RATION

Deterministic safety rails for AI agents with money.

soonami **Venturethon 11** (with KPMG) — **"On the Fast Track"**

`github.com/caelum0x/ration`

The problem

AI agents are getting **real spending authority** — approving refunds, funding cards against expected deposits, moving money between accounts — faster than anyone has built a trust layer for it.

Every team in this space runs into the same question:

> **"Who proves the agent didn't exceed what it was allowed to do — and can that proof be checked by someone who isn't the vendor?"**

Today's answer is almost always: **"we log it, trust us."**

Who has this problem (the market)

- **Fintechs & card-issuing platforms** running LLM-driven approval/decisioning loops — Ration's own home turf today.
- **Agent-commerce and agent-economy platforms** giving agents wallets, procurement authority, or negotiation power.
- **Auditors, regulators, and BaaS/bank partners** being asked to sign off on AI-operated money flows with no independent verification primitive.
- **KPMG's own lens on this problem** — audit and assurance for systems nobody can fully black-box trust — is exactly the gap Ration closes.

The product

Ration: a deterministic rule envelope an LLM can never override, plus a tamper-evident, independently-verifiable audit trail.

- Every money-moving decision is re-verified **three separate times** (agent → card-issuance → ledger-core) before it's allowed to happen.
- Journaled through a crash-safe write-ahead log, mirrored into a separate, **append-only, hash-chained** audit trail.
- That chain's tip is anchored to a **real NEAR Testnet smart contract** we wrote and deployed — so anyone, not just us, can verify the trail wasn't tampered with.
- A real, running **11-service system** (Rust ledger core + Go services) — not a hackathon-only demo app.

Proof — verifiable right now, not a slide

Claim	Evidence
Deployed, correct NEAR contract	`code_hash` independently re-read via raw RPC, matches our build exactly
3 real on-chain anchor commits	each independently re-confirmed via `EXPERIMENTAL_tx_status`, not just our own tooling
Replay/tamper guard enforced **by the chain**, not our code	deliberate replay attempt rejected on-chain: "*sequence must strictly advance...*"
Zero regressions	15/15 ledger-core tests, full `go test ./... -race` clean, 9 new NEAR integration tests
Judge can run it live, not just watch a recording	one command: `near-anchor demo` — recorded run already in the public repo

Built to **two independent external rubrics**: Airwallex's Agentic Banking Hackathon (original build) and NEAR's AI Security & Defense track — not self-graded.

Business model (hypothesis — not yet priced)

Ration is infrastructure, not an app — the buyer is **any platform that is about to give an AI agent spending authority**:

- **Verification-as-a-service**: usage-based pricing per anchored audit event, for platforms that want a third-party-checkable trust primitive without building one.
- **Licensed deterministic-envelope engine**: embed Ration's rule-envelope + audit core directly into a fintech's or agent platform's own stack.
- **Open-source top-of-funnel**: the hand-rolled Go NEAR signer (no maintained SDK exists) ships as a standalone open-source package — a credible, developer-first way to reach the exact platforms above.

Honest flag: **no pricing has been tested with a real customer yet.** This is a hypothesis to validate, not a confirmed model.

Go-to-market

- **Lineage, not a cold start**: born at Airwallex's Agentic Banking Hackathon, hardened across 21+ real iteration rounds since — not a weekend demo left untouched.
- **Outbound to fintechs and agent-platform teams** already shipping LLM-driven money decisions (the exact audience in "Who has this problem").
- **Open-source the Go NEAR signer** as a developer-facing distribution channel into the NEAR/Web3 developer base.
- **Hackathon rubric validation as credibility signal**: two independent external judging processes (Airwallex, NEAR) as proof points in outbound conversations.

Traction — honest

- **$0 revenue, 0 paying customers today.**
- What exists instead of a revenue chart: **tech a panel can verify themselves, live** — real public repo, real on-chain transactions, a real one-command judge-facing demo.
- What's genuinely still missing: no production customers, no scheduled/automatic on-chain anchoring yet (today a manually-invoked tool, not wired into the live agent loop), no mainnet transaction — hardened infrastructure proven on Testnet, not a commercial product yet.
- That gap — proven safety primitive → sellable product — is exactly what this sprint is for.

The ask — what the 10-day sprint delivers

Per soonami's own rules (a genuinely **new** feature, not a restated NEAR-hackathon submission):

1. **Wire `near-anchor` into the live agent loop** as a scheduled, automatic anchoring job — today it's a manually-invoked CLI command, not triggered by the system itself.
2. **A judge-facing live dashboard** showing real-time anchor status, that lets a reviewer **trigger a tamper attempt themselves** and watch nearcore reject it live — instead of only watching a recorded run.
3. Validate the business-model hypothesis above with at least one real outbound conversation with a fintech/agent-platform team.

Team

Arhan Subaşı (GitHub: `caelum0x`) — solo founder/engineer. Builds production infrastructure: distributed systems, payment ledgers, AI-agent safety rails.

soonami's rules require at least 2 co-founders per team ("no 1-person teams"). Co-founder: [PLACEHOLDER — not yet recruited]. This is a real, open gap — not filled with an invented name. Recruiting a real co-founder/teammate (via Foundance's "create a Role" feature, or soonami's Discord) is an explicit prerequisite before this submission is eligible, and is tracked as an owner-only action, not a doc-fixable one.

Links

- Repo: **github.com/caelum0x/ration**
- Track: **On the Fast Track** (existing venture, not a new idea formed during this event)
- Demo video (same recording used for the NEAR Hackathon submission): `docs/near-hackathon/media/ration-near-demo-2026-11-11.mp4`
- Full technical writeup: `docs/near-hackathon/HACKATHON.md`

Thank you — questions welcome.