

## v3.4-ZKP-003 – Proof Chaining, Nested Validation & Zero-Knowledge Fork Integrity

Document Title	Proof Chaining, Nested Validation & Zero-Knowledge Fork Integrity
Version	v3.4
Document ID	v3.4-ZKP-003
Date	2025-03-22
Author	Take Back Your Data – Cryptographic Trust Group
Document Type	Public / Certification / Internal

### 1. Purpose & Scope

This document defines the architecture and logic of proof chaining, nested validation and fork integrity in MaxOneOpen. It ensures that all forks maintain verifiable ZK-histories, composable proof structures and untampered execution lineage.

### 2. Proof Chaining Design

- All ZK-proofs must be chained across twin lifecycle stages
- Each proof step anchors the previous context and output
- No external state or opaque references permitted in the chain
- Chains must support full recursive replay and context sealing

### 3. Nested Validation Logic

Validation Type	Target Layer	Rejection Trigger
Context Replay	Twin execution stage	Mismatch in schema hash
Runtime Trace	Inference or storage	Logic path divergence
Snapshot Seal	Fork merge or import	Invalid proof origin
Audit Chain	Cross-fork verification	Missing proof depth

### 4. Certification Hooks

- Proofs must chain across all lifecycle actions and runtime states
- Forks must expose recursive proof structure and nesting depth
- Audit chains must allow revalidation by independent peers

### 5. Certification Triggers

- Broken chains, partial proof coverage or black-box references disqualify fork
- Missing context anchors or non-replayable logic is non-compliant

### 6. Certification Relevance

Forks must demonstrate consistent proof chaining and integrity across the entire lifecycle. Only forks with recursively verifiable, context-bound chains qualify for MaxOneOpen certification.