

v3.4-OPS-001 – MaxOps Orchestration, Monitoring & Execution Stack

Document Title	MaxOps Orchestration, Monitoring & Execution Stack
Version	v3.4
Document ID	v3.4-OPS-001
Date	2025-03-22
Author	Take Back Your Data – Sovereign Ops Division
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines the orchestration, execution control and monitoring components within the MaxOps stack. It ensures sovereign forks are fully automatable, observable and lifecycle-controllable without reliance on centralized platforms.

2. Architecture & Components

- MaxOps coordinates execution across all active twins and services
- The orchestration kernel supports event-based, trigger-based and timer-based flows
- Monitoring agents track runtime, data, inference, token and network states
- Faults and exceptions are routed via self-healing controller to fallback twins

3. Orchestration Layer Capabilities

Capability	Scope	Trigger Source
Service Meshing	Across twins & edge nodes	Twin manifest
State Rebind	Context sync & hot-restart	Runtime fault detect
Execution Rollback	Request undo & shadow fork	Error policy
Auto-Scaling	Load-driven twin activation	Performance watch

4. Monitoring & Observability Stack

- Every fork must expose internal state changes to twin audit ports
- Monitoring stack includes latency, throughput, error traces and schema diff
- Dashboards are auto-generated from twin logs (ZK-anchored)
- Fork maintainers must validate self-repair logs upon event trigger

5. Certification Triggers

- Certification requires full orchestration graph & twin activation trace
- Forks lacking rollback logic or observable control flow are disqualified
- Monitoring logic must cover all critical execution paths and be replayable

6. Certification Relevance

Sovereign MaxOneOpen forks must include MaxOps or equivalent orchestration stack to qualify for certification. Forks without structured execution control, monitoring visibility or rollback capability are non-compliant.