Security Oracles - a new active approach to Smart Contracts Security

Milestone Report

Team Sentinel

May 18, 2025

Contents

I. Introduction

Security challenges in decentralized systems continue to evolve, necessitating robust and adaptive threat detection strategies. This project set out to demonstrate how Security Oracles can enable Cardano smart contracts to adapt their execution logic in response to real-time threats. These oracles assume the existence of trusted and reliable detection systems capable of identifying such threats off-chain.

While the detailed design and implementation of detection engines fall outside the scope of this project, the purpose of this report is to explore potential strategies and methodologies for building them. We begin by reviewing the landscape of threat detection approaches in Web3, identifying those most compatible with Cardano's extended UTXO (eUTXO) model. We then propose several viable paths the Cardano ecosystem can pursue to incentivize the development and sustained operation of such infrastructure.

Through this exploration, we aim to lay the groundwork for ensuring that Cardano remains resilient in the face of evolving attack vectors—by equipping its contracts with the tools to respond intelligently and autonomously to credible off-chain threat intelligence.

II. The Threat-Detection Landscape

Off-chain security monitoring has matured rapidly, producing a spectrum of service models that differ in trust assumptions, scalability, and economic incentives. From a Cardano perspective, five architectural *archetypes* dominate today's Web 3.0 ecosystem.

Table 1.: Dominant archetypes for off-chain security monitoring

#	Archetype	How it works	Representative solutions	Typical output
1	**Centralized SaaS analytics**	Single vendor ingests full-node data (often multi-chain), enriches with heuristics/ML, and exposes APIs.	Chainalysis *Reactor*, CertiK *SkyInsights*, TRM Labs, Hypernative	Risk scores, tagged addresses, push alerts
2	**Decentralized monitoring networks**	Node operators stake a work token and run community-authored "detection bots"; social–economic consensus (reputation + slashing) enforces quality.	Forta, Pessimism (OP Stack), Caldera *Watcher*	Signed JSON alerts, on-chain attestations
3	**Oracle-integrated feeds**	Threat intelligence delivered through existing oracle rails; validator scripts verify signatures in-contract.	Chainlink *Data Streams*, Supra, RedStone *Security Feeds*	On-chain data points, e.g. `threat_level = 3`
4	**Mempool / pre-chain firewalls**	RPC middleware inspects pending transactions, scores risk, blocks or timelocks suspicious submissions before inclusion.	Forta *Firewall*, Blockfence *Guardian*, Hexagate	Allow / deny verdicts
5	**Protocol-native runtime monitors**	A protocol team embeds an off-chain daemon that tracks invariants and can pause contracts via admin keys or governance.	MakerDAO *Circuit Breaker*, Aave *Guardian*, Lido *Safety Module*	Admin transactions (pause, rate-limit)

1. Core functional building blocks

1. **Data ingestion layer** — archive/full nodes, mempool listeners, external APIs.
2. **Analytics & detection** — static rules, heuristics, machine-learning anomaly models, graph queries, transaction simulation.
3. **Alert signing & transport** — ECDSA/EdDSA signatures, threshold schemes, oracle aggregation.
4. **On-chain response hooks** — pausable contracts, UTXO-consuming "watchdog" scripts, roll-up settlement guards.

2. Key comparison dimensions

Table 2.: Important evaluation dimensions and current industry ranges

Dimension	Why it matters	Current industry range
Chain coverage	Cross-chain exploits propagate quickly; multi-chain visibility is mandatory.	Single-chain (e.g. Solana-only) $\rightarrow$ 20 + chains
Data freshness	Shorter detection lag reduces economic damage.	$\sim$250 ms (mempool firewalls) $\rightarrow \approx$1 block delay
Sustainable throughput	Determines the number of transactions/events an engine can analyse without bottlenecks.	800 tx/s (lightweight mempool filters) $\rightarrow <$ 10 tx/s (heavy ML inference)
Alert authenticity	Contracts must trust that alerts are untampered.	Vendor API keys $\rightarrow$ multi-sig oracle attestations
Operator incentives	Drives service longevity and quality.	Subscription fees, token staking rewards, DAO grants

3. Trends and gaps relevant to Cardano

- *From monolithic SaaS to modular networks* — teams pair deep forensic SaaS with decentralized real-time alert feeds.
- *Mempool-aware blocking becomes standard* — UTXO ecosystems require analogous "transaction guardians."
- *Convergence on signed-alert formats* — e.g. W3C Verifiable Credentials, Sign Protocol.

- *Economic experimentation* — work-token and pay-per-trigger models replace flat subscriptions.
- *Under-explored UTXO support* — most engines assume account-based state; adapting them to eUTXO presents an open R&D opportunity.

This landscape sets the stage for the next section, in which we distil *three concrete approaches* tailored to Cardano, complete with throughput, freshness, and response-time benchmarks required to deliver practical Security Oracles.

III. Cardano-Focused Approaches to Off-Chain Security Monitoring

In light of the landscape analysis, we identify *three* concrete aproaches to off-chain security monitoring that map cleanly onto Cardano's extended-UTXO (eUTXO) model

Table 3.: Benchmarks for Cardano-relevant monitoring approaches

Approach	Core idea	Sust. through-put[1]	Data freshness	Target response time
A1. Community Staked Detection Network	Forta-style network; operators stake ADA or a work token and run open-source "Cardano bots" that stream block and mempool data via Ogmios.	$\sim$50 tx/s per node; horizontal scaling with n nodes	<1 block (≈ 20 s)	Submit mitigation tx in <40 s total
A2. Oracle Attestation Layer	Chainlink-like oracle feed that publishes signed threat scores every block; contracts verify the signature in-script and adapt logic.	Negligible on-chain load; oracle must handle $\sim$3 MB/day data uplink	1–2 blocks (20–40 s)	Contract branch executed in the same block that consumes the attestation
A3. Treasury-Funded Watch-Dog Service	A central analytics provider (or consortium) funded by DAO/treasury grants runs deep ML models off-chain and submits watchdog transactions that pause/escrow funds when anomalies are detected.	Up to full-chain bandwidth ($\approx$250 tx/s) on provider side	<5 s event ingest; 1 block chain latency	Pause UTXO or flip datum in <30 s

1. A1 – Community Staked Detection Network

Architecture.

- Each operator stakes ADA (or a work token) into a `MonitorRegistry` contract.
- Nodes consume Ogmios/WebSocket streams, run detection bots, and sign alerts with Ed25519 keys registered on-chain.
- A lightweight on-chain aggregator contract accepts the *first* k matching signatures (threshold scheme) and mints a "`ThreatAlert`" UTXO that downstream contracts can inspect.

Advantages. Decentralised, censorship-resistant, aligns with ADA staking culture, can scale horizontally, and slashing discourages false alerts.

Challenges. *Bootstrap problem* (critical-mass of honest operators), higher complexity for threshold aggregation, and 1-block latency may be too slow for flash-loan-style attacks.

2. A2 – Oracle Attestation Layer

Architecture. A consortium of oracle nodes (e.g. re-used Chainlink infrastructure) pushes a signed JSON blob $\langle \text{tx\_hash, threat\_score} \rangle_{\text{sig}}$ into a reference script every block. Contracts validate the signature inside their spending scripts and reject or re-route value when `threat_score` exceeds a threshold.

Advantages. Minimal on-chain load, deterministic gas costs, and direct contract-level branching compatible with eUTXO statelessness.

Challenges. Relies on oracle honesty (Byzantine assumptions), still incurs 1–2-block lag, and requires every dApp to integrate the verification logic.

3. A3 – Treasury-Funded Watch-Dog Service

Architecture. A protocol treasury funds an ML-powered SOC (Security-Operations-Centre) that:

1. Streams full-chain data, external intel, and mempool submissions.
2. Runs heavy simulations (e.g. MBO "dry-run") to detect exploits.
3. Signs and submits a privileged "pause" or "freeze" transaction (multi-sig with protocol guardians) that flips a datum or spends a control UTXO.

[1] Throughput figures assume the current main-net parameter (`maxBlockBodySize` ≈ 88 kB) and typical block intervals of 20 s.

Advantages. Deep analytics capabilities, sub-block detection latency (< 5 s), clear accountability to the funding DAO.

Challenges. Centralisation risk, long-term OPEX burden, and possible over-reach (*false positives* causing unnecessary contract halts).

Summary

These three approaches span the decentralisation spectrum—from fully community-run to protocol-controlled—while meeting benchmark targets for throughput, freshness, and reaction time on Cardano. Subsequent work can combine elements (e.g. oracle attestations *plus* a staked network) to strike bespoke trade-offs for individual dApps.

IV. Economic and Governance Models for Cardano Security-Monitoring

Reliable monitoring infrastructure will not emerge without clear, self-reinforcing incentives. Below we evaluate four business models that could sustain off-chain threat-detection services in the Cardano ecosystem. Each model is assessed for stakeholder alignment, long-term sustainability, and practicality under current Cardano governance and treasury mechanisms.

Table 4.: Economic models for funding and operating Cardano threat-detection systems

Model	Core funding logic	Primary stakeholders	Incentive mechanism	Sustainability outlook
M1. Treasury Grants & DAO Bounties	Catalyst-style grants or protocol-specific DAOs allocate budgets for "Security Operations" milestones; recurring bounties for high-severity alerts.	Cardano Treasury, dApp-specific DAOs, security teams, community voters	Reputation + milestone payouts; bounty bonuses for critical saves	Medium: depends on continuous governance engagement and treasury health
M2. Work-Token Staking Network	Operators stake a `MONITOR` token; rewards stream from protocol-set "monitoring fee" charged per transaction. Slashing for invalid alerts.	Node operators, token holders, dApp developers, delegators	Block-by-block emission + fee share proportional to stake and SLA score	High if fee rate auto-adjusts to network load; boot-strap via airdrop
M3. Pay-per-Trigger Service Fee	dApps earmark a tiny fraction (e.g. 2–5 bps) of each transaction or vault harvest into an escrow. Smart contract pays *only* when a signed alert is consumed.	dApp treasuries, integrators, insurance underwriters, oracle operators	Direct revenue linked to successful detection; no work = no pay	Moderate–High: cost scales with delivered value; requires robust false-alert arbitration
M4. Enterprise SaaS + Insurance Wrap	Central SOC (Security-Operations-Centre) sells subscription tiers to exchanges, wallets, and institutional vaults; optionally bundles on-chain exploit insurance.	SOC vendor, custodians, institutional funds, reinsurers	Flat monthly fee + performance rebate; insurance premium offsets black-swan risk	High for enterprise segment; weaker community externalities

1. M1 – Treasury Grants and DAO Bounties

Mechanics. Funding flows from Catalyst funds or protocol-level DAOs to open-source monitoring teams via milestone-based contracts. Additional performance bounties can be crowdsourced: each dApp escrows a small ADA pool; the first detector to publish a valid alert that averts a loss receives the bounty.

Strengths. Leverages Cardano's established governance rails; no need for new tokens. Aligns community good-will with concrete deliverables.

Weaknesses. Lumpy cash-flow; subject to governance fatigue. Response time to new threats depends on DAO voting cadence.

2. M2 – Work-Token Staking Network

Mechanics. Mirrors Forta: operators bond `MONITOR` tokens; a monitoring fee (e.g. 0.02 ADA/tx) is hard-coded in a CIP and routed to a `RewardSplitter` script that pays stakers by SLA score. Malicious or low-quality nodes are slashed.

Strengths. Continuous, programmatic rewards; skin-in-the-game via slashing; permissionless entry promotes decentralisation.

Weaknesses. Requires new token economics and potentially divisive governance to set the base fee; initial cold-start of token liquidity.

3. M3 – Pay-per-Trigger Service Fee

Mechanics. Each integrator contract escrows funds into a `TriggerEscrow` datum. When a signed alert is consumed by the contract, the validator releases the pre-negotiated fee to the alert publisher. Arbitration layer adjudicates disputed or false alerts.

Strengths. Payment strictly proportional to value delivered; no idle overhead; easy to pilot on a per-dApp basis.

Weaknesses. Detect-then-pay loop demands robust on-chain arbitration for false positives; revenue can be bursty.

4. M4 – Enterprise SaaS with Insurance

Mechanics. A regulated SOC offers API keys, dashboard access, and signed on-chain feeds to large custodians, wallets, and funds. An optional insurance rider pays out if clients suffer a loss not prevented by the SOC's alerts.

Strengths. Clear legal contracts, guaranteed SLAs, predictable revenue, ability to fund heavy ML pipelines.

Weaknesses. Centralisation; limited coverage of long-tail dApps; trust anchored in legal recourse rather than on-chain slashing.

Comparative Insights

- **Stake-based networks (M2)** offer the best blend of continuous incentives and decentralisation, but need careful fee-rate governance.

- **Pay-per-trigger (M3)** aligns payment with actual defence value—ideal for high-TVL vaults that can tolerate micro-fees.
- **Treasury grants (M1)** excel for boot-strapping open tooling, whereas **enterprise SaaS (M4)** fills compliance-oriented niches.

A layered strategy—Catalyst grants for initial R&D, followed by a work-token network supplemented with per-dApp trigger fees—can share costs fairly across the ecosystem while preserving Cardano's ethos of decentralised, formally verified security infrastructure.