

v3.4-ZKP-003 – Twin-Certified ZK Contracts & Deterministic Policy Enforcement

Document Title	Twin-Certified ZK Contracts & Deterministic Policy Enforcement
Version	v3.4
Document ID	v3.4-ZKP-003
Date	2025-03-22
Author	Take Back Your Data – ZK Sovereignty Lab
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines the architecture and requirements for twin-certified zero-knowledge contracts and deterministic policy enforcement in MaxOneOpen forks. It ensures execution certainty, sealed governance logic, and verifiable compliance under sovereign conditions.

2. Twin-Certified ZK Contract Logic

- All smart contracts must be cryptographically signed by execution twins
- Contracts must expose sealed ZK circuits for audit, replay, and scope validation
- Twin signatures must include key binding to policy path, role map, and version control
- ZK logic must remain reusable and match lifecycle traceability standards

3. Deterministic Policy Enforcement

Enforcement Layer	Policy Role	ZK Verification Anchor
Policy Engine	Match trigger to contract	Schema path + logic seal
Twin Verifier	Certify execution twin	Signature + TTL proof
Action Capsule	Lock function scope	Execution hash + proof set
Override Watcher	Validate fallback/revoke	Anchor log + trace delta

4. Certification Hooks

- Forks must bind contracts to twin signatures and verify policy compliance via ZK logic
- No fork may execute runtime functions outside sealed policy-scope
- All fallback, revoke, and emergency paths must be pre-certified

5. Certification Triggers

- Contracts without twin certification or outside policy scope disqualify forks
- Dynamic policy override without audit trace invalidates certification

6. Certification Relevance

Only forks with twin-bound, ZK-certified contracts and deterministic policy logic are eligible for MaxOneOpen certification. Execution must be verifiable, auditable, and sealed within trusted role-control paths.