

v3.4-EXE-001 – Execution Strategy & Deployment Typology

Document Title	Execution Strategy & Deployment Typology
Version	v3.4
Document ID	v3.4-EXE-001
Date	2025-03-22
Author	Take Back Your Data – Deployment Engineering Unit
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines the execution strategy and possible deployment topologies of MaxOneOpen. It ensures that all certified deployments follow a sovereign, reproducible, and scalable structure aligned with UDUH and modular control principles.

2. Execution Logic Overview

- Execution is always context-triggered, never idle-persistent
- All active logic is launched by MaxControl or validated external signals
- Twin logic dictates specialization and redundancy path
- Inference containers are sandboxed, short-lived, and quota-bound

3. Deployment Typology

Type	Location	Target Group	Characteristics
Sovereign Node	On-premise / DMZ	Governments, critical infra	Fully isolated, certified modules only
Edge Runtime	Local/remote edge	Decentralized inference	Lightweight, containerized, adaptive
Federated Cluster	Multiple trusted peers	Interconnected orgs	Mesh logic, fallback, joint trust
Single-Device	Laptop, mobile, IoT	Offline / micro deployments	Local model inference with policy enforcement

4. Control Layer Execution Role

- All deployments must integrate MaxControl or compatible scheduler
- Control Layer activates, rotates and terminates runtime twins
- Execution traces are hashed and optionally notarized
- No deployment may execute outside control signal bounds

5. Fork Deployment Requirements

- Forks must declare deployment type(s) they support
- All execution logic must follow certified topology behavior

- Deployment documentation must include control mapping and fallback logic
- Deviations require certification extension and test results

6. Certification Relevance

Each certified deployment must conform to one or more defined typologies. Execution must be auditable, bounded, and align with MaxOneOpen's dynamic twin logic. Unverified deployment models are considered non-compliant.