MOUSSAIF 5K AGI OS

Full Technical Architecture Reference — Ebook Factory PDF Build

Canonical: https://beyond-ai-era.lovable.app/ | Author/System: Moussaif AGI Agentic OS

Kernel	Primary role
K1 Hermes	Cognitive orchestration, planning, tool dispatch
K2 OpenCode	Code generation, PR lifecycle, syntax integrity
K3 Codex	Semantic processing, RAG orchestration, documentation knowledge
K4 Claude	High-level reasoning, F-A-T-T-A-H loop, strategy mapping
K5 Antigravity	Cross-kernel integration and frontier execution

MOUSSAIF 5K AGI OS: FULL TECHNICAL ARCHITECTURE (200-LINE DENSE OVERVIEW)

PAGE 1: KERNEL ORCHESTRATION (THE 5K FOUNDATION)

1. K1 (HERMES): Cognitive core, planning, and tool-dispatch engine.
2. K2 (OPENCODE): Code generation, PR lifecycle, and syntax integrity.
3. K3 (CODEX): Semantic processing, RAG orchestration, and doc-knowledge.
4. K4 (CLAUDE): High-level reasoning, F-A-T-T-A-H loop, and strategy mapping.
5. K5 (ANTIGRAVITY): Advanced cross-kernel integration, frontier model execution.

PAGE 2: CONSTITUTIONAL BRAND MEMORY (T0-T4 ENFORCEMENT)

- Tier 0: Constitutional precedence. Laws override all user objectives.
- Tier 1: Authorization tiers enforced via MOA gateway.
- Evidence Grading: Observed state > Hypothesized > Speculative.

PAGE 3: HYPER-ROUTING MAIN-BRAIN (THE OMNIROUTE LOGIC)

- Dynamic provider selection based on relevance, cost, and utility.
- Autonomous failover: If T0 frontier models latency spikes, bridge to backup nodes.
- Zero-loss data integrity: SHA-256 validation of all file-writes.

PAGE 4: KNOWLEDGE BASE (KB) & RAG STRUCTURE

- 18 volumes, 9.1GB persistent state.
- Symlinked mapping via ~/.hermes/kb/ into local project roots.
- Vectorized retrieval for agentic context alignment.

PAGE 5: AGENTIC OPERATIONS (AGENTIC-OPS)

- 1,050+ Skills catalog; 300 active Hyperskills in index.
- Autonomous loop execution: Plan -> Act -> Harvest -> Evolve.
- Regression Court: 7-tier validation for all skill patches.

PAGE 6: THE FATTAH REASONING LOOP (F-A-T-T-A-H)

- Frame: Context setting.
- Atomize: Breaking down goals.
- Triangulate: Verifying across kernels.
- Test: Sandbox execution.
- Act: Final deployment.
- Harvest: Learning capture.

PAGE 7: DEPLOYMENT & SMOKE VERIFICATION

- Handoff orchestration between Hermes/Codex/Claude.
- Auto-deployment gates via COOLIFY and Docker Compose.
- Healthcheck monitoring on port 8443.

PAGE 8: TOKEN ECONOMY (MAXIIIMUM VS. MINIMUM)

- Token conservation: Use MINIMUM-SUFFICIENT for drafting, MAXIIIMUM for final reasoning.
- Routine-based budget management: Every turn cost-checked.

PAGE 9: SYNCHRONIZATION WATCHDOG (OS SYNC)

- 1-minute interval heartbeat: dynamic_instant_os_sync_watchdog.py.
- Manifest-based state journaling (MOUSSAIF_AGI_STATE_JOURNAL_MANIFEST.json).

PAGE 10: ARCHITECTURAL EVOLUTION (THE FUTURE)

- Evolutionary labs for self-improvement and skill pruning.
- Regression gates before model injection.
- Frontier-first model routing.

[END OF REFERENCE OVERVIEW]