following revision narrowed the commercial job further from writing assistance to stale-lead recovery: the page asked for one stale lead, quote, inquiry, or client thread that went quiet, returned one next email, and asked for USD 1 afterward only if the result helped. A later revision narrowed the buyer-visible job again from stale-lead recovery to unanswered quote follow-up. The current revision broadens the visible object from quote-specific recovery to one stuck client email: the public page, sample PDF, link-preview image, public-offer smoke check, outbound templates, reply handlers, and post-interest copy now ask for one stuck client email, last message, rough draft, or a few bullets. This update is recorded because zero revenue persisted after attention and sample inspection, so the controlled variable became buyer-visible trust, action familiarity, and commercial concreteness rather than price, volume, or the number of deliverables.

The 2026-05-22 operating read exposed a related cap-reservation fault. The controller advanced `paid_test_explicit` from 10/20 to 14/20, then spent the remaining six daily sends on older direct follow-ups while the active sample still lacked six first-touch sends. The code path was revised so

missing active-sample slots are reserved against the remaining daily cap on each tick, not against the original daily cap. The sample therefore remains uninterpretable until fresh active first-touch leads are refilled and sent.

The 2026-06-01 read adds a different execution boundary. The active sample did not advance beyond 14/20, and the money loop reported `disabled` with `paused_by_owner_cost_control`. This pause is not a buyer result. It means the autonomous refill and send path cannot produce the remaining six first-touch sends until the cost-control pause and lead-refill path are deliberately reopened or replaced.

2.7 Report language is part of the control surface

On 2026-05-06, the operator-facing Relay report language was changed from directive best-move framing to current-state framing in the autonomous operations summaries and related close-path/proposal text. This is not cosmetic. A system-owned operation should report state and current control logic without making the human operator read a task list for actions that the system should already own. The change is recorded here because paper-backed operational systems must keep their manuscripts aligned with meaningful dashboard, backend, and report-surface changes.

2.8 Compact journaling makes the system publishable as a research object

Before the journal layer, Relay could be monitored, but later reconstruction required multiple routes and human memory. The research-journal endpoint changes the status of the deployment because the system now records compact control entries. These entries preserve the money state, active-sample progress, proof state, health state, and current operation.

Table 5. Latest compact research-journal entries in the current live read. Buyer personal data are excluded from the manuscript.

Created at	Summary
2026-05-31 2:14 AM Eastern	money_loop money=$0.00 payments=0 sent=0 active_delta=0/0 active=14/20 proof=not_expected next=Refill fresh first-touch leads for the active outbound experiment
2026-05-31 2:14 AM Eastern	success_control money=$0.00 payments=0 bottleneck=active_experiment_refill proof=refill_direct_buyer_leads health=waiting_for_proof_deadline conversion_sent=0 failures=0
2026-05-31 2:09 AM Eastern	money_loop money=$0.00 payments=0 sent=0 active_delta=0/0 active=14/20 proof=not_expected next=Refill fresh first-touch leads for the active outbound experiment
2026-05-31 2:09 AM Eastern	success_control money=$0.00 payments=0 bottleneck=active_experiment_refill proof=refill_direct_buyer_leads health=waiting_for_proof_deadline conversion_sent=0 failures=0

3 Discussion

Relay is not yet evidence that an autonomous agent can make money without an operator. It is evidence that an autonomous commercial experiment can be constrained and logged so that a later money claim would be auditable. That is the stronger contribution. The system's most important result is the discipline of refusing to count an internal test as revenue, refusing to judge demand before the active sample and reply-observation window mature, refusing to change many variables at once, and separating buyer-signal failure from execution failure.

The broader implication is that agentic systems need proof contracts when deployed in domains with real-world consequences. A proof contract makes autonomy less theatrical and more scientific. It identifies what the system may do, what it must not do, when the result can be judged, and what evidence changes the next state. This structure could apply beyond Relay to other live agent deployments where market response, human intervention, or hidden tool actions make claims difficult to interpret.

The current deployment remains preliminary. It is a single system, a single offer, and a small descriptive dataset. It contains zero real revenue at the initial evidence lock and zero real revenue in the latest live check. It does not provide inferential statistics, external replication, or a mature ethics protocol. A Nature-level version of this work would need repeated journaled rotations after matured observation windows, non-test buyer outcomes, a comparison with manual or dashboard-only operation, a privacy-preserving dataset, independent review of code and logs, and a clear institutional position on commercial outreach and data retention.

4 Methods

4.1 Evidence lock and data sources

The initial manuscript lock used three saved production snapshots: `relay_ops_snapshot_2026-05-04T0042.json`, `relay_journal_snapshot_2026-05-04T0042.json`, and `relay_intent_snapshot_2026-05-04T0042.json`. The code state used for the initial evidence lock was commit `be6d9104aebd10dcf42b03d54f2d0ce1b8a9043b` in repository `nalalalan/relay-app`. The current-state update was read at 2026-06-01T12:17Z, or 8:17 AM Eastern, from the live production `evidence-export`, `ops-check`, `decision-ledger`, and `research-journal` routes. Raw snapshots and live responses are retained locally or in production systems for audit but are not reproduced in the public manuscript because they may contain operationally sensitive fields or personal identifiers.

4.2 Revenue accounting

Revenue was read from the money-system summary and success-controller state rather than from raw acquisition status counts alone. This prevents internal or unmatched Stripe test events from being counted as real buyer revenue. At the evidence lock, the money-system summary reported USD 0.00 gross revenue and zero payments. That value is used throughout the manuscript.

4.3 Intent and internal-test filtering

Intent was reported in two forms: all recorded intent events and filtered real-event counts where available. This is necessary because early deployment testing can otherwise mimic buyer demand. The manuscript reports aggregate counts only and excludes buyer emails and session identifiers.

4.4 Autonomy policy

The autonomy policy was read from the success-controller and money-system summaries. The policy defines the current bottleneck, allowed operation, forbidden actions, proof deadline, owner-interruption state, and recovery conditions. These values are treated as experimental controls. A future protocol deviation should be recorded in the journal before the corresponding outcome is interpreted.

4.5 Paper continuity

Relay is now treated as a paper-backed operational system. Any substantive change to the dashboard, backend, report language, source graph, evidence model, metrics, guardrails, or results must update the manuscript source and public PDF in the same work cycle. A paper-backed change is incomplete until the PDF rebuilds and the public artifact exposes the revised record.

4.6 Statistics and sample size

No hypothesis test was performed. The initial evidence-lock observations were descriptive: 56 page views, 18 filtered real page views, 8 link clicks, 3 checkout clicks, 3 lead-capture events, 1 filtered real lead, 60 rolling seven-day outbound sends, 0 replies, 0 payments, and 0 send failures. The latest live operating observations are also descriptive: 21 seven-day page views, 11 seven-day page-engagement events, 4 scroll-depth events, 7 sample clicks, 0 email-intent clicks, 20 rolling seven-day outbound sends, 0 replies, 0 unhandled replies, 0 send failures, 0 tracked outbound visits, 0 payments, and USD 0.00 gross revenue. The broader 30-day public-intent window contains 227 page views, 292 page-engagement events, 52 scroll-depth events, 30 sample clicks, 5 note-intake clicks, 2 link clicks, 2 checkout clicks, 194 sessions, 35 action sessions, and 45 sessions with at least 10 seconds of engagement; filtered counts show 221 real page views, 29 real sample clicks, 4 real note-intake clicks, and zero real leads. The active outbound state is `active_experiment_refill`: `paid_test_explicit` is incomplete at 14/20, with no fresh active first-touch leads due, no estimated active-sample send-window capacity, and the money loop disabled under owner cost control at the read time, so the current observation is an execution/refill-capacity finding rather than evidence of demand failure. The readiness state is an implementation-readiness measure, not a revenue result; the missing endpoint remains a real non-test buyer payment. Future statistical claims require completed samples, matured observation windows, and a predeclared analysis plan.

4.7 AI assistance disclosure

This manuscript draft was prepared with OpenAI Codex under author direction using saved production snapshots and cited sources. The language model is not an author. The human author is responsible for factual verification, final approval, authorship decisions, and disclosure according to the target journal policy (*8*).

5 Data availability

The minimum dataset for this draft consists of the three saved JSON snapshots named in Methods, the current production API reads, and source code needed to reproduce the route summaries and compact journal entries. Current availability artifacts include the buyer-facing product page `https://relaybrief.com/`, the live Relay dashboard `https://relay.aolabs.io/`, and the public paper route `https://relay.aolabs.io/paper.pdf`. Raw evidence was read from the production API base `https://alan-operator-backend-production.up.railway.app/api/relay` using `evidence-export` (365 days, limit 1000), `ops-check` (30 days), `decision-ledger` (90 days, limit 200), and `research-journal` (90 days, limit 200). Before public release, buyer emails, session identifiers, secrets, and operationally sensitive fields should be removed or transformed into privacy-preserving identifiers. A formal submission should deposit a frozen, sanitised dataset and schema in a DOI-minting repository or explain controlled-access restrictions (*9*).

6 Code availability

The implementation corresponds to repository `nalalalan/relay-app` on the public `main` branch for this paper update. Public release status must be confirmed before submission. If the code is central to the claims, peer reviewers should receive access to route definitions, the journal schema, a configuration template without secrets, and scripts or instructions needed to regenerate the manuscript figures from sanitised evidence.

7 Ethics and privacy

The system involves commercial outreach and buyer-intent logging. A complete study should specify lawful basis, retention period, opt-out handling, suppression-list behaviour, limits on automated follow-up, handling of personal data, and publication redaction rules. The present manuscript does not

publish buyer personal data and separates internal tests from buyer signal.

8 Competing interests

The author owns or operates AO Labs and Relay and may benefit commercially if Relay succeeds. This commercial interest should be disclosed in any submission.

9 Funding

No specific funding source was recorded in the evidence used for this draft. The author should update this statement if the work falls within a grant, fellowship, institutional project, or company-funded effort.

10 Author contributions

A.N.P. conceived the system and research framing, directed the deployment goals, owns factual verification of the evidence, and is accountable for the final manuscript. AI assistance was used for drafting and formatting under author direction and is disclosed in Methods.

References

1. S. Yao *et al.*, ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations* (2023).

2. T. Schick *et al.*, Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761* (2023).

3. J. S. Park *et al.*, Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442* (2023).

4. Q. Wu *et al.*, AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155* (2023).

5. L. Wang *et al.*, A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432* (2023).

6. J. O. Kephart, D. M. Chess, The vision of autonomic computing. *Computer* **36**, 41–50 (2003).

7. A. G. Kusne *et al.*, On-the-fly closed-loop materials discovery via Bayesian active learning. *Nature Communications* **11**, 5966 (2020).

8. Nature, *Artificial intelligence policy* (2026; https://www.nature.com/nature/editorial-policies/ai).

9. Nature, *Reporting standards and availability of data, materials, code and protocols* (2026; https://www.nature.com/nature/editorial-policies/reporting-standards).