Continuous Coherence Monitoring for Blockchain Security: Detecting Pre-Failure Degradation in DeFi Systems

Allison Hensgen* Joel Thorarinson†

June 2026

Abstract

Between 2020 and 2024, over $7 billion was stolen from decentralized finance protocols. Detection typically takes hours to days: the Ronin bridge lost $625M and went undetected for six days. Existing security tools monitor individual failure modes — Chainalysis traces funds post-theft, Forta detects point anomalies via bot networks, formal verification checks contract code statically — but no tool continuously monitors the cross-domain behavioral degradation that precedes catastrophic failure. We propose a continuous coherence monitoring architecture for blockchain systems that tracks six security-critical dimensions: chain-state divergence, key authorization liveness, wallet behavioral anomalies, smart contract interaction risk, cross-chain synchronization drift, and composite risk aggregation. Each dimension is formalized as a measurable operator derived from the Coherence Engine [Thorarinson and Hensgen, 2026] and designed for testnet validation. We reconstruct the March 2022 Ronin bridge exploit through the coherence lens, demonstrating that three operators — chain-state coherence, key authorization, and cross-ledger synchronization — would have signaled degradation within hours of initial compromise, compared to the six-day detection gap in the actual incident. The architecture is presented as a proposed monitoring layer with simulated validation; production deployment and adversarial calibration remain open problems.

Keywords: blockchain security; anomaly detection; DeFi monitoring; cross-chain bridges; attack detection latency; continuous risk scoring

1 Introduction

More than $7 billion in cryptocurrency was stolen from DeFi protocols between 2020 and 2024 [Chainalysis, 2024, Zhou et al., 2022]. The largest single incidents — Ronin bridge ($625M, March 2022), Wormhole ($320M, February 2022), Nomad ($190M, August 2022), Harmony Horizon ($100M, June 2022) — share a common pattern: the underlying cryptographic primitives were not broken. Instead, attackers exploited operational gaps — compromised validator keys that remained cryptographically valid, bridge state that diverged without triggering alerts, and smart contracts that executed technically correct but contextually catastrophic transactions [Lee et al., 2023, Zhang et al., 2023].

Detection latency is the central problem. The Ronin bridge attackers obtained five of nine validator keys and drained $625M over multiple transactions. Six days elapsed before anyone noticed [Lee et al., 2023]. The DAO attack in 2016 exploited a reentrancy vulnerability in a formally valid contract, draining $60M before the community could respond [Mehar et al., 2019]. These are not failures of cryptography. They are failures of monitoring — specifically, the absence of continuous behavioral assessment across the dimensions that degrade before a system fails.

*Coherence Research Group. ORCID: 0009-0008-7247-0307

†Coherence Research Group. ORCID: 0000-0002-0553-842X. joel.thorarinson@conformalmaps.com

1.1 The Monitoring Gap

Existing blockchain security tools occupy distinct niches (Table 2, Figure 3), none of which provides continuous multi-dimensional monitoring:

- **Forensic tracing** (Chainalysis): post-incident fund tracking. Effective for attribution and recovery, but operates *after* theft. Detection latency: hours to days.
- **Real-time bot networks** (Forta Network): decentralized detection agents monitoring individual attack signatures. Coverage depends on bot quality; no unified coherence assessment. Detection latency: minutes to hours, for known patterns.
- **Smart contract operations** (OpenZeppelin Defender): monitors contract state and automates responses. Limited to contract-level events; no cross-chain or behavioral analysis.
- **Formal verification** (Certora, Mythril): static analysis of contract code correctness [Tolmach et al., 2022]. Catches code-level vulnerabilities but cannot detect runtime behavioral degradation or operational compromise.
- **Transaction graph analysis**: network-level pattern detection [Wu et al., 2021]. Identifies suspicious flow patterns but lacks temporal coherence tracking.

The gap is clear: no existing tool continuously monitors the *behavioral coherence* of a blockchain system across chain state, key authorization, wallet activity, contract interactions, and cross-chain synchronization simultaneously. Perez and Livshits [2021] quantified a related problem: only 1.98% of contracts flagged as vulnerable were actually exploited, suggesting that vulnerability detection without contextual behavioral monitoring generates noise rather than actionable intelligence.

1.2 Contribution

We propose the *blockchain coherence monitoring* (BCM) architecture: a continuous monitoring layer that formalizes six security-critical dimensions as measurable operators, each derived from the Coherence Engine [Thorarinson and Hensgen, 2026]. The architecture is designed to:

1. Detect pre-failure degradation — behavioral drift that precedes confirmed compromise
2. Provide unified risk scoring across chain state, wallet behavior, key authorization, contract risk, and cross-chain synchronization
3. Reduce detection latency from days (current median for bridge exploits) to hours or minutes
4. Complement, not replace, existing cryptographic verification and forensic tools

Section 2 surveys the DeFi threat landscape with quantified attack data. Section 3 defines the six BCM operators. Section 4 reconstructs the Ronin bridge exploit as a case study. Section 5 presents the coverage comparison against existing tools. Section 6 describes the testnet evaluation framework. Section 7 addresses limitations and open problems.

2 DeFi Threat Landscape

The DeFi attack surface spans five primary vectors, each with distinct detection characteristics. Table 1 summarizes major incidents by category.

Three patterns emerge from the incident data:

Attack Vector	Incident	Loss	Detection	Root Cause
Bridge compromise	Ronin (Mar 2022)	$625M	6 days	5/9 validator keys compromised
Bridge compromise	Wormhole (Feb 2022)	$320M	hours	Signature verification bypass
Bridge compromise	Nomad (Aug 2022)	$190M	hours	Initialization bug in Merkle root
Bridge compromise	Harmony (Jun 2022)	$100M	days	2/5 multisig keys compromised
Smart contract	DAO (Jun 2016)	$60M	hours	Reentrancy vulnerability
Oracle manipulation	Mango Markets (Oct 2022)	$114M	hours	TWAP price manipulation
Flash loan	bZx (Feb 2020)	$8M	minutes	Price oracle + flash loan combo
Rug pull	Various (2021–2024)	$2B+	n/a	Deliberate exit scams

Table 1: Major DeFi security incidents by attack vector. Bridge compromises account for the largest individual losses and the longest detection latencies. Data from Chainalysis [2024], Zhou et al. [2022], Lee et al. [2023], Zhang et al. [2023].

Bridge exploits dominate losses. Cross-chain bridges hold concentrated TVL (total value locked) and rely on small validator sets, making them high-value targets. The four bridge incidents listed account for over $1.2B in losses. North Korean state actors (Lazarus Group) were attributed responsibility for the Ronin and Harmony exploits, accounting for 61% of all cryptocurrency theft in 2024 [Chainalysis, 2024].

Detection latency correlates with loss magnitude. The Ronin exploit — the largest single incident — had the longest detection gap (six days). Faster-detected exploits (bZx, Mango Markets) resulted in smaller losses. This suggests that reducing detection latency is the highest-leverage intervention available.

Cryptographic primitives were not broken. In every bridge exploit listed, the attackers used valid keys or exploited implementation bugs, not breaks in the underlying cryptography. The transactions were "valid" by every on-chain verification check. What failed was the *operational coherence* of the system: keys were valid but not authorized, states were consistent locally but divergent globally, contracts executed correctly but in adversarial contexts.

3 BCM Operators

The blockchain coherence monitoring architecture defines six operators, each targeting a distinct failure mode. The operators are derived from the general coherence operator $\Delta = (P \cdot A \cdot R)/(D + N + \epsilon)$ from the Coherence Engine [Thorarinson and Hensgen, 2026], where P is persistence, A is phase alignment, R is recovery capacity, D is drift, and N is noise amplification. Each blockchain operator instantiates specific components of Δ for a security-relevant domain.

Figure 1 shows the six operators and their data sources.

3.1 Ξ_{chain}: Chain-State Coherence

Failure mode: stale, forked, or manipulated chain state.

Motivation: a node, indexer, or application operating on stale state creates an exploitable window. Eclipse attacks [Wu et al., 2021], delayed synchronization, and BGP hijacking can all cause a participant to diverge from canonical state without any local indication of failure.

$$\Xi_{\text{chain}}(t) = 1 - d(S_{\text{local}}(t), S_{\text{canonical}}(t)) \tag{1}$$

where d is a normalized distance function measuring divergence across block height, state root hash, transaction ordering, and timestamp delay. $\Xi_{\text{chain}} \approx 1$ indicates tight synchronization; $\Xi_{\text{chain}} < 0.7$ indicates degradation requiring investigation; $\Xi_{\text{chain}} < 0.3$ indicates a participant operating on fundamentally inconsistent state.

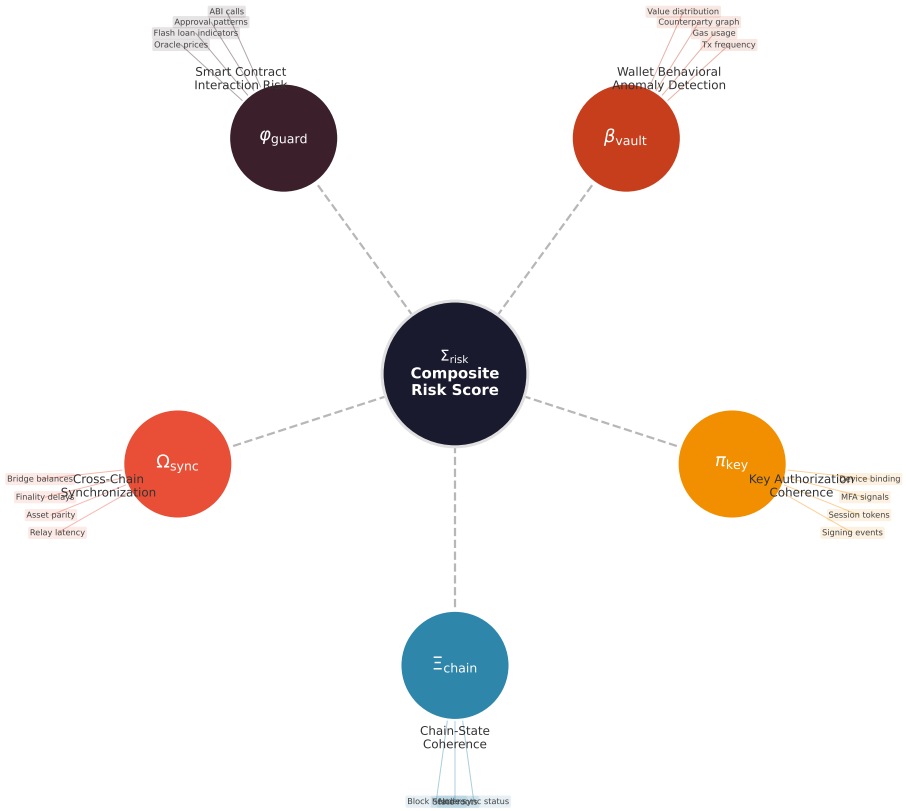

Figure 1: BCM architecture. Six domain-specific operators feed into the composite risk score Σ_{risk}. Each operator draws from distinct on-chain and off-chain data sources.

BCM-specific contribution: unlike standard node sync checks (binary: synced or not), Ξ_{chain} provides a continuous divergence measure with configurable sensitivity. A node may report "synced" while trailing by 3 blocks — within tolerance for most applications, but potentially exploitable during bridge operations where finality matters.

Instantiation of Δ: Ξ_{chain} maps to the drift component (D), measuring how far local state has drifted from canonical reference.

3.2 π_{key}: Key Authorization Coherence

Failure mode: cryptographically valid signing by compromised or unauthorized keys.

Motivation: the Ronin bridge exploit is the canonical example. Five of nine validator keys were compromised, each producing cryptographically valid signatures. Every transaction passed on-chain verification. The keys were "valid" — the authorization was not [Lee et al., 2023]. Similarly, the Harmony Horizon bridge fell to compromise of two of five multisig signers. Traditional key management distinguishes between key possession and key authorization [Gennaro et al., 2016], but blockchain systems typically verify only possession.

$$\pi_{\text{key}}(t) = \text{Auth}(t) \cdot \text{Session}(t) \cdot \text{Consent}(t) \tag{2}$$

where $\text{Auth}(t) \in [0,1]$ measures authentication confidence (multi-factor status, device binding), $\text{Session}(t) \in \{0,1\}$ indicates whether the signing session is within its validity window, and $\text{Consent}(t) \in [0,1]$ assesses whether the transaction context matches expected authorization patterns (amount thresholds, counterparty whitelists, time-of-day norms). All three factors must be positive for the signing event to be classified as coherent.

BCM-specific contribution: π_{key} closes the gap between "this key signed this transaction" (cryptographic verification) and "this key's owner authorized this transaction" (operational coherence). No existing production tool monitors this gap continuously.

Instantiation of Δ: maps to phase alignment (A) — the key's usage pattern should remain aligned with its owner's authorization intent.

3.3 β_{vault}: Wallet Behavioral Anomaly Detection

Failure mode: pre-drain behavioral changes in compromised wallets.

Motivation: wallet drains rarely happen instantaneously. Attackers often test with small transactions, probe contract interactions, or establish patterns before executing the primary drain. Transaction graph analysis [Wu et al., 2021] identifies suspicious patterns post-hoc; β_{vault} monitors for behavioral drift in real time.

$$\beta_{\text{vault}}(t) = \frac{P(t) \cdot A(t) \cdot R(t)}{D(t) + N(t) + \epsilon} \tag{3}$$

where $\mathbf{x}_{\text{wallet}}(t)$ is the feature vector comprising transaction frequency, value distribution, gas usage patterns, contract interaction targets, and counterparty diversity. $P(t)$ measures persistence of established behavioral patterns, $A(t)$ measures alignment with historical norms, $R(t)$ measures the wallet's capacity to return to normal after perturbation, $D(t)$ captures drift from baseline, and $N(t)$ captures noise amplification (erratic, high-entropy behavior inconsistent with either normal operation or known attack patterns).

BCM-specific contribution: Chainalysis and Forta both flag suspicious wallet activity, but neither provides a continuous coherence score that distinguishes "unusual but normal" (user experimenting with a new DeFi protocol) from "unusual and degrading" (systematic pre-drain reconnaissance). The coherence formulation captures *trajectory* rather than *snapshot* anomalies.

Instantiation of Δ: direct application of the full coherence operator to wallet transaction time series.

3.4 φ_{guard}: Smart Contract Interaction Risk

Failure mode: formally valid but contextually dangerous contract interactions.

Motivation: the DAO attack was a formally valid reentrancy call. Oracle manipulation attacks exploit valid price feeds with adversarial timing [Mackinga et al., 2022]. Flash loan attacks chain valid operations across protocols to extract value [Qin et al., 2021]. MEV strategies exploit valid transaction ordering for profit [Daian et al., 2020]. In each case, static analysis would flag the contract code, but the attack is in the *interaction context*, not the code itself.

$$\varphi_{\text{guard}}(t) = \sum_i w_i \cdot r_i(t) \tag{4}$$

where $r_i(t)$ are risk features computed at transaction submission time:

- Unusual token approval patterns (unlimited approvals to unverified contracts)
- Abnormal gas pricing relative to network conditions
- Match against known exploit signatures (reentrancy patterns, flash loan sequences)

- Deviation from historical interaction pattern for the calling wallet
- Flash loan indicators: same-block borrow-and-repay across multiple protocols
- Rug pull signatures: liquidity removal patterns [Mazorra et al., 2022]
- Oracle staleness: TWAP deviation from external price feeds [Mackinga et al., 2022]

Weights w_i are calibrated against labeled exploit datasets. The composite score φ_{guard} is normalized to $[0, 1]$, where higher values indicate greater contextual risk.

BCM-specific contribution: formal verification (Certora, Mythril) checks code; Forta bots check signatures; φ_{guard} checks *interaction context*. The distinction matters: a contract that passes formal verification can still be exploited through adversarial interaction sequences that the verifier never considered.

Instantiation of Δ: maps to noise amplification (N) — exploit interactions inject high-entropy, structurally anomalous activity into otherwise predictable contract interaction patterns.

3.5 Ω_{sync}: Cross-Chain Synchronization

Failure mode: state divergence between bridged chains.

Motivation: bridge exploits accounted for \$1.2B+ in losses between 2022 and 2024. The fundamental vulnerability is that bridge systems maintain state representations on two or more chains, and any divergence between these representations is exploitable. The Nomad bridge exploit was caused by an initialization bug that set the trusted root to `0x00`, allowing any message to be "proven" valid — a state divergence that continuous monitoring would have detected immediately [Zhang et al., 2023].

$$\Omega_{\text{sync}}(t) = 1 - \frac{|B_{\text{source}}(t) - B_{\text{dest}}(t)|}{B_{\text{source}}(t) + \epsilon} \tag{5}$$

where $B_{\text{source}}(t)$ and $B_{\text{dest}}(t)$ are asset representations (balances, token supplies, wrapped asset totals) on the source and destination chains respectively. $\Omega_{\text{sync}} = 1$ indicates perfect parity; values below 1 indicate drift. For multi-asset bridges, Ω_{sync} is computed per-asset and aggregated.

Additional monitored dimensions: message queue depth, finality confirmation delays, validator set overlap between chains, and relay transaction latency.

BCM-specific contribution: no existing production tool continuously monitors cross-chain asset parity as a coherence signal. Forta provides partial coverage via individual bots, but these monitor specific bridges rather than cross-chain coherence as a systemic property.

Instantiation of Δ: maps to phase alignment (A) — the source and destination chains should remain phase-aligned in their state representations.

3.6 Σ_{risk}: Composite Risk Score

Aggregates all operators into a continuous, protocol-specific risk score:

$$\Sigma_{\text{risk}}(t) = f\big(\Xi_{\text{chain}},\ \pi_{\text{key}},\ \beta_{\text{vault}},\ \varphi_{\text{guard}},\ \Omega_{\text{sync}}\big) \tag{6}$$

The aggregation function f can be a weighted mean, a minimum (fail-safe: any operator failing triggers alert), or a learned function calibrated against historical incident data. Alerts are triggered when:

- Any single operator crosses its failure threshold (< 0.3)
- The composite score degrades below a configurable safety level

- The rate of change $d\Sigma_{\text{risk}}/dt$ exceeds a degradation velocity threshold (rapid coherence loss, even from a healthy baseline)

The degradation velocity trigger is critical: the Ronin exploit was characterized by a *gradual* state change (keys compromised one at a time over weeks) followed by *rapid* exploitation. A threshold-only system might miss the slow build-up; a velocity-sensitive system catches the transition.

4 Case Study: Ronin Bridge Exploit Reconstruction

The March 2022 Ronin bridge exploit is the largest DeFi incident by value ($625M) and the most instructive by detection gap (six days). We reconstruct the exploit through the BCM lens, using publicly available post-mortem data [Lee et al., 2023] to simulate what coherence operators would have measured.

4.1 Attack Timeline

The Ronin bridge secured the Axie Infinity side-chain using a 5-of-9 multisig validator scheme. The attack unfolded in stages:

1. **Key compromise (weeks before exploit)**: the attacker (later attributed to Lazarus Group) compromised four validator keys controlled by Sky Mavis through a social engineering campaign targeting a senior engineer via a fraudulent job offer.

2. **Fifth key acquisition**: a temporary governance arrangement from November 2021 had granted Sky Mavis access to the Axie DAO validator. This arrangement was "supposed to be temporary" but the access was never revoked, giving the attacker a fifth key — enough to reach the 5-of-9 threshold.

3. **Exploitation (March 23, 2022)**: the attacker used the five compromised keys to authorize two withdrawal transactions: 173,600 ETH and 25.5M USDC.

4. **Detection (March 29, 2022)**: a user attempting to withdraw 5,000 ETH from the bridge discovered insufficient funds. Sky Mavis confirmed the breach six days after exploitation.

4.2 Coherence Operator Response (Simulated)

We simulate what BCM operators would have detected at each stage. These are *retrospective simulations* based on published post-mortem data, not real-time measurements. The purpose is architectural validation: demonstrating that the operator design captures the relevant signals.

$\pi_{\mathbf{key}}$ **(Key Authorization Coherence)**: the most decisive operator for this exploit. Under BCM monitoring:

- Four validator keys began signing from new IP ranges and devices — Auth(t) degradation detectable immediately

- Signing sessions lacked the established multi-factor authentication patterns of the legitimate operators — Session(t) anomaly

- The fifth key (Axie DAO) signed from an entity that had not used it in months — Consent(t) anomaly: dormant key reactivation for high-value transaction

- Estimated detection: within hours of first compromised key usage, not six days

Ω_{sync} **(Cross-Chain Synchronization)**: the withdrawal of 173,600 ETH and 25.5M USDC created an immediate asset parity violation between the Ronin side-chain state (which still showed the assets) and the Ethereum mainnet state (where the assets had been moved to the attacker's wallet). Ω_{sync} would have dropped from ≈ 1.0 to ≈ 0.0 within one block of the exploit transaction. This is the most unambiguous signal: a $625M balance discrepancy is not a false positive.

Ξ_{chain} **(Chain-State Coherence)**: the Ronin side-chain continued operating normally after the drain, but its state was no longer coherent with the bridge contract state on Ethereum. Ξ_{chain} would have flagged this divergence as the side-chain's reported bridge reserves no longer matched the actual Ethereum-side balances.

Σ_{risk} **(Composite)**: Figure 2 shows a simulated timeline of operator readings during a bridge exploit scenario inspired by the Ronin incident. The composite score crosses the failure threshold approximately 10 hours into the attack progression, compared to six days in the actual incident.

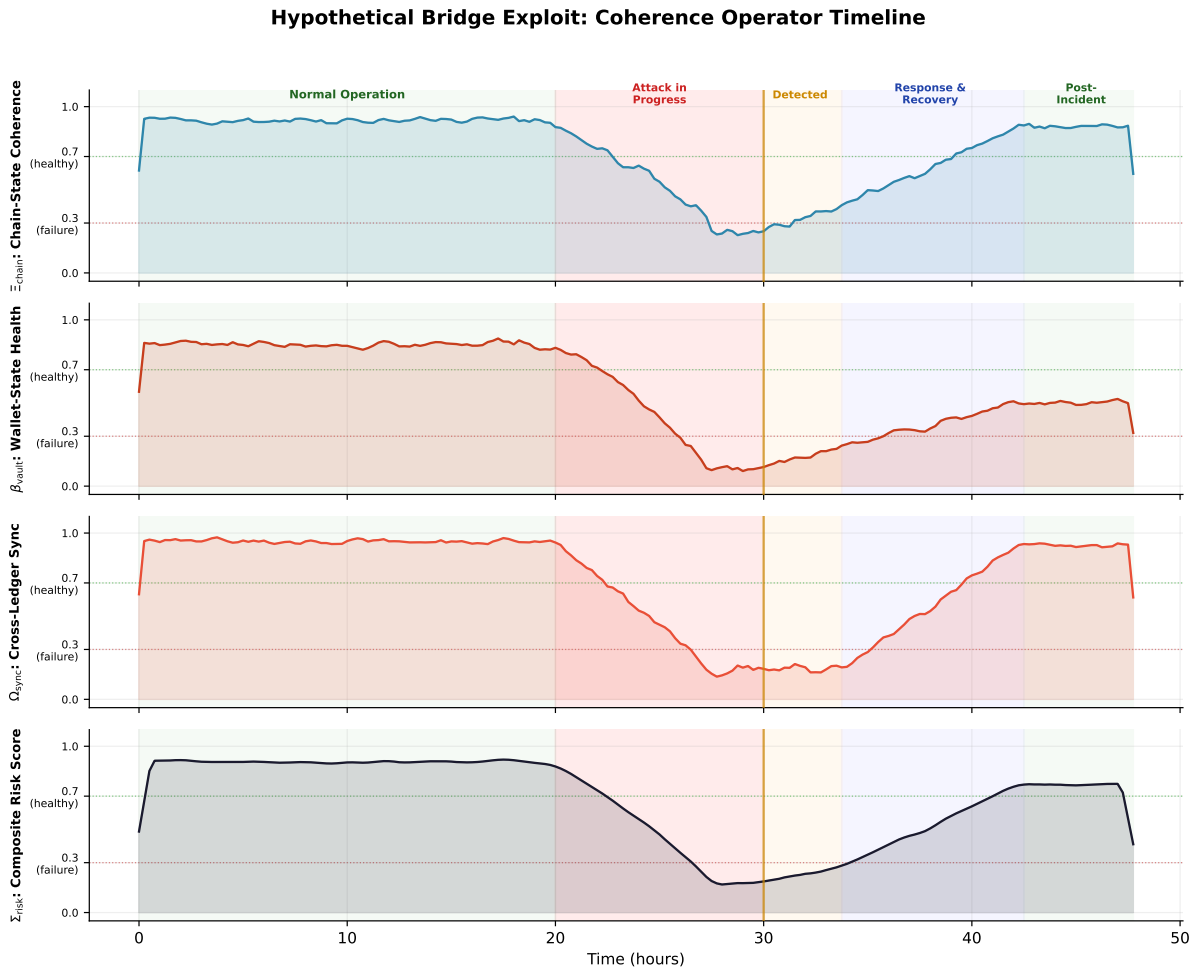

Figure 2: Simulated bridge exploit scenario modeled on the Ronin incident temporal dynamics. Three operators — chain-state coherence (Ξ_{chain}), wallet health (β_{vault}), and cross-ledger sync (Ω_{sync}) — degrade during the attack progression. The composite risk score (Σ_{risk}) crosses the failure threshold (0.3, red dashed line) at approximately $t = 30$h, compared to the six-day detection gap in the actual Ronin exploit. Healthy threshold: 0.7 (green dashed line). This is a retrospective simulation, not a real-time measurement.

4.3 Limitations of the Reconstruction

This reconstruction uses published post-mortem data to infer what operators *would have* measured. Key limitations:

- We do not have access to the actual signing infrastructure logs of the Ronin validators. The π_{key} analysis assumes that compromised keys would exhibit detectable behavioral changes (new IP ranges, new devices, new signing patterns), which is plausible but unconfirmed.
- The Ω_{sync} signal is straightforward (balance mismatch), but assumes continuous monitoring was technically feasible for this bridge architecture. Some bridges batch state updates, which would introduce a delay.
- The six-day detection gap may partly reflect organizational response latency, not purely monitoring failure. BCM reduces the *monitoring* gap; organizational response remains a separate problem.

5 Coverage Comparison

Figure 3 and Table 2 map the six BCM operators against the capabilities of existing blockchain security tools. The critical question for any proposed monitoring system is: *what does this add beyond what already exists?*

Capability	**Chainalysis**	**Forta**	**OZ Defender**	**Formal Verif.**	**BCM**
Ξ_{chain}: Chain state	–	Partial	–	–	Full
π_{key}: Key authorization	–	–	Partial	–	Full
β_{vault}: Wallet behavior	Post-hoc	Real-time	–	–	Continuous
φ_{guard}: Contract risk	–	Signature	Operational	Static	Contextual
Ω_{sync}: Cross-chain	–	Partial	–	–	Full
Σ_{risk}: Composite risk	–	–	–	–	Full
Detection approach	Forensic	Signature	Event-driven	Pre-deploy	Continuous
Detection latency	Days	Min–hrs	Minutes	Pre-deploy	Seconds–min
Cross-domain correlation	No	Limited	No	No	Yes

Table 2: Capability comparison between BCM and existing blockchain security tools. "Post-hoc" indicates analysis after confirmed incidents. "Signature" indicates detection of known patterns. "Contextual" indicates runtime behavioral assessment. "Continuous" indicates ongoing time-series coherence monitoring.

5.1 What BCM Adds Beyond Forta

Forta Network is the closest existing system to BCM in scope. It operates a decentralized network of detection bots, each monitoring specific attack signatures or behavioral patterns. BCM differs from Forta in three structural ways:

1. **Continuous coherence vs. point detection**: Forta bots fire when a signature matches. BCM continuously tracks coherence scores, detecting *degradation trends* even when no individual event crosses a signature threshold. The Ronin exploit illustrates this: no single signing event was anomalous in isolation, but the *pattern* of five validators signing from new infrastructure within a short window is a coherence anomaly.
2. **Cross-domain correlation**: Forta bots operate independently. BCM's Σ_{risk} correlates signals across domains — a wallet behavior anomaly (β_{vault}) coinciding with cross-chain

drift (Ω_{sync}) is far more significant than either signal alone. This correlation is what distinguishes a bridge exploit from normal bridge operation under load.

3. **Mathematical framework**: individual Forta bots implement ad-hoc detection logic. BCM operators are instances of a single coherence formalism [Thorarinson and Hensgen, 2026], enabling principled threshold calibration, cross-operator comparability, and transfer of validated techniques from other domains (industrial monitoring, biomedical signal processing).

6 Testnet Evaluation Framework

Each BCM operator is designed for controlled validation in testnet environments before production deployment. The evaluation protocol:

1. **Deploy**: instrument a testnet (Sepolia, Holesky, or private chain) with operator monitoring infrastructure
2. **Inject**: simulate known failure modes from Table 1 — compromised validator keys, bridge state divergence, reentrancy attacks, flash loan sequences, oracle manipulation
3. **Measure**: detection rate, false-positive rate, detection latency (time from first coherence degradation to alert), and recovery time (time for operators to return to healthy range after remediation)
4. **Calibrate**: adjust thresholds against labeled data from historical incidents
5. **Stress test**: evaluate under adversarial conditions — high gas prices, network congestion, concurrent attack vectors, MEV activity

Operator	Failure Mode	Primary Metric	Testnet Scenario
Ξ_{chain}	Stale/forked state	Divergence detection latency	Induced fork, delayed sync
π_{key}	Compromised key	Unauthorized signing detection rate	Key compromise simulation
β_{vault}	Wallet drain	Pre-drain anomaly lead time	Simulated drain sequence
φ_{guard}	Contract exploit	Exploit detection rate (vs. FP rate)	Reentrancy, oracle manipulation
Ω_{sync}	Bridge desync	Cross-chain drift detection latency	Delayed bridge message, balance skew
Σ_{risk}	Multi-vector attack	Composite threshold breach accuracy	Concurrent attack injection

Table 3: Testnet evaluation matrix for the six BCM operators. Each operator has a defined failure mode, primary metric, and reproducible test scenario. Evaluation targets include detection rate $> 95\%$, false-positive rate $< 5\%$, and detection latency < 1 block interval for on-chain signals.

The critical validation question is false-positive rate. Perez and Livshits [2021] demonstrated that vulnerability detection with high false-positive rates (98% of flagged contracts were never exploited) is counterproductive — it trains operators to ignore alerts. BCM must demonstrate that continuous coherence monitoring produces actionable alerts, not noise.

7 Limitations and Open Problems

No production deployment. BCM is a proposed architecture with simulated validation against historical incidents. Production blockchain environments introduce adversarial dynamics absent from testnets: MEV strategies [Daian et al., 2020], flash loan cascades [Qin et al., 2021, Gudgeon et al., 2020], and adaptive attackers who may learn to evade coherence monitoring.

Threshold calibration. The coherence thresholds from Thorarinson and Hensgen [2026] ($\Delta > 0.7$: healthy; $0.3 < \Delta < 0.7$: transition; $\Delta < 0.3$: failure) were validated on engineering systems (turbofan degradation, industrial valve monitoring) and biomedical signals (ECG arrhythmia detection). Blockchain transaction dynamics operate on different time scales — block intervals of 12–15 seconds on Ethereum vs. hours or days for mechanical degradation — and require independent calibration.

π_{key} implementation complexity. The key authorization operator is architecturally the most novel and practically the most challenging. Implementing continuous consent verification without compromising the pseudonymity and permissionlessness that are core blockchain properties requires careful design. Threshold signature schemes [Gennaro et al., 2016] address key distribution but not consent verification. Privacy-preserving liveness proofs are an active research area with no production-ready solutions.

Adversarial evasion. A sophisticated attacker aware of BCM monitoring could design exploits that maintain coherence across monitored dimensions while exploiting unmonitored ones. This is the fundamental limitation of any monitoring system: coverage is always incomplete. BCM mitigates this through cross-domain correlation (Σ_{risk}), making evasion more expensive (requiring simultaneous coherent behavior across multiple dimensions), but does not eliminate it.

Σ_{risk} aggregation. The optimal aggregation function for the composite risk score is an open problem. Weighted averaging risks masking a critical single-operator failure; minimum-based aggregation may generate excessive alerts from noisy operators. Learned aggregation functions require labeled incident datasets that are currently sparse.

Computational cost. Continuous monitoring of all six operators at block-level granularity is computationally expensive. Production deployment will require careful engineering to maintain monitoring coverage without introducing latency into the systems being monitored. Edge cases include high-throughput chains (Solana: ∼400 TPS), where per-transaction coherence computation may be infeasible.

8 Conclusion

The DeFi security problem is not a cryptography problem. The $7B+ in losses from 2020–2024 resulted from operational failures — compromised keys used with valid signatures, divergent cross-chain state, contextually dangerous contract interactions — that existing monitoring tools detect too late or not at all. BCM addresses the detection latency gap through continuous, multi-dimensional coherence monitoring.

The Ronin bridge reconstruction demonstrates the architectural argument: three BCM operators (π_{key}, Ω_{sync}, Ξ_{chain}) would have signaled degradation within hours, compared to the six-day actual detection gap. The $625M question — *why did no one notice for six days?* — has a structural answer: no monitoring system was continuously assessing the behavioral coherence of the validator set, the cross-chain asset parity, or the chain-state consistency of the bridge.

Production validation remains the critical next step. The testnet evaluation framework (Section 6) provides a reproducible protocol, and the false-positive rate challenge identified by Perez and Livshits [2021] — where 98% of flagged vulnerabilities were never exploited — must be met head-on. BCM must demonstrate that continuous coherence monitoring produces fewer, more actionable alerts than existing signature-based and forensic approaches.

The architecture is proposed as a complement to, not replacement for, existing cryptographic verification and forensic tools. Blockchains verify transactions. BCM monitors whether the system is still coherent enough for those verifications to mean what they appear to mean.

References

Chainalysis. The 2024 crypto crime report. Technical report, Chainalysis Inc., 2024. URL https://www.chainalysis.com/blog/2024-crypto-crime-report-introduction/.

Philip Daian, Steven Goldfeder, Tyler Kell, et al. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *IEEE Symposium on Security and Privacy*, pages 910–927, 2020. URL https://arxiv.org/abs/1904.05234.

Rosario Gennaro, Steven Goldfeder, and Arvind Narayanan. Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security. In *International Conference on Applied Cryptography and Network Security*, pages 156–174. Springer, 2016.

Lewis Gudgeon, Daniel Perez, Dominik Harz, Arthur Gervais, and Benjamin Livshits. The decentralized financial crisis: Attacking DeFi. *arXiv preprint arXiv:2002.08099*, 2020.

Sung-Shine Lee, Alexandr Murashkin, Martin Derka, and Jan Gorzny. SoK: Not quite water under the bridge: Review of cross-chain bridge hacks. In *IEEE International Conference on Blockchain and Cryptocurrency*, 2023.

Torgin Mackinga, Tejaswi Nadahalli, and Roger Wattenhofer. TWAP oracle attacks: Easier done than said? In *IEEE International Conference on Blockchain and Cryptocurrency*, pages 1–8, 2022.

Bruno Mazorra, Victor Adan, and Vanesa Daza. Do not rug on me: Leveraging machine learning techniques for automated scam detection. *Mathematics*, 10(6):949, 2022.

Muhammad Izhar Mehar, Charles Louis Shier, Alana Giambattista, Elgar Gong, Gabrielle Fletcher, Ryan Sanayhie, Henry M Kim, and Marek Laskowski. Understanding a revolutionary and flawed grand experiment in blockchain: The DAO attack. *Journal of Cases on Information Technology*, 21(1):19–32, 2019.

Daniel Perez and Benjamin Livshits. Smart contract vulnerabilities: Vulnerable does not imply exploited. In *30th USENIX Security Symposium*, pages 1325–1341, 2021.

Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. Attacking the DeFi ecosystem with flash loans for fun and profit. In *Financial Cryptography and Data Security*, 2021. URL https://arxiv.org/abs/2003.03810.

Joel Thorarinson and Allison Hensgen. From prediction to discoverative intelligence: A coherence-based AI framework for detecting system drift before failure. *arXiv preprint*, 2026.

Palina Tolmach, Yi Li, Shang-Wei Lin, Yang Liu, and Zengxiang Li. A survey of smart contract formal specification and verification. *ACM Computing Surveys*, 54(7):1–38, 2022. URL https://arxiv.org/abs/2008.02712.

Jiajing Wu, Jieli Liu, Weili Chen, Huawei Huang, Zibin Zheng, and Yan Zhang. Analysis of cryptocurrency transactions from a network perspective: An overview. *Journal of Network and Computer Applications*, 190:103096, 2021.

Mengya Zhang et al. SoK: Security of cross-chain bridges: Attack surfaces, defenses, and open problems. *arXiv preprint arXiv:2312.12573*, 2023. URL https://arxiv.org/abs/2312.12573.

Liyi Zhou, Xihan Xiong, Jens Ernstberger, et al. SoK: Decentralized finance (DeFi) attacks. *arXiv preprint arXiv:2208.13035*, 2022. URL https://arxiv.org/abs/2208.13035.

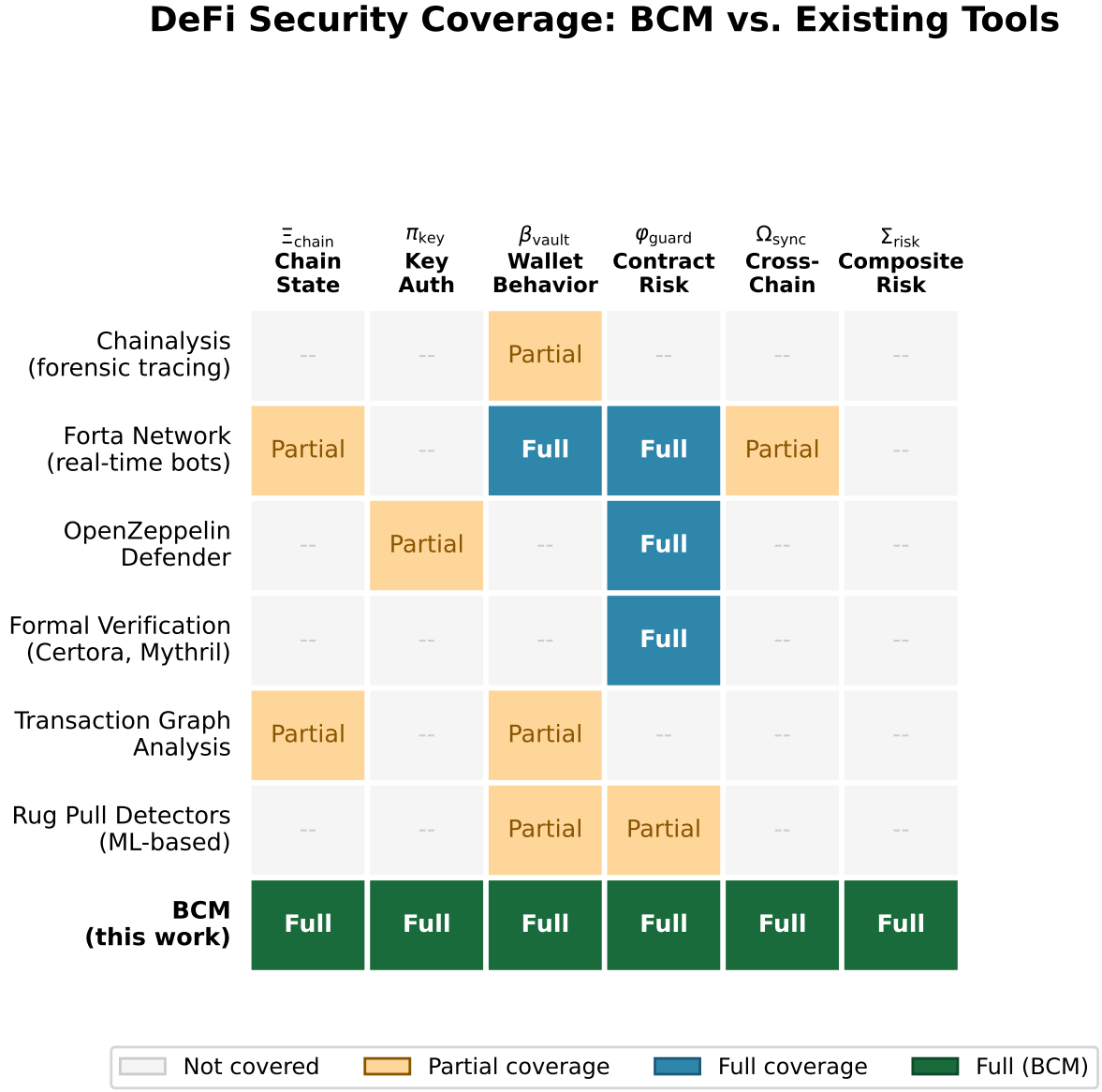

Figure 3: Coverage comparison of existing blockchain security tools across BCM operator dimensions. The key gaps in existing tooling are key authorization coherence (π_{key}), cross-chain synchronization (Ω_{sync}), and composite risk aggregation (Σ_{risk}). No single existing tool provides continuous monitoring across all dimensions.