TRIX (TriCoin) - LLD Technical Whitepaper

TRIX (TriCoin) - LLD Technical Whitepaper (Draft)

Version: Draft
Date: 2026-02-05 This is a low-level design (LLD) companion to the TRIX Introduction. It is intended for engineering/security review. Not investment advice. Not a delivery guarantee.

---1. Purpose and scope

TRIX is a concept for a three-pillar reference-value system:- W (Wellbeing) - localized environmental wellbeing signals- L (Local Activity) - local commerce signals- G (Global) - international market signals

Each contributes one-third to the reference value.

This technical whitepaper focuses on:- component boundaries and responsibilities- data contracts and provenance- oracle quorum, staking, and slashing- scoring/aggregation rules with stability guardrails- threat model, abuse detection, and failsafes

---2. Key principles- Deterministic computation: given the same approved inputs for an epoch, all validators/oracles compute the same indexes.- Multi-source by design: no single feed determines any component.- Privacy by default: prefer aggregated environmental data; avoid storing precise movement/location trails.- Auditability: commit cryptographic summaries to an immutable ledger.- Fail-safe operation: circuit breakers and bounded changes; graceful degradation.

---3. System architecture (LLD view)3.1 Subsystems

Subsystem	Responsibilities	Notes
Data sources	Environmental, commerce, and global market feeds	Provenance required
Oracle nodes	Fetch + validate + aggregate + sign	Stake + slashing
Scoring engine	Normalize, smooth, clamp, compute W/L/G	Versioned parameters
Ledger contracts	Store reference value and key parameters	Minimal surface
Observability	Monitoring, anomaly detection, IR	Off-chain allowed |3.2 Suggested event flow (epoch-based)

1. Collect measurements for epoch t
2. Validate provenance and quality
3. Aggregate per-region component indexes
4. Achieve oracle quorum + signatures
5. Publish index updates / roots
6. Compute/update reference value with guardrails

---4. Data contracts4.1 Canonical measurement record (suggested)- measurementId: UUID- source.provider: string- source.type: environment | commerce | global- source.uri: canonical reference to raw input- source.signature: signature over the referenced payload- region.regionId: stable region key- metric.name: e.g. pm25, noise, temp- metric.value, metric.unit, metric.quality- observedAt, ingestedAt: ISO timestamps4.2 Oracle submission record (suggested)- epoch: integer- regionId- component: W | L | G- inputsMerkleRoot: hash of raw inputs included- aggregate.index: float- aggregate.method: median | trimmed_mean |

weighted_average- signatures[]: quorum signatures

---5. Scoring and valuation5.1 Component computation- Wellbeing W- normalize each metric to a comparable unitless scale- weight by signal quality and source trust tier- aggregate per region (trimmed mean or median recommended)- Local activity L- derive LAI from local transaction volume and velocity- include verified merchant participation signals- avoid incentives that encourage wash trading (see abuse section)- Global G- aggregate multiple global feeds- clamp rapid changes to prevent volatility spikes5.2 Reference value

[math block omitted]5.3 Guardrails- clamp: per-epoch max delta per component- smooth: EMA or rolling median- outliers: IQR/z-score trimming- caps: optional global cap for G drift per window

---6. Oracle integrity model- quorum-based acceptance (e.g. 7-of-10)- oracle staking + slashing for provable misconduct- input provenance: each feed must be attributable- transparency: publish oracle set membership and historical performance metrics

---7. Threat model & abuse detection7.1 Expected attack classes- oracle manipulation / collusion- fake sensor inputs / spoofed environmental signals- wash trading to inflate local activity- geofence spoofing / impossible travel patterns- denial-of-service on indexers or APIs7.2 Controls- stake-backed oracle sets- multi-source reconciliation- anomaly detection on index movements and transaction patterns- rate limits and circuit breakers- investigative workflow + appeal where applicable

---8. Failsafes- circuit breaker for anomaly spikes- rollback to last known good parameters- freeze/rotate oracle sets under incident response- checkpointing and audit trails

---9. Appendix: implementation checklist- finalize region taxonomy + baseline model- define metric normalization functions- specify LAI and anti-wash-trade rules- implement oracle signing + quorum + slashing- implement guardrails + circuit breakers- schedule independent security audits