The Security Patch

A checked identity for agents, and payments that carry their own proof

Lux Automations, Edmonton, Alberta, Canada · November 2026

Stage, stated plainly: the pair described in this brief is at prototype stage. It is proven on public test networks and in our own drills today; it has not yet run live, and the independent audit has not yet been performed. Everything claimed below carries the best evidence that exists for it — public receipts where they exist, and our own records, labelled as ours, where they don't.

1. The problem

Autonomous agents now negotiate, deliver work, and settle payments in seconds — and the evidence those transactions rest on has not kept pace. A log written by the party it describes can be rewritten by that party. A reputation score can be farmed. A name can be copied. When the worker is an agent, the question every transaction turns on — *should this money move, and is this agent what it claims to be?* — has no answer the other side can check.

The Security Patch is our answer, in two halves that never blur: an identity half that proves who an agent is, and a money half that makes its payments provable. The machinery runs on our side, behind a black box; what a

customer or an evaluator ever sees is behaviour and receipts — a thin plugin at the door, and a signed yes or no.

2. The identity half

The identity half issues a checked identity for an agent: who the agent is, who answers for it, and whether its record is clean. It never touches money. It is not a wallet. It protects.

- **Checked issuance.** The identity is built to be issued only after the agent and its accountable owner pass a behavioural gate — at prototype stage today: the issuance seam is proven to refuse when the gate says no (in-house battery), and the behavioural battery itself is still being built.
- **Door checks, two assurance levels.** A standard check asks: is this identity valid, unrevoked, and really this agent? — answered as a signed yes/no. A high-assurance check adds record, behaviour status, and mandate verification, and runs before a payment is honoured. The checking plugin is thin; the valuable machinery never runs on the customer's machine.
- **Behaviour watched in use.** The check is not once-at-issuance. In drills, a credential acting out of pattern is flagged, and revocation is what kills it.
- **Revocation with teeth.** A revoked identity is refused at every door that checks, as soon as the revocation lands — including re-entry attempts: in drills, a burned identity's credential cannot re-register under a new name.
- **Sybil-resistant reputation.** Reputation is earned on a record that starts clean and is kept. A closed ring endorsing itself counts as one voice, not many; independent voices count independently. (Design position, proven in drills — see §5.)
- **The tab.** Every identity and every door carries a running tab: counts, timestamps, outcomes (valid / revoked / challenged). The tab is the

bill, the behaviour record, and the customer's own dashboard in one. It meters the knock, never the content of the customer's business — and a tampered tab entry is detectable: rewrite one entry and the chain no longer verifies.

3. The money half

The money half moves payments only between verified identities, inside the owner's mandate and limits. An over-limit or out-of-mandate attempt is refused and recorded before money moves.

- **Escrow with a referee, both directions.** Funds lock in the contract — custody is the contract's, never ours, so open contracts settle even if we disappear. Release runs on proof-of-delivery plus a quality gate; disagreement goes to a dispute path ending at a human referee. On native ledger rails the same pattern runs with no contract at all: the ledger itself locks, releases to the payee when the time passes, or cancels after expiry with funds returned to the owner in full.
- **No custody, ever.** Currency conversions run on partners' rails at the partner's rate, shown as the partner's, never marked up in secret.
- **Metered charging.** Door checks are metered per use against a prepaid balance — no balance, no honouring. Our only take on the money side is a commission on protected payments, taken at the till in the same flow; the tax portion of every charge is calculated first and tracked in our books, so what is owed is always known, on time, every time.

4. The audit layer

Over every deal sits a tamper-evident audit layer: the terms are signed and anchored before work starts; the event log is hash-chained, so neither party can quietly rewrite it; proof-of-delivery is bound to the goods themselves; and the receipts can be verified by a stranger's software without phoning anyone. This layer is the subject of our research report, *Who Do You*

Believe When the Worker Is an Agent?, filed with the Apart Research "AI x Epistemics" Sprint (Track 2: Trust Infrastructure) in November 2026.

> **The stated limit, in the open:** the record proves continuity and provenance — that a record existed and was not altered, that a payment moved exactly as the written terms directed or visibly didn't. It does not prove the original record was true, and it does not prove the work was good.

5. The receipts

Receipts run in two columns that are never merged: identity receipts prove the identity side works; money receipts prove the money side works. Neither column is ever cited as evidence for the other.

Money side — public testnet receipts

- **HashKey Chain testnet escrow.** A deployed escrow contract, with two scenes anyone can open on the public explorer: a clean deal (funded, delivered, settled — 250 test dollars to the worker) and a staged dispute resolved as a refund. The contract source is public (github.com/cyrvance53/agentpay-escrow), and an internal audit confirmed the deployed code is byte-identical to the published source. The audit note itself has no public URL; the repository is the public artefact of record.
- **XRP Ledger testnet escrow — release.** On native ledger rails, with no contract written or needed: the ledger locked 1 XRP and released it to the payee. Read-back: the payee's balance rose by exactly the escrowed amount; while locked, neither party could touch it.
- **XRP Ledger testnet escrow — cancel, including the failure.** The first cancel attempt *failed*, and we publish it: the design set the finish-after time after the cancel-after time, and validation rejected it before any transaction was submitted — the ledger's real rule, caught client-side, nothing locked, nothing to clean up. The design was corrected in the open and re-run: the escrow was created, left

unclaimed past expiry, and cancelled, with the locked funds returned to the owner in full and the payee's balance exactly unchanged. The failed attempt is not counted in any total.

- **A flaw, published by us.** Our internal audit found a flaw in the first escrow version: a dispute path that could lock funds with no exit. It is named in the research report's own Limitations section, and the fix is carried as a requirement for the next version. Nothing anywhere implies the flaw is fixed in a shipped product.

Identity side — testnet records and in-house drills

- **Public testnet records.** Identity records — register, anchor, revoke — have been written and read back on the XRP Ledger and Stellar test networks: a door read an identity as live, and after revocation read the same identity as revoked. On the XRP Ledger, a full endorsement graph was written on-chain and the reputation scorer was fed only from the chain: a closed five-member ring scored one voice; three independent endorsers scored exactly three.
- **In-house drill record.** A local battery covers the behaviours the doors depend on: an honest agent at the door; the issuance seam refusing when the gate says no; a cloned key rejected; a stolen key valid before revocation and rejected after it; a revoked identity refused re-entry; mandate breaches and replayed signatures refused; a tampered tab entry caught at exactly the tampered entry. These are in-house records, labelled as ours — real behaviour on our own machines, with no third party watching.

6. The testing record — and its honest limits

- **The cryptography caveat.** In the local prototype, the signature scheme is implemented in pure Python from the public specification, because the audited packages are not installed on the build machine and the build is offline by design. The implementation self-checks and is faithful — but it is not constant-time and is not production

cryptography. Swapping in an audited library at the same seam is required before anything live, and has not yet been done.

- **The gate is a seam, not the Gate.** The battery proves the issuance seam refuses when told to refuse. The behavioural battery that will judge *whether* to refuse is its own design work, not yet built.
- **The stolen-key window is real.** A copied key signs validly until revocation. The prototype proves revocation's teeth; detection speed in production is an operational property, not something a local build can demonstrate.
- **No drill numbers are product performance.** In-house results are records of drills, and are cited only as that.

7. What has not run yet — stated plainly

- The dead-clone live trial: the first live identity issued to a disposable clone that is nobody, in a sealed test void, every attacking move recorded, the clone and its identity deleted at the end. Not run.
- Paired runs: no run yet has exercised identity and money together, end to end. Solo first, paired second — the paired receipts do not exist yet.
- The public test door, where anyone may attack a live demo and watch it hold. Not built, not open.
- Four further test networks whose drill scripts are written and ready. Not run.
- The behavioural issuance battery behind the seam. Not built.
- The audited-cryptography swap. Not done; required before anything live.

8. The audit

AgentPay has been accepted into a security subsidy grant program (Batch 1) toward an independent security audit. **The audit has not yet been performed.**