

v3.4-PRT-003 – Cross-Hardware Consistency & Migration Audit Logic

Document Title	Cross-Hardware Consistency & Migration Audit Logic
Version	v3.4
Document ID	v3.4-PRT-003
Date	2025-03-22
Author	Take Back Your Data – Portability & Mobility Group
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines the requirements and verification logic for maintaining twin consistency across heterogeneous hardware and auditing twin migrations within MaxOneOpen. It ensures auditability, reproducibility and determinism across environments.

2. Cross-Hardware Execution Guarantees

- Runtime must produce identical outputs across certified architectures
- Any drift must trigger revalidation, quarantine or fallback
- Critical operations must be deterministic and stateless during comparison
- Forks must expose proof of runtime parity across device types

3. Migration Audit Trails

Migration Phase	Required Artifacts	ZK Verification Hook
Pre-Export	Snapshot manifest, schema lock	Proof of context integrity
Handoff	Signed export hash, runtime token	Keypair trace + lineage log
Rebind	Twin init + runtime diff	Fallback trigger + audit replay
Post-Execution	Delta map + output hash	Schema fingerprint verification

4. Certification Hooks

- Cross-hardware forks must pass reproducibility tests
- All runtime logs must be ZK-verifiable and anchored
- Hardware abstraction deltas must be declared and reviewable

5. Certification Triggers

- Drift, log mismatch or audit gaps disqualify fork
- Undeclared abstraction paths or runtime divergence are non-compliant

6. Certification Relevance

To qualify for MaxOneOpen certification, forks must demonstrate cross-hardware consistency and maintain full audit trails for all twin migrations. Deterministic, verifiable execution is mandatory across environments.