

## v3.4-SEC-003 – Twin Integrity, Audit Trails & Notarization

Document Title	Twin Integrity, Audit Trails & Notarization
Version	v3.4
Document ID	v3.4-SEC-003
Date	2025-03-22
Author	Take Back Your Data – Audit & Compliance Unit
Document Type	Public / Certification / Internal

### 1. Purpose & Scope

This document defines the mechanisms used to ensure runtime integrity, provide audit trails, and support notarization of MaxOneOpen twin-based executions. It guarantees traceability and forensic analysis across sovereign infrastructures.

### 2. Twin Integrity Model

- Each twin runtime is signed at launch (code, config, hash)
- Any deviation invalidates the runtime and triggers fallback
- Integrity must be validated locally and logged
- No execution without signature-match verification

### 3. Audit Trail Design

Event Type	Record Format	Verification
Twin Launch	Signed metadata header	Hash match + source check
Runtime Decision	Token + prompt hash	ZK-compatible proof
Quota Use	Token/sec + task ID	Quota policy enforcement
Termination	Exit code + memory hash	End-state notarization

### 4. Notarization Logic

- Every critical action emits a notarization pulse
- Pulses are signed locally and optionally broadcasted to mesh validators
- Forks must support notarization interfaces or bridge wrappers
- Notarization does not log content – only cryptographic traces

### 5. Compliance & Traceability

- All forks must expose audit trail endpoints
- Twin memory and runtime identity must be loggable
- Logs must be cryptographically immutable
- Certification requires random forensic reproducibility tests

## 6. Certification Relevance

Integrity assurance, audit trails and notarization are mandatory for MaxOneOpen compliance. Deployments must support full lifecycle traceability and runtime identity validation, with or without external connectivity.