

v3.4-STK-004 – APIs & Model Interaction Logic

Document Title	APIs & Model Interaction Logic
Version	v3.4
Document ID	v3.4-STK-004
Date	2025-03-22
Author	Take Back Your Data – API Integration Unit
Document Type	Public / Certification / Internal

1. Purpose & Scope

This document defines the API logic, access structure, and interaction principles between MaxOneOpen modules and external systems. It ensures secure, efficient, and verifiable control of LLM interactions under sovereign conditions.

2. API Types & Access Modes

- **MaxAPI (Core Interface):** Governs model interaction, twin spawning, and tokenization control
- **TwinAPI:** Activates, configures, and terminates twin modules
- **ControlAPI:** Secure endpoint for governance, ZKP and quota tracing
- **Custom APIs:** Optional modules that extend use-case interaction

3. Invocation Models

- **Direct Call (Synchronous):** Real-time processing via MaxAPI
- **Asynchronous Task Chain:** Multi-stage requests with twin coordination
- **Passive Monitoring Hooks:** Event-driven twin activation or logging trigger
- **External Signal Binding:** Hardware or network-level control integration

4. API Governance Logic

Each API interaction follows a zero-trust, cryptographically validated flow:

- Each endpoint is stateless and tied to a specific permission token
- Requests are signed and hashed against access policy
- Non-authorized interaction triggers rollback or shadow clone review
- Forked APIs must register schema & control interface logic

5. API Certification Criteria

Criterion	Requirement	Verifiability
Schema Clarity	OpenAPI 3.1 or equivalent	Yes
Permission Control	Token-based + policy reference	Yes
Twin State Logic	Must define twin state transitions	Yes
Fallback Safety	Rollback or containment path	Yes

Zero Data Exposure	No API may expose raw input or system memory	Yes
--------------------	----------------------------------------------	-----

6. Certification Relevance

All certified forks must expose APIs that comply with MaxOneOpen's zero-trust interaction model. Any deviation, abstraction, or expansion must be auditable, formally defined, and cryptographically validated.