

**v3.4-STK-002 – Inference Stack & Runtime Design**

Document ID: v3.4-STK-002 | Status: Final | Version: v3.4

Date: 2025-03-22

Author: Take Back Your Data – Runtime Core Unit

Document Type: Public / Certification / Internal

**1. Purpose & Scope**

This document defines the structure, components, and execution logic of the MaxOneOpen inference stack. It enables technical teams to deploy optimized runtime environments tailored to specialized twin modules.

**2. Inference Components**

- MetaLLM: Specialized model optimized for modular control
- Tokenizer / Detokenizer: Pluggable, hardware-adaptive
- Prompt Assembly Layer: Structuring system input with minimal latency
- Vector Handler: Optional embedding generation for memory/search
- Inference Core: Low-latency transformer execution unit
- Twin Execution Wrapper: Dynamic container, instantiates specialized twins

Versioning Matrix:

Component	Version	Update Path	Maintainer Role
MetaLLM	v1.2.0	Stable API	LLM Core / Runtime Core
Tokenizer / Detokenizer	v0.9.3	Modular Swap	Twin Integrator
Prompt Assembly Layer	v1.1.0	Fork-per-Context	Flow Maintainer
Inference Core Engine	v1.0.5	Platform-Linked	Hardware Abstraction

**3. Execution Flow**

1. Input (user/system) is routed via prompt logic layer
2. Active twin container is spun up for relevant context
3. MetaLLM receives preprocessed token stream

4. Output is delivered back to system, user or logger
5. Process terminates or remains latent based on context policy

#### 4. Runtime Optimization

- Models are optimized for context-short hops (low token window)
- Edge-first fallback minimizes inference delay
- Vector memory is optional and not required for functional use
- Hardware acceleration modularity (GPU/TPU/NPU/CPU/FPGA)
- Runtime logic is layered: init → execute → terminate/passivate

#### 5. Inference Twin Typology

Twin Type	Purpose	Lifecycle
----- ----- -----		
Static Twin	Fixed-purpose, cached	Preloaded → Execute → Idle
Dynamic Twin	Task/context-specific	Spin up → Execute → Terminate
Shadow Twin	Observational/feedback	Activated via Control Layer
Failover Twin	Redundancy/fallback	Cold start on trigger event

#### 6. Certification Relevance

All certified MaxOneOpen forks must implement the inference stack using the layered logic, typology and lifecycle models defined here. Any deviation must be documented and justified within the certification request.