

v3.4-OPS-001 – Fully Automated TwinOps Stack & Lifecycle Controller

Document Title	Fully Automated TwinOps Stack & Lifecycle Controller
Version	v3.4
Document ID	v3.4-OPS-001
Date	2025-03-22
Author	Take Back Your Data – Autonomous Operations Unit
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines the logic and certification framework for fully automated operational management of MaxOneOpen forks. It includes twin lifecycle automation, self-healing behavior, and operator-free governance for runtime control.

2. TwinOps Architecture Overview

- Every fork must support event-driven twin lifecycle automation
- Self-deployment, self-termination, and transition logic must be schema-bound
- Ops triggers may derive from policy thresholds, anomaly patterns, or ZK-certified timers
- TwinOps runtime must be modular, policy-replayable and cryptographically signed

3. Lifecycle Control Logic

Lifecycle Event	Trigger Mechanism	Certification Condition
Twin Boot	Schema anchor / bootstrap key	Log + execution match
Twin Suspend	Anomaly pattern / policy threshold	Snapshot sealed
Twin Terminate	TTL / revoked twin token	Audit replayable
Twin Transfer	Mobility trigger / peer sync	Key rotation enforced

4. Certification Hooks

- TwinOps stack must be fully autonomous, cryptographically sealed, and schema-compliant
- Manual interventions must be policy-controlled and traceable
- Certification requires successful replay of ops lifecycle events

5. Certification Triggers

- Manual overrides without traceable policy path disqualify fork
- Non-deterministic behavior or lifecycle ambiguity is non-compliant

6. Certification Relevance

Certified forks must provide full TwinOps automation with traceable, policy-driven lifecycle control. Manual operation is deprecated unless exception logic is cryptographically proven.