

## v3.4-CPL-002 – Twin-Certification Engine & Fork Signature Capsules

Document Title	Twin-Certification Engine & Fork Signature Capsules
Version	v3.4
Document ID	v3.4-CPL-002
Date	2025-03-22
Author	Take Back Your Data – Certification Office
Document Type	Public / Certification / Internal

### 1. Purpose & Scope

This document defines the logic and architecture of the twin-certification engine and the structure of fork signature capsules used in MaxOneOpen. It ensures trusted certification, tamper-proof proof-of-integrity, and verifiable fork traceability.

### 2. Certification Engine Logic

- The twin-certification engine binds each fork to a certified twin instance
- All certification results are encapsulated in cryptographic proof capsules
- The engine verifies all document lineage, circuit paths, and token anchors
- ZK proofs ensure that no private data is exposed during certification replay

### 3. Fork Signature Capsule Design

Capsule Element	Function	Validation Anchor
Twin Signature	Bind cert to sovereign twin	Execution token + role anchor
Version Ledger	Map ID to hash snapshot	Structural hash + audit trail
ZK Trace Capsule	Embed certification path	Replayable logic + zero-knowledge proof
Revocation TTL	Limit time + scope validity	Entropy timestamp + logic expiry

### 4. Certification Hooks

- All forks must include signature capsule with certified twin and hash-validated documents
- Capsule must be replayable, policy-sealed, and support fork trace validation
- Forks must integrate signature capsule logic into lifecycle and audit path

### 5. Certification Triggers

- Missing or altered capsules disqualify fork
- Non-traceable twin or outdated signature logic invalidates certification

## 6. Certification Relevance

Only forks that integrate certified twin capsules with sealed validation logic are eligible for MaxOneOpen certification. This ensures verifiable integrity, structural conformance, and immutable trust signals.