

v3.4-SEC-005 – Breach Containment, Twin Failover & Self-Healing Security Layers

Document Title	Breach Containment, Twin Failover & Self-Healing Security Layers
Version	v3.4
Document ID	v3.4-SEC-005
Date	2025-03-22
Author	Take Back Your Data – Security Resilience Engineering
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines breach containment protocols, twin-level failover strategies and self-healing layers for sovereign MaxOneOpen deployments. It ensures runtime resilience, continuity and cryptographic boundary integrity under fault or attack conditions.

2. Breach Containment Logic

- Breach triggers: anomaly, signature mismatch, token hijack
- Containment zones: runtime, schema, peer session
- Active forks must isolate and log breach source in real-time
- ZK logs must reflect cause, impacted entities, and quarantine state

3. Twin Failover Strategy

Failover Mode	Activation Trigger	Continuity Mechanism
Live Mirror	Twin heartbeat lost	Snapshot reload + new token
Quorum Activate	Majority peer report	Signed redeploy trigger
Self-Promote	Primary twin breached	Auto-elevation with ZK log
Manual Override	Admin key injection	User-flagged reinit only

4. Self-Healing Security Patterns

- Forks must implement at least one recovery-capable runtime layer
- Healing may not regrant access without proof-of-clear
- Healing logs must be verifiable and publicly provable on request
- No healing may occur outside schema-signed execution path

5. Certification Triggers

- Forks must simulate at least two breach scenarios
- Certification logs must show full containment lifecycle
- Gaps in failover path or healing logic disqualify certification

6. Certification Relevance

All certified MaxOneOpen forks must include breach containment, verifiable twin failover logic and self-healing security mechanisms. Continuity, isolation and log integrity are essential to certification.