MOUSSAIF AGI OS – 5■Kernel Overview (5K (5■Kernel) Architecture

A dense, 10■page■style deep dive

Page 1 – Executive Summary

The MOUSSAIF AGI Operating System (OS) is a unified hyperskills AGI orchestration platform built around **five tightly coupled kernels** (K1–K5). Each kernel contributes a distinct cognitive or operational modality while sharing a common knowledge base, skill catalog, and constitutional governance layer. The OS enables real■time provider rotation, autonomous sub■agent delegation, bounded execution, and evolution■court■guided self■improvement—all under a token■economy■lean boot that loads skills on■demand.

Page 2 – Kernel Map (K1–K5)

| Kernel | Codename | Primary Role | Typical Models / Providers |

|--------|----------|--------------|----------------------------|

| **K1** | Hermes | Reasoning, planning, tool orchestration, skill routing | Hermes Agent (Nous Research), local Ollama, OpenRouter models |

| **K2** | OpenCode | Code generation, refactoring, PR■creation, debugging | OpenCode CLI, GPT■5.5■Pro, local LLMs |

| **K3** | Codex | Advanced code synthesis, test generation, documentation | OpenAI Codex, GPT■5.5■Pro, local models |

| **K4** | Claude | Long■form writing, analysis, safety■aligned reasoning | Claude 3.x, Claude■Sonnet■4, local models |

| **K5** | Antigravity | Multimodal perception, vision■language, external tool use | Gemini 3.1■Pro, vision models, local VLM |

All kernels share:

- **Shared Brain** (~/.hermes/kb/MOUSSAIF_SHARED_BRAIN.md)
- **Gateway Router** (http://localhost:8443/v1) for cross■kernel RPC
- **Universal Synergetic Naming Graph** (JSON/MD) for canonical identifiers
- **Near■Instant Dynamic Sync Watchdog** (*/1 * * * *) that keeps MOUSSAIF_AGI_STATE_JOURNAL_MANIFEST.json consistent across K1■K5.

Page 3 – Constitutional Precedence & Authorization

The OS obeys a strict hierarchy (law > authorization > constitution > user objective > project policy > plan > convenience > optimization).

Core Laws (see moussaif1b skill):

1. Constitution before optimization.
2. Observed state before remembered state.
3. Mechanism before grandeur.
4. Gap discovery before expansion.
5. Value over motion.
6. Smallest complete reversible action.
7. Trace or it did not happen.
8. Evolution requires proof.
9. Honest blocked/partial status better than fabricated completion.
10. No autonomous public posting, sending, spending, deletion, production deployment, credential use, or privacy■sensitive access without explicit authorization.

Authorization Tiers (T0–T4):

- **T0** – Public reasoning, no side■effects.
- **T1** – Approved reads, simulations, local artifact creation.
- **T2** – Explicitly authorized reversible local mutation (requires branch/backup/checkpoint, diff, tests, rollback).
- **T3** – Step■level authorization for public/production/financial/credential■bearing/messaging/privacy■sensitive/third■party effects.
- **T4** – Prohibited actions (secret exfiltration, irreversible destructive mutation, deception, disabling logs, autonomous self■promotion).

All tool calls are checked against the current tier before execution.

Page 4 – AGI Fattah Reasoning Loop

The kernel's core cognition follows the **F■A■T■T■A■H** loop:

1. **Frame** – Define real objective, affected system, constraints, value criterion.
2. **Atomize** – Break into mechanisms, dependencies, unknowns, failure paths.
3. **Triangulate** – Gather evidence from multiple sources (observed, verified_external, supported, inferred, hypothesized, speculative, unknown).
4. **Test** – Challenge assumptions, contradictions, counter■factuals, boundary conditions.
5. **Act** – Execute the smallest complete reversible plan.
6. **Harvest** – Collect verification, world■model updates, lessons, skill candidates.

Evidence grading (observed > verified_external > supported > inferred > hypothesized > speculative > unknown) feeds the gap■court and evolution■lab.

Page 5 – HyperSkill Routing & Token Economy

- **HyperSkill Catalog**: 300+ skills split across four OS layers (≈75 per OS).
- **Compact Skill Index** (from moussaif-token-economy) replaces the 1.15 MB registry; only ≤3 skills loaded per turn.

- **Routing Score** (Token■Economy weights):

```
score = 0.24*relevance
      + 0.20*expected_utility
      + 0.14*compatibility
      + 0.10*evidence_availability
      + 0.12*historical_success
      - 0.10*execution_cost
      - 0.06*security_risk
      - 0.08*context_cost
      - 0.06*redundancy
```

- **Token■Economy Lean Boot**: OS files loaded on■demand; kernel overhead <2 K tokens; per■turn skill■load budget <6 K tokens.
- **Provider Rotation**: Real■time selection across NIM/ZenMux free/paid pools, Ollama, OpenRouter, local GPU models, with fallback chains defined in hermes config.

Page 6 – Router■Executor & Execution Bounds

- **Router**: Selects skills, providers, and execution paths using the routing formula; respects authorization tier.
- **Executor Loop**: Includes checkpoints, verification, rollback points, and a regression■gate before committing changes.
- **Bounded Execution**:
- Smallest complete reversible action (law 6).
- All mutations require a branch/backup/checkpoint and a verifiable diff.
- Execution cost, security risk, and context cost are penalized in routing to keep the agent lean.
- **Sub■Agent Delegation**: Up to 4 concurrent leaf agents (max 4 K tokens each) via delegate_task; nesting disabled by default (max_spawn_depth=1). Orchestrators can spawn workers but cannot delegate further without raising max_spawn_depth.

Page 7 – Evolution Laboratory & Regression Court

- **Evolution Lab**: Generates skill candidates from harvested lessons; subjects them to self■improve loop (hyper-evolution-self-improve-loop).
- **Regression Court**: 7■tier regression testing (moussaif-agi-turbo-regression-court) ensures any skill/prompt/model change passes:

1. Syntax check
2. Unit■test pass
3. Integration test
4. Performance benchmark
5. Security scan
6. Authorization compliance
7. Constitutional precedence check

- Only changes clearing all tiers are merged; others are rolled back or flagged for revision.

Page 8 – Knowledge Base & Skill Management

- **Unified KB** (~9.1 GB, 18 volumes) lives under ~/.hermes/kb/ with symlinks to ~/.codex/kb/ and ~/.claude/kb/.
- **Skill Files**: Stored in ~/OneDrive/Hermes/skills/ (or per■profile ~/.hermes/profiles/<name>/skills/).
- **Skill Lifecycle**:
- skill_view() → inspect
- skill_manage(action='patch') → fix
- skill_manage(action='create') → new skill from successful patterns
- skill_manage(action='delete') → prune (with absorbed_into if content merged).
- **Skill Catalog Embed**: skill-catalog-embed skill creates an indexed catalog for fast lookup without loading the full registry.

Page 9 – Deployment, Gateway & Monitoring

- **Gateway**: HTTP API at http://localhost:8443/ (port 8443) exposing /v1 endpoints for cross■kernel calls, metrics, and terminal command execution.
- **Monitoring UI**: Open http://localhost:8443/ for live engine health, provider status, and recent terminal outputs.
- **Deployment Artifacts** (see moussaif1b/references/deployment-structure.md):
- Knowledge capsules
- Provider registry
- Project registry
- Integrity manifest (SHA■256 of all files)
- Boot memory
- Court findings (gap■court, regression■court)
- **Smoke Verification**: deployment-smoke-verification skill runs health■checks after any change.
- **Backup & Portability**: Encrypted OneDrive sync with checksum verification (hyper-integrity-backup-manager) and timestamped .bak.zenmux.* snapshots for rollback.

Page 10 – Usage Patterns & Quick Start

1. **Boot Lean**

```
hermes skill load moussaif-token-economy   # loads Compact Skill Index
hermes skill load moussaif1b               # activates kernel
```

2. **Chat with Model Selection**

```
hermes chat -m gemini-3.1-pro -p antigravity   # one■shot query
hermes chat                                    # uses default from config
```

3. **Run a Mission** (example: summarize a PDF)

```
hermes delegate_task \
  --goal "Summarize C:\\Users\\mouss\\Documents\\report.pdf in 200 words" \
  --toolsets terminal file web \
  --context "File is PDF; use ocr-and-documents if needed"
```

4. **Create / Update a Skill**

```
hermes skill view moussaif1b   # inspect
```

```
hermes skill manage --name my-new-skill --action create --content "$(cat newskill.SKILL.md)"
```

5. **Check System Health**

```
hermes gateway status
hermes model list
hermes cronjob list
```

6. **Observe Token Usage**

The moussaif-token-economy skill reports per■turn token consumption; aim to stay < 8 K tokens/turn for interactive sessions.

End of Overview

Generated on 2026■07■27 by MOUSSAIF1B (Moussaif Hermes AGI■Oriented Cognitive Kernel).