

v3.4-ZKP-002 – ZK-Driven Audit Trails, Identity Blinding & Compliance Proofs

Document Title	ZK-Driven Audit Trails, Identity Blinding & Compliance Proofs
Version	v3.4
Document ID	v3.4-ZKP-002
Date	2025-03-22
Author	Take Back Your Data – ZK Sovereignty Lab
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines the use of zero-knowledge logic for creating audit trails, identity blinding, and compliance validation in MaxOneOpen. It ensures that traceability and legal alignment are preserved without sacrificing privacy or sovereignty.

2. ZK-Audit Trail Mechanism

- All forks must implement cryptographically verifiable audit trails using ZK anchors
- Audit trails must support full replayability without revealing identities or raw data
- Event traces must include circuit output, policy match, and sealed state transitions
- Anchors must be checkpointed and tamper-proof across node states

3. Identity Blinding & Compliance Architecture

ZK Layer	Function Scope	Certification Constraint
Blind Anchor Hash	Obfuscate source identity	Signature unlinkability
Event Chain Seal	Integrity path binding	Circuit + hash path match
Policy Proof Set	Compliance check traces	Schema + logic constraint
Forensic Replay	Trace snapshot without exposure	Entropy-safe reconstruction

4. Certification Hooks

- Audit trails must be replayable and ZK-sealed at each step of twin logic
- Identity visibility must be cryptographically blinded across all state transitions
- Compliance circuits must be reusable for audit agencies without data leakage

5. Certification Triggers

- Forks that expose real identity or use plain audit trails are disqualified
- Opaque traces, incomplete chains or unverifiable state transitions invalidate certification

6. Certification Relevance

All certified forks of MaxOneOpen must embed privacy-preserving audit trails with ZK-blinded identity mechanisms. Sovereignty requires proof without exposure – and validation without compromise.