CSE3000 Research Project · EEMCS · Delft University of Technology

An Owner-Sovereign Integrity Layer for an AI Agent

Feasibility, and the cost of strict enforcement

Lucas Pereira Barrios Supervisors: Johan Pouwelse, Bulat Nasrulin

Three standard cryptographic primitives give a self-hosted AI agent an integrity layer its owner alone controls. No central authority, almost no cost.

An autonomous agent fails silently

Self-hosted agents act without a human in the loop. That autonomy lets three attacks happen unseen.

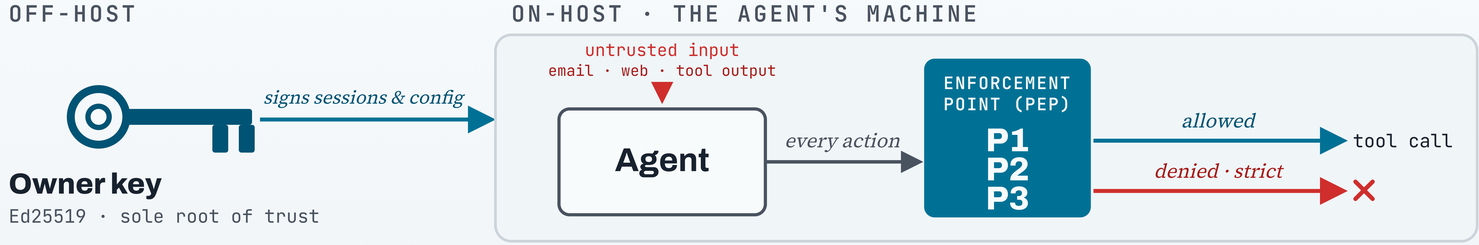
IDENTITY
Impersonate the owner
Issue instructions the agent treats as genuine.

CONFIGURATION
Edit the rulebook
Change what the agent is allowed to do.

ACTION LOG
Rewrite the action log
Erase the record of what it actually did.

One key, held off the host, is the only root of trust

The owner keeps one **Ed25519 key off the host** and never hands it over. No certificate authority, no consensus. Every action passes through **one enforcement point**.



The owner's private key never reaches the host. On-host primitives verify owner signatures and chain the action log; the owner independently anchors and re-verifies that log off-host.

Three attacks, three primitives

→ IMPERSONATE THE OWNER

P1
Owner-Anchored Identity

Every session carries an **Ed25519 signature**, checked against the owner's off-host key. Forge it without that key and **strict mode denies the call**.

✓ **Forged owner session denied 10 / 10**

→ REWRITE THE ACTION LOG

P2
Signed Action Log

Sign-then-chain: each entry signs its link to the one before. Edit any entry and its **signature stops verifying**. Hashes are public; signatures are not.

✓ **97-entry log, tamper caught 3 ways**

→ EDIT THE RULEBOOK

P3
Tamper-Evident Configuration

One **digest** covers the constitution, the tool and model allowlist, and the MCP setup. Change a single byte and it **no longer matches the owner's signature**.

✓ **Tampered configuration denied 10 / 10**

Strict enforcement costs almost nothing

0 of 566 owner actions
wrongly blocked

Every one of 566 legitimate owner actions was allowed, across four AgentDojo suites, run twice, on Haiku 4.5. The true loss rate stays below **0.67% (pre-registered Wilson 95% bound)**.

Where it holds, and where it does not

No attack-success-reduction claim
Baseline attack-success already floors near zero on a 2026 model. This measures **feasibility and cost, not prevention**.

Host compromise
Signatures protect **data, not code**. Off-host owner verification is the backstop.

Replay
No per-session nonce yet, so a captured session can be **replayed**.

Static configuration
Zero cost holds for fixed config; a legitimate change has an **unmeasured re-attestation cost**.

Where it goes next

→ **Cross-agent log verification** , no central server.

→ A **web of trust** over owner keys.

→ **Tamper-evident config** across a fleet.