

Agents that prepare institutional transactions, with custody and the signature left exactly where they are

A short view of Steward and the Aomi execution harness underneath it, written for a custodian.

In one sentence. Aomi is the execution layer that turns an instruction from a person or an AI agent into an exact, simulated, policy-checked transaction for the owner's own signer. Steward is our institutional application on it: vault and treasury operations where the agent prepares the work and the institution keeps the decision, the custody and the signature.

01 The problem a custodian sees

- **Institutions want agents in treasury and vault operations**, for faster valuation, faster incident response and fewer spreadsheets between a risk signal and a signed transaction.
- **They can't hand an agent a key, and a signature isn't a review.** An approver needs to know why a transaction exists, what it does to the position, and that it matches the mandate.
- **Models are not byte-exact.** In our testing, a 452-byte calldata payload relayed through a model came back as 441, 453 and 461 bytes. Any design that lets the model write transaction bytes will eventually sign the wrong thing.

02 What we built

The Aomi harness is a hosted, non-custodial, wallet-agnostic runtime. Typed, protocol-scoped tools build the transaction; it is simulated on a fork of live chain state; policy hooks check it; and it is handed to the owner's signer in manual, auto-approve (scoped session) or delegated mode. The model chooses intent and arguments, never raw bytes. Aomi runs today as a chat app, CLI, MCP server and SDK across EVM chains and Solana, including a [mainnet transfer on Arc](#) executed through the CLI.

Steward adds what an institution needs on top: a block-pinned NAV Oracle reconciled against posted values, an Incident Playground for stress scenarios against the mandate, and a Transaction Builder that scores alternatives, encodes the chosen plan exactly, simulates it and emits an unsigned Safe package. Its live incident replay reconstructs a real January 2025 Morpho vault response from public Ethereum data at steward-aomi-labs.vercel.app/app/replay.

03 Why this matters to a custodian

Custodian requirement	How Steward and Aomi meet it
Nothing moves without our controls	Steward has no commit, sign or broadcast tool. Today it emits an unsigned, simulated Safe package for the owner's own signer. The pilot deliverable is the custodian-signer integration: delivering that package as a request for authorization into the custodian's own policy engine and quorum.
Approvers can audit what they sign	Every package carries the pinned snapshot, policy digest, alternatives considered, simulation result and target/calldata identity check, then the receipt after execution.
Policy can't drift	The policy digest is frozen per session and attached to every output. Plans are generated and scored by code, not improvised by the model.
Defense in depth	Aomi's three-gate model: runtime prevention (simulation and guards), an independently administered wallet policy (the custodian), and settlement-time assertion enforcement on supported routes.

04 Traction and stage

- **Early pilot with a working prototype.** The Steward app runs nine typed tools on the Aomi runtime (replay, verify, inspect vault, policy, plan, simulate, finalize, verify execution, export evidence). The public site and incident replay are live.
- **Early conversations with RWA providers and institutional vault operators** about vault operations, oracle incidents and FX workflows.
- **Circle ecosystem:** accepted into Circle's Arc Acceleration Season (September 2026).
- **Funding:** \$1.5M pre-seed at a \$12M valuation, backed by Anagram and Nascent.

05 Business model

Outcome-based pricing. Because the harness simulates an action's result before it is signed, we price the outcome, not the API call. **Hosting on the Vercel model:** applications ship to Aomi

as compiled Rust plugins; we host and scale them and charge per app per month. Steward is the first institutional application on that runtime.

06 Where Anchorage fits

The next Steward milestone is the custodian signer: mapping the unsigned package to a custodian's transaction-request and approval APIs, so quorum policies, address allowlists and settlement workflows stay where they are today. We would like Anchorage as the design partner for that integration, and as an investor in the company building it.

Cecilia Zhang, founder · cecilia@foameo.ai · [book a call](#) · deck scrum.aomi.dev/main-deck · aomi.dev · github.com/aomi-labs