

v3.4-AI-003 – Model Forking, Containment & Schema Isolation

|                |   |
|----------------|---|
| Document Title | Model Forking, Containment & Schema Isolation   |
| Version        | v3.4  |
| Document ID    | v3.4-AI-003                                     |
| Date           | 2025-03-22                                      |
| Author         | Take Back Your Data – Model Sovereignty Systems |
| Document Type  | Public / Certification / Internal               |

1. Purpose & Scope

This document defines MaxOneOpen mechanisms for forking, containing, and isolating AI models within schema-bound execution. It ensures sovereign control over forked logic and eliminates execution leakage or uncontrolled model inheritance.

2. Forking Rules & Boundaries

- Forked models must generate a new ID and hash lineage
- Schema inheritance must be explicit and declared
- Forked logic must remain execution-sandboxed
- Fork eligibility tied to certification tier and role scope

3. Containment Logic

| Containment Mode | Execution Constraint          | Trigger Condition     |
|------------------|-------------------------------|-----------------------|
| Soft Contain     | Local-only, scoped to task    | Uncertified fork      |
| Hard Contain     | Isolated VM with sealed vault | Shadow model class    |
| Policy Guarded   | Scoped to schema branch       | User-bound override   |
| ZK-Enforced      | Proof-driven containment      | Runtime signal breach |

4. Schema Isolation Design

- Models must be schema-bound: structure + runtime profile
- Execution outside schema scope must be blocked
- Each schema defines data fields, tokens, rights, twin hooks
- Forks may add logic but not override base schema without versioning

5. Audit Trail & Runtime Logs

- All forks must emit runtime logs: init, exec, result hash
- Contained runs must be flagged in result metadata
- Isolation logs must be exported during certification review
- Certification failure occurs on log tampering or omission

## 6. Certification Relevance

Forking and containment enforcement is critical to MaxOneOpen certification. Only schema-bound, traceable forks may be executed within sovereign runtime environments.