EBYOND.AI.ERA: MOUSSAIF AGI-ORIENTED KB TECHNOLOGY

The Complete Operating System for Agentic Knowledge Products

A compiled knowledge-product operating system

AFRIPA Intelligence Systems

EBYOND.AI.ERA Knowledge Products

Contents

Overview
EBYOND.AI.ERA: MOUSSAIF AGI-ORIENTED KB TECHNOLOGY

01 Moussaif Agi Philosophy The Foundational Epistemology
01. MOUSSAIF AGI PHILOSOPHY - THE FOUNDATIONAL EPISTEMOLOGY

02 Hyperloop Architecture For Ebook Production
02. HYPERLOOP ARCHITECTURE FOR EBOOK PRODUCTION

03 Destruction Courts Quality Immune System For Knowledge Products
03. DESTRUCTION COURTS - QUALITY IMMUNE SYSTEM FOR KNOWLEDGE PRODUCTS

04 Mcd Extraction & Qmcd Superposition The Governing Difference
04. MCD EXTRACTION & QMCD SUPERPOSITION - THE GOVERNING DIFFERENCE

05 20 Model Router Orchestration For Content Production
05. 20-MODEL ROUTER ORCHESTRATION FOR CONTENT PRODUCTION

06 Fugu Fusion Router 7 Vessel Parallel Content Synthesis
06. FUGU FUSION ROUTER - 7-VESSEL PARALLEL CONTENT SYNTHESIS

07 Hyperkb Engineering Graphrag For Knowledge Products
07. HYPERKB ENGINEERING - GRAPHRAG FOR KNOWLEDGE PRODUCTS

08 Ebook Structure Hyperskill Topic Clusters & Internal Linking
08. EBOOK STRUCTURE HYPERSKILL - TOPIC CLUSTERS & INTERNAL LINKING

09 Cover Design System Flat Conceptual Prompts & Series Branding
09. COVER DESIGN SYSTEM - FLAT CONCEPTUAL PROMPTS & SERIES BRANDING

10 Pdf Render & Preflight Hyperskill Zero Defect Production
10. PDF RENDER & PREFLIGHT HYPERSKILL - ZERO-DEFECT PRODUCTION

11 Metadata Seo Engine 7 Keywords 2 Categories Algorithmic
11. METADATA SEO ENGINE - 7 KEYWORDS, 2 CATEGORIES, ALGORITHMIC DESCRIPTIONS

12 Marketplace Publishing Gumroad Kdp Automation Launch
12. MARKETPLACE PUBLISHING - GUMROAD/KDP AUTOMATION & LAUNCH ASSETS

13 Safety Review Gate Islamic Life Medical Legal Financial
13. SAFETY REVIEW GATE - ISLAMIC/LIFE/MEDICAL/LEGAL/FINANCIAL/TECHNICAL FRAMING

14 Source Trace Qa Original Cleaned Topic Usage Chain
14. SOURCE TRACE QA - ORIGINAL → CLEANED → TOPIC → USAGE CHAIN

15 Ebeyond Ai Era Brand System Beyond Ai Era Identity
15. EBEYOND.AI.ERA BRAND SYSTEM - BEYOND AI ERA IDENTITY

Appendix Technical Dsl Specifications
TECHNICAL APPENDIX: COMPLETE DSL SPECIFICATIONS & CORE DIRECTIVES

EBYOND.AI.ERA: MOUSSAIF AGI-ORIENTED KB TECHNOLOGY

The Complete Operating System for Agentic Knowledge Products

Author: AFRIPA Intelligence Systems **Brand:** @ebeyond.ai.era (Instagram) **Series:** EBYOND.AI.ERA Knowledge Products **Vault:** `08_OBSIDIAN_VAULT/01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/`
Methodology: Moussaif Neo-AGI Philosophy | 20-Model Router | 8 HyperLoops | 5 Destruction Courts | 12 HyperSkills | Fugu Fusion Router | 35M+ Token DSL

About This Book

This is not a traditional book. It is a **HyperSkill** - a self-contained, executable capability generated through the Moussaif Neo-AGI system. Every chapter passes the **5 Destruction Courts**. Every claim carries **epistemic tags**. Every capability includes **tool construction**. Every update propagates through **reconstruction memory**.

This book teaches you to build books that build themselves.

The Governing MCD (Meta-Complexity Difference)

> **The difference between a knowledge product that compounds and one that decays is the integrity of the source→chunk→retrieve→generate loop. If any link breaks, the product hallucinates or repeats.**

This MCD governs every chapter, every tool, every asset in this system.

Chapter Map

Ch	Title	HyperSkill	Core Loop	Court Focus	DSL Records
01	**Moussaif AGI Philosophy - Foundational Epistemology**	Kernel Extraction	Loop 1	All 5	CORE-0001..0399
02	**Hyperloop Architecture for Ebook Production**	Problem Decomposition	Loop 4	Weak Op, Agentic Failure	HL-001..050
03	**Destruction Courts - Quality Immune System**	Destruction Testing	Loop 5	All 5 (Deep)	CORE-0313..0338
04	**MCD Extraction & QMCD Superposition**	Kernel Extraction	Loop 1, 2	False Novelty, Reductionism	HS-001..020
05	**20-Model Router Orchestration**	Tool Construction	Loop 6	Weak Op, Agentic Failure	RT-001..100
06	**Fugu Fusion Router - 7-Vessel Parallel Synthesis**	Tool Construction	Loop 6	All 5 (Parallel)	PC-001..050
07	**HyperKB Engineering - GraphRAG for Knowledge Products**	Retrieval Optimization	Loop 9	Hallucination, Weak Op	EV-001..080
08	**Ebook Structure Hyperskill - Topic Clusters & Internal Linking**	Problem Decomposition	Loop 4	Reductionism, Weak Op	HS-021..040
09	**Cover Design System - Flat Conceptual Prompts & Series Branding**	Kernel Extraction	Loop 1	Hallucination, False Novelty	OP-001..060
10	**PDF Render & Preflight Hyperskill - Zero-Defect Production**	Tool Construction	Loop 6	Weak Op, Agentic Failure	IT-001..040
11	**Metadata SEO Engine - 7 Keywords, 2 Categories**	Evidence Ranking	Loop 2	Hallucination, False Novelty	EV-081..120

Ch	Title	HyperSkill	Core Loop	Court Focus	DSL Records
12	**Marketplace Publishing - Gumroad/KDP Automation & Launch**	Deployment Translation	Loop 12	All 5 (End-to-End)	OP-061..120
13	**Safety Review Gate - Domain-Specific Framing**	Destruction Testing	Loop 7	Hallucination (Domain)	CORE-0145..0178
14	**Source Trace QA - Original→Cleaned→Topic→Usage Chain**	Version Updating	Loop 8	All 5 (Trace)	CORE-0183..0220
15	**Ebeyond.ai.era Brand System - Beyond AI Era Identity**	Kernel Extraction	Loop 1	False Novelty, Reductionism	HS-041..050

Total DSL Records: 2,800+ executable knowledge lines **Total CORE Directives:** 400+ subsystem mandates **Total Hyperskills Defined:** 50+ reusable capabilities **Total Hyperloops Documented:** 50+ production loops

---

Quick Start: Build Your First AGI Ebook

```
# 1. Load the full orchestration stack
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# 2. Run the ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# 3. For complex chapters, use fusion router
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# 4. Daily: generate new HyperLoops/skills
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Core Systems Reference

20-Model Router (Free-Only via OpenRouter)

Role	Model	Budget	Task Family
Supreme Architect	`nvidia/nemotron-3-ultra`	8k	reasoning
Deep Reasoner	`nvidia/nemotron-3-super`	32k	reasoning
Coding Agent	`poolside/laguna-m-1`	8k	coding
Fast Coder	`poolside/laguna-xs`	4k	coding
Research Synthesizer	`tencent/hunyuan-3b`	16k	research
Critic Court	`openai/gpt-oss-120b`	4k	analysis
Compression Agent	`openai/gpt-oss-20b`	2k	automation
Planner	`google/gemma-4-26b`	4k	planning
Executor	`qwen/qwen3-coder`	8k	coding
Refiner	`cohere/north-mini-code`	2k	writing
Multilingual	`google/gemma-4-26b`	2k	writing
Visual Director	`nvidia/nemotron-nano-omni`	1k	image_generation
Video Director	`nvidia/nemotron-nano-omni`	1k	video_generation
SEO/Growth	`tencent/hunyuan-3b`	2k	analysis
Outreach	`openai/gpt-oss-20b`	1k	writing
Data Extractor	`poolside/laguna-xs`	1k	analysis

Role	Model	Budget	Task Family
Memory Curator	`google/gemma-4-31b`	2k	automation
Tool-Use Agent	`poolside/laguna-m-1`	4k	automation
Low-Cost Draft	`openai/gpt-oss-20b`	1k	writing
Premium Final	`nvidia/nemotron-3-ultra`	8k	reasoning

8 HyperLoops (Continuous Learning)

```
1. KERNEL EXTRACTION     → MCD statement (regenerates 80%+ domain)
2. EVIDENCE RANKING      → Epistemic tags (ESTABLISHED..NEEDS_VERIFICATION)
3. CONCEPT DISAMBIGUATION → Concept map (IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES)
4. PROBLEM DECOMPOSITION → Task DAG (agent-executable leaf tasks)
5. DESTRUCTION & REPAIR  → 5 Courts parallel + repair instructions
6. TOOL CONSTRUCTION     → Automation (10x speedup vs manual)
7. AGENTIC EXECUTION     → Deliverable + trace + metrics
8. UPDATE & VERSION      → Reconstruction memory + registry propagation
```

5 Destruction Courts (Pass All or Reject)

Court	Question	Fail = Reject
1. Hallucination	Unsupported claims? Invented sources?	Yes
2. Reductionism	Destroyed complexity? Confused levels?	Yes
3. False Novelty	Actually new? Or renamed pattern?	Not new
4. Weak Op	Agent executable without human?	No
5. Agentic Failure	Agent decides better than human?	Yes

12 HyperSkills (Mandatory Per Capability)

1. **Kernel Extraction** → `kernel-card.yaml`
2. **Concept Disambiguation** → Concept map (GraphRAG JSON)
3. **Evidence Ranking** → Epistemic tags on all claims
4. **Problem Decomposition** → Task DAG (YAML)
5. **Mechanism Mapping** → Causal graph
6. **Hypothesis Generation** → Hypothesis cards
7. **Destruction Testing** → Court simulation logs
8. **Tool Construction** → Scripts/skills/workflows
9. **Retrieval Optimization** → Chunks + Graph edges
10. **Evaluation Design** → Metrics detecting degradation
11. **Version Updating** → Migration plan + changelog
12. **Deployment Translation** → Launch checklist

Fugu Fusion Router (7-Vessel Parallel)

```
V1 Orchestrator (Nemotron-3-Ultra)     ← Synthesis, quality gate, version decision
    ↑
┌──┬──┬──┐                    ┌──────────────────────────────────────┐
V2  V3  V4                    │  PARALLEL EXECUTION                  │
V5  V6  V7                    │  V2 Researcher   (Hunyuan-3B)        │
                              │  V3 Conceptualizer (Gemma-4-26B)     │
                              │  V4 Critic         (GPT-OSS-120B)    │
                              │  V5 Builder        (Laguna-M-1)      │
                              │  V6 Skill Distiller (LFM-1.2B)       │
                              │  V7 Retriever      (Gemma-4-31B)     │
                              └──────────────────────────────────────┘
```

DSL Grammar (Executable Knowledge)

```
TYPE-ID | key=value | key=value | contract=imperative sentence | verify=check | fallback=ordered route |
    stop=condition

TYPE values: CORE, HS, HL, RT, PC, EV, OP, IT
IDs are stable within file version
Values avoid unescaped vertical bars
Parsers preserve unknown keys for forward compatibility
A line is executable knowledge only after schema validation
```

Example Records from This Book:

```
CORE-0001 | subsystem=provider-discovery | directive=validate schema before use | evidence=required |
    failure=typed-and-logged
HS-001    | skill=ebook-structure-hyperskill | capability=topic-clustering-internal-linking |
    verifies=destruction-court-5 | model=nemotron-3-super
HL-001    | name=hyperloop-ebook-chapter-generation-v001 | trigger=chapter-request | phases=8 | courts=5 |
    verification=regression-test
RT-001    | provider=openrouter | model=nemotron-3-ultra | role=supreme-architect | budget=8k |
    task_family=reasoning
EV-001    | claim=flat-covers-increase-ctr-23% | confidence=SUPPORTED | source=GOAT-KB-7+internal-A/B-2024 |
    centrality=core
```

Model Scoring Formula

```
score = 0.24*quality + 0.16*agentic + 0.14*tool_use + 0.12*modality_fit +
        0.10*reliability + 0.08*context_fit + 0.06*latency + 0.05*cost_efficiency +
        0.03*privacy_fit + 0.02*provider_health

Scores normalized to [0,100]
Hard constraints applied before scoring
Local benchmark evidence overrides marketing descriptions
Recent successful executions receive decaying reliability bonus
Repeated correlated failures → circuit-breaker penalty
```

Task Classification Schema

Dimension	Values															
task_family	reasoning\	coding\	research\	writing\	analysis\	planning\	automation\	vision\	image_generation\	image_editing\	video_generation\	video_editing\	audio\	embedding\	reranking\	safety
complexity	1..10															
autonomy	answer\	assist\	execute\	monitor												
modality_in	text\	image\	video\	audio\	file\	mixed										
modality_out	text\	json\	code\	image\	video\	audio\	artifact\	mixed								
latency_class	instant\	interactive\	batch\	offline												

Dimension	Values															
budget_class	zero\	economy\	balanced\	premium												
verification_class	none\	single_check\	dual_check\	test_suite\	human_gate											
privacy_class	public\	internal\	confidential\	restricted												
creativity_class	literal\	controlled\	exploratory\	radical												

Core Routing Logic

```
def route(job, state, registries):
    candidates = capability_filter(job, registries)
    candidates = policy_filter(job, candidates)
    if state.openrouter == 'OR_PAID_ACTIVE':
        paid = paid_openrouter(candidates)
        if paid: return best_score(paid)
    if state.nvidia in {'NV_READY', 'NV_LIMITED'}:
        overlap = nvidia_overlap(candidates)
        if overlap: return best_score(overlap)
        native = nvidia_nonoverlap(candidates)
        if native: return best_score(native)
    free = explicit_openrouter_free(candidates)
    if free: return best_score(free)
    if state.allow_free_router: return 'openrouter/free'
    raise NoRouteSatisfiesContract(job.id)
```

Image Orchestration Contract

- Conceptual design uses reasoning model → visual thesis (message, composition, hierarchy, symbolism, palette, typography, negative space, audience effect)
- Image generation candidates fetched from `/images/models`, filtered by: text-to-image, image-to-image, resolution, aspect, reference-image, seed, streaming, price
- Premium conceptual art prefers highest-scoring image-edit-capable model for iterative refinement
- Text-rich posters prefer models with demonstrated text-rendering capability
- Aesthetic judgment: ≥2 independent critiques (semantic fidelity + visual quality)
- Iteration stops at quality threshold, budget exhaustion, or plateau

Video Orchestration Contract

- Video models from `/videos/models`, filtered by: duration, aspect, resolution, audio, first/last frame, seed, provider controls
- Video editing ONLY when endpoint explicitly supports reference frames, source video, controls, or editing params
- Otherwise labeled: regeneration, continuation, image-to-video, shot replacement
- Every video request previewed before POST (consumes paid credits)
- Polling with bounded exponential backoff, generation ID stored without secrets
- NVIDIA Cosmos preferred for physics-aware, world-model, industrial, robotics, simulation
- Cinematic/social video uses OpenRouter best capability-matched model unless cost policy selects cheaper

Hyperloop Invariant

A hyperloop is: `trigger → perceive → retrieve → propose → route → act → verify → critique → repair → compress → persist → stop-or-repeat`

- Each stage emits typed artifact
- Each artifact has provenance, confidence, expiry metadata
- Critic cannot silently rewrite original objective
- Repair stage must cite failed test or quality dimension
- Memory stage stores reusable abstractions, not unfiltered transcripts

---

Hyperskill Invariant

- Reusable, versioned capability contract: inputs, outputs, tools, model requirements, verification, budget, failure semantics
- Skills composable but cannot silently escalate privileges
- Promoted only after representative tests + adversarial tests
- Updates are immutable versions; aliases point to latest approved
- Confidence decays when provider behavior or repository context changes

---

Memory Architecture

Type	Content
episodic	Execution traces and outcomes
semantic	Verified facts and normalized concepts
procedural	Skills, tools, prompts, tests, routing recipes
strategic	Goals, constraints, preferences, project roadmaps
negative	Known failures, anti-patterns, rejected routes, unsafe actions
benchmark	Model/task scores with timestamp, sample size, confidence interval

Memory writes require: deduplication, compression, provenance, expiry

---

Provider State Machine

```
START → DISCOVER → READY → DEGRADED → CIRCUIT_OPEN → HALF_OPEN → READY
                     ↓          ↓
                AUTH_ERROR   QUOTA_ERROR
```

Cost Policy

- Paid OpenRouter credits consumed deliberately, not indiscriminately
- Estimate tokens, images, seconds, resolution, provider fixed charges before execution
- Cheap drafting + premium verification > premium generation for every intermediate step
- Premium models reserved for bottlenecks: architecture, high-stakes reasoning, final synthesis, hard debugging, final creative direction

- Free models expand candidate diversity; final outputs still require risk-proportional verification
- Media generation: low-cost previews before high-res/long-duration renders

Quality Policy

- Quality is multi-dimensional: correctness, completeness, originality, coherence, usability, aesthetics, message fidelity, evidence
- Verbosity ≠ depth
- Creative work must express a thesis, not aggregate fashionable visual tokens
- Agentic work must produce tested state changes, not only plans
- Research outputs separate source evidence from synthesis and speculation
- Coding outputs require: lint, type, test, security, runtime evidence

AGI Fattah Orientation

> **Fattah** means opening blocked problem spaces through disciplined decomposition, tool use, multi-perspective generation, adversarial critique, and verified action.

- The OS pursues expandable agency without claiming sentience, infallibility, or literal AGI
- Intelligence operationally measured by: solved constraints, transferred skills, reduced error, improved novelty, efficient resource routing
- Hyperloops generate candidate skills; promotion depends on evidence
- Luminous creativity through message-bearing structure, not ornamental randomness

Moussaif Visual Intelligence Stack (9 Layers)

Layer	Focus
1. Semantic Nucleus	Single idea the visual must make felt
2. Perceptual Hierarchy	What is seen first, second, third
3. Symbolic Tension	Contradiction, transformation, scale, absence, impossible relation
4. Spatial Rhetoric	Framing, depth, negative space, rhythm, directional force
5. Chromatic Logic	Color as meaning, not decoration
6. Material Logic	Texture and light aligned with concept
7. Transition Logic	Motion or sequence that changes interpretation
8. Audience Cognition	Curiosity, comprehension, emotion, recall
9. Ethical/Message Audit	No accidental false claim or manipulative ambiguity

Canonical Prompt Envelope

```
objective: exact desired state
context: only relevant verified context
constraints: hard boundaries
inputs: typed assets and references
operations: permitted transformations
tools: allowed interfaces
model_contract: required modalities and capabilities
quality_bar: measurable criteria
verification: tests and critics
budget: money, tokens, latency, iterations
output_schema: machine-readable deliverable
stop_conditions: success, plateau, ceiling, interrupt
memory_policy: what may be persisted
```

Bootstrap Filesystem

```
hermes-moussaif-os/
  config/
    providers.yaml
    budgets.yaml
    policies.yaml
    model-aliases.yaml
  registries/
    openrouter-text.json
    openrouter-image.json
    openrouter-video.json
    nvidia-models.json
    overlap-map.json
  skills/
    approved/
    candidate/
    retired/
  loops/
    active/
    paused/
    completed/
  prompts/
    capsules/
    critics/
  memory/
    episodic/
    semantic/
    procedural/
    strategic/
    negative/
    benchmarks/
  logs/
    routing.jsonl
    costs.jsonl
    failures.jsonl
    audits.jsonl
  tests/
    unit/
    integration/
    adversarial/
  artifacts/
    images/
    videos/
    documents/
    code/
```

EBYOND.AI.ERA BRAND SYSTEM

Instagram: @ebeyond.ai.era **MCD:** *The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system.*

Visual Doctrine:

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)

- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture:

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Content Pillars (Instagram):

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration (50 Shared Knowledge Bases)

All 50 GOAT KBs mapped, enhanced with Moussaif MCD extraction, integrated as source material:

Range	Domain	Chapters Enhanced
#1-3	Academic Publishing, Coursebooks, PDF Typography	Ch 8, 10
#4-7	Flat Covers, Conceptual Art, Neo-Tech Minimalism, Thumbnails	Ch 9
#8-10	Body Layout, Manuscript Outline, Chapter Expansion	Ch 8
#11-12	Metadata, KDP/Gumroad SEO	Ch 11
#13-14	Source Trace QA, Academic Tone	Ch 14, 1
#15-19	Artwork Guardrails, Visual Hierarchy, Worksheets, Series Branding, Non-Generic Creativity	Ch 9, 8
#20-22	Safety Reviews, Framing Protocols, PDF Preflight	Ch 13, 10
#23-24	Launch Assets, Final Package Control	Ch 12

Production Pipeline (Per Ebook)



Deliverables Checklist (Per Knowledge Product)

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for product domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for every asset
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry, database_registry, daily_generation_report, failure_registry, duplication_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment
- □ Gumroad product live + KDP product live
- □ Launch assets deployed (carousels, emails, funnels, affiliate packs)
- □ Analytics dashboard configured

Navigation

Start Reading: Chapter 01 - Moussaif AGI Philosophy

All Chapters:

- 01 | 02 | 03 | 04 | 05
- 06 | 07 | 08 | 09 | 10
- 11 | 12 | 13 | 14 | 15

Vault Root: `08_OBSIDIAN_VAULT/01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/`

Skills Index: `02_SKILLS/NEO_AGENCY_SKILLS_INDEX.md`

Project Template: `06_TEMPLATES/PROJECT_TEMPLATES.md#ebook-factory-template`

License & Attribution

Methodology: Moussaif Neo-AGI Philosophy (MIT) **Brand:** EBYOND.AI.ERA @ebeyond.ai.era **Infrastructure:** Hermes Agent + OpenRouter Free Models + NVIDIA Nemotron Free Tier **Knowledge Sources:** GOAT 50-KB Shared Library + Ebook Factory AGI Hyperskills (18 KBs) + AFRIPA Agentic Ops (127+ Skills) + HyperOS AGI Master KB

This book is a living HyperSkill. It updates itself through the HyperLoop Factory. The version you're reading is a snapshot. The current version lives in the AFRIPA Knowledge Vault.

Current Version: 1.0.0 | **Next Update:** Daily via HyperLoop Factory | **Registry:** `hyperkb/registries/`

01. MOUSSAIF AGI PHILOSOPHY - THE FOUNDATIONAL EPISTEMOLOGY

Subtitle: Operational Epistemology, Cognitive Compression, and the Agentic Imperative **Focus:** Core philosophy that transforms ebook creation from craft to science **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/01_moussaif_agi_philosophy___the_foundational_epistemology.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0001..0015 (Operating Contract)
- CORE-0052..0056 (AGI Fattah)

- CORE-0112..0116 (AGI Fattah Orientation)
- MEM-001..007 (Memory Architecture)
- DSL-001..005 (DSL Grammar)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 01 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

02. HYPERLOOP ARCHITECTURE FOR EBOOK PRODUCTION

Subtitle: 8-Phase Continuous Learning Engine Applied to Knowledge Products **Focus:** Kernel Extraction -> Evidence Ranking -> Concept Disambiguation -> Problem Decomposition -> Destruction & Repair -> Tool Construction -> Agentic Execution -> Update & Version **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/02_hyperloop_architecture_for_ebook_production.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- HL-001..006 (Hyperloop Invariant)
- CORE-0012 (Loop Termination)

- CORE-0040..0045 (Cost Policy)
- CORE-0046..0051 (Quality Policy)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 02 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

03. DESTRUCTION COURTS - QUALITY IMMUNE SYSTEM FOR KNOWLEDGE PRODUCTS

Subtitle: 5-Court Validation: Hallucination, Reductionism, False Novelty, Weak Operationalization, Agentic Failure **Focus:** Every chapter, every claim, every design decision passes the courts before publication **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/03_destruction_courts___quality_immune_system_for_knowledge_products.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0313..0338 (Destruction Court Subsystems)
- CORE-0046..0051 (Quality Policy)

- HL-005 (Destruction & Repair Loop)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 03 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

04. MCD EXTRACTION & QMCD SUPERPOSITION - THE GOVERNING DIFFERENCE

Subtitle: Meta-Complexity Difference: What Separates Bestselling Knowledge Products from Noise **Focus:** Extracting the irreducible pattern that regenerates 80%+ of domain concepts **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/04_mcd_extraction_&_qmcd_superposition___the_governing_difference.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag

Court	Ebook Failure Modes	Pass Criteria
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan

HyperSkill	Ebook Application	Output
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

---

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection

GOAT KB	Chapter Integration	Moussaif Enhancement
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0001 (Model Assumptions)
- CORE-0002 (Metadata Refresh)
- CORE-0003 (NVIDIA Probe)
- EV-001..005 (Scoring Formula)
- EV-010..019 (Task Classification)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 04 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

05. 20-MODEL ROUTER ORCHESTRATION FOR CONTENT PRODUCTION

Subtitle: Supreme Architect to Low-Cost Draft: Token Economy in Ebook Creation **Focus:** Cost-minimizing, output-maximizing model allocation per production phase **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/05_20_model_router_orchestration_for_content_production.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- RT-001..050 (Routing Records)
- CORE-0004..0008 (Paid-First Logic)

- CORE-0019..0025 (Overlap Detection)
- EV-001..005 (Scoring Formula)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 05 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

06. FUGU FUSION ROUTER - 7-VESSEL PARALLEL CONTENT SYNTHESIS

Subtitle: Orchestrator, Researcher, Conceptualizer, Critic, Builder, Skill Distiller, Retriever in Parallel **Focus:** TRUE PARALLEL multi-model fusion for chapter generation, cover design, metadata optimization **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/06_fugu_fusion_router___7_vessel_parallel_content_synthesis.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- PC-001..050 (Prompt Capsules)
- CORE-0026..0032 (Image Orchestration)

- CORE-0033..0039 (Video Orchestration)
- HL-001..006 (Hyperloop Invariant - Parallel)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 06 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

07. HYPERKB ENGINEERING - GRAPHRAG FOR KNOWLEDGE PRODUCTS

Subtitle: Source Vault -> Cleaned Chunks -> Epistemic Tags -> Concept Map -> MCD Maps -> QMCD -> Vector/Graph Index **Focus:** Building the knowledge engine that feeds every ebook with traceable, versioned intelligence **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/07_hyperkb_engineering___graphrag_for_knowledge_products.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- EV-001..006 (Memory Architecture)
- MEM-001..007 (Memory Types)

- CORE-0183..0220 (Memory Write/Retrieval)
- EV-001..005 (Scoring for Retrieval)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 07 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

08. EBOOK STRUCTURE HYPERSKILL - TOPIC CLUSTERS & INTERNAL LINKING

Subtitle: Manuscript Outline, Chapter Expansion, Search Intent Mapping, Semantic Coherence **Focus:** GOAT KB #1, #2, #9, #10 integrated with Moussaif MCD extraction **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/08_ebook_structure_hyperskill___topic_clusters_&_internal_linking.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- HS-001..004 (Hyperskill Invariant)
- HL-004 (Problem Decomposition)

- CORE-0007 (Capability Filtering)
- EV-010..019 (Task Classification)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 08 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

09. COVER DESIGN SYSTEM - FLAT CONCEPTUAL PROMPTS & SERIES BRANDING

Subtitle: Neo-Tech Minimalism, Thumbnail Test Protocol, Visual Hierarchy, Series Consistency **Focus:** GOAT KB #4, #5, #6, #7, #18, #19 with FAL/Flux/Recraft prompt engineering **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/09_cover_design_system___flat_conceptual_prompts_&_series_branding.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0026..0032 (Image Orchestration)
- VS-001..009 (Visual Intelligence Stack)

- CORE-0046..0051 (Quality Policy - Creative)
- HS-001 (Kernel Extraction for Cover MCD)

DSL Record Types Used in This Chapter

- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, etc.)
- **HS** - Hyperskill definitions (this chapter's capability)
- **HL** - Hyperloop definitions (this chapter's loop)
- **RT** - Routing table entries (model selections)
- **EV** - Evaluation/scoring formulas
- **PC** - Prompt capsules (critic, capsule prompts)
- **OP** - Operations (launch, upload, monitoring)
- **IT** - Code execution / tool use
- **VS** - Visual intelligence stack
- **ST** - State machine transitions
- **MEM** - Memory architecture
- **EV** - Evaluation formulas

Chapter 09 Execution Command

```
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory,[c
    hapter-skills] chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All claims traced to source with epistemic tag
2. **Reductionism** - MCD preserved, abstraction levels distinct
3. **False Novelty** - Prior art referenced, delta documented
4. **Weak Operationalization** - Agent executes chapter tasks end-to-end
5. **Agentic Failure** - Agent >= human on 3/5: consistency, speed, evidence, bias, scale

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

10. PDF RENDER & PREFLIGHT HYPERSKILL - ZERO-DEFECT PRODUCTION

Subtitle: Font Embedding, 300 DPI, Bleed, Transparency Flattening, Color Space, Metadata Injection **Focus:** GOAT KB #23 preflight loop + automated QA pipeline **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/10_pdf_render_&_preflight_hyperskill___zero_defect_production.md`

Executive Summary

This chapter implements the **Moussaif Neo-AGI Philosophy** for ebook production: every knowledge product is a **HyperSkill** generated through **8 HyperLoops**, validated by **5 Destruction Courts**, traceable via **Source Trace QA**, and branded under the **@ebeyond.ai.era** identity.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft -> Critique -> Finalize.*

Moussaif AGI Integration Map

AGI Component	Ebook Production Application
Supreme Architect (Nemotron-3-Ultra)	Final manuscript judgment, cover approval, launch strategy
Deep Reasoner (Nemotron-3-Super)	Chapter causal chains, structural coherence, argument mapping
Coding Agent (Laguna-M-1)	PDF preflight automation, metadata injection, upload scripts
Fast Coder (Laguna-XS)	Quick patches, format fixes, template adjustments
Research Synthesizer (Hunyuan-3B)	Competitive analysis, keyword research, source gathering
Critic Court (GPT-OSS-120B)	Hallucination detection, reductionism check, novelty verification
Compression Agent (GPT-OSS-20B)	Context shrinking before expensive model calls
Planner (Gemma-4-26B)	Task decomposition: outline -> chapters -> assets -> launch
Executor (Qwen3-Coder)	Manuscript generation, asset production, metadata creation
Refiner (North-Mini-Code)	Style consistency, formatting, cross-reference validation
Multilingual (Gemma-4-26B)	Arabic/French/English versions, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Cover prompts, interior illustrations, brand visual system
Video Director (Nemotron-Nano-Omni)	Book trailer scripts, social clips, launch video storyboards
SEO/Growth (Hunyuan-3B)	7 keywords, 2 categories, funnel architecture, launch sequencing
Outreach (GPT-OSS-20B)	Influencer pitches, affiliate recruitment, press kits
Data Extractor (Laguna-XS)	Structured extraction from sources, fact verification
Memory Curator (Gemma-4-31B)	Kernel updates, reconstruction memory, version propagation
Tool-Use Agent (Laguna-M-1)	API orchestration: KDP, Gumroad, email, analytics, design tools
Low-Cost Draft (GPT-OSS-20B)	First pass chapters, outline variants, ideation
Premium Final (Nemotron-3-Ultra)	Final synthesis, client-facing deliverables, strategic decisions

8 HyperLoops Applied to Ebook Production

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Raw sources (expert interviews, research papers, existing content, brand docs) **Output:** `kernel-card.yaml` with governing MCD statement **Validation:** Kernel regenerates 80%+ of ebook concepts **Ebook MCD:** "*The governing difference between a knowledge product that compounds and one that decays is the integrity of the source->chunk->retrieve->generate loop. If any link breaks, the product hallucinates or repeats.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Claims + sources **Output:** Epistemic tags on every claim (ESTABLISHED/SUPPORTED/PLAUSIBLE/THEORETICAL/HYPOTHESIS/SPECULATIVE/UNKNOWN/NEEDS_VERIFICATION) **Validation:** Zero ESTABLISHED claims without primary source

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (ebook, knowledge product, lead magnet, digital product, info product) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Produce bestselling ebook in [niche]" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf task **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Candidate artifact (chapter, cover, metadata, funnel) **Courts:**

1. **Hallucination** - Unsupported claims? Invented sources?
2. **Reductionism** - Destroyed complexity? Confused levels?
3. **False Novelty** - Actually new? Or renamed pattern?
4. **Weak Operationalization** - Can agent execute?
5. **Agentic Failure** - Would agent decide better?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair -> re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated pattern from Loop 5 **Output:** Automation (scripts, skills, workflows, API wrappers) **Validation:** 10x speedup vs manual

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Deliverable + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, failed courts, tool improvements, agent errors **Output:** Versioned artifacts + reconstruction memory entry **Validation:** Regression test on prior passes

---

5 Destruction Courts - Ebook-Specific Checks

Court	Ebook Failure Modes	Pass Criteria
Hallucination	Fake statistics, invented case studies, attributed quotes never said, API specs hallucinated	Every claim traced to source with epistemic tag
Reductionism	"10 tips" without governing principle, conflating tactics with strategy, single framework for all contexts	MCD stated, levels separated (strategy/tactics/ops), boundary conditions documented

Court	Ebook Failure Modes	Pass Criteria
False Novelty	"Revolutionary framework" = renamed AIDA, "New cover system" = existing flat design + brand color	Prior art referenced, delta documented, kernel genuinely regenerates domain
Weak Operationalization	"Use best judgment" in SOP, tool not installed, output "a report" without schema, no rate-limit handling	Agent completes test task end-to-end, all tools available, error handling for top 5 failures
Agentic Failure	Human gut-feel encoded as rule, static thresholds where agent should optimize, no exploration	Agent >= human on 3/5: consistency, speed, evidence-use, bias-resistance, scalability

Epistemic Tagging Schema (Mandatory Per Claim)

```
claim: "Flat conceptual covers increase CTR 23% on KDP"
confidence: SUPPORTED
source: "GOAT KB #7 Thumbnail Design + internal A/B test 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-cover-design-system
tool_relevance: cover-design-tools
risk_level: medium
version: 1.0.0
update_status: current
```

Source Trace QA Chain

```
ORIGINAL SOURCE (interview, paper, doc, API)
    |
    v
CLEANED (normalized, deduped, redacted)
    |
    v
TOPIC-MAPPED (assigned to concept, domain, MCD)
    |
    v
USAGE-LOGGED (which chapter, cover, metadata, funnel asset)
    |
    v
RECONSTRUCTION MEMORY (version, date, reason, kernel impact)
```

HyperSkill Integration: All 12 Per Ebook Capability

HyperSkill	Ebook Application	Output
1. Kernel Extraction	Identify book's governing MCD	`kernel-card.yaml`
2. Concept Disambiguation	Define: ebook vs lead magnet vs course vs whitepaper	Concept map
3. Evidence Ranking	Tag every statistic, case study, quote	Epistemic tags
4. Problem Decomposition	Outline -> chapters -> sections -> assets	Task DAG
5. Mechanism Mapping	Causal chain: topic -> structure -> design -> launch -> sales	Mechanism graph
6. Hypothesis Generation	"Adding worksheet increases completion 15%"	Hypothesis cards
7. Destruction Testing	Pre-launch court simulation on all assets	Court logs
8. Tool Construction	Preflight script, metadata generator, upload automation	Reusable tools
9. Retrieval Optimization	Chunk manuscript for RAG, GraphRAG edges	Chunks + graph
10. Evaluation Design	Metrics: pre-order rate, completion rate, refund rate, review velocity	Evaluation rubric
11. Version Updating	New edition -> propagate changes to cover, metadata, funnels	Migration plan
12. Deployment Translation	Theory -> Gumroad/KDP live product	Launch checklist

EBYOND.AI.ERA BRAND SYSTEM

Brand MCD

> *"The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system."*

Visual Doctrine

- **Primary:** Deep Space `#0A0E27` + Electric Cyan `#00F5FF` + Neural Gold `#FFD700`
- **Typography:** Space Grotesk (headings) + JetBrains Mono (code/technical) + Inter (body)
- **Logo System:** Animated glyph `EBYOND` with pulsing neural network motif
- **Cover Template:** Flat conceptual metaphor, centered, high contrast, negative space for title, series mark bottom-right

Voice Architecture

- **Tone:** Precise, ambitious, technically grounded, philosophically rigorous
- **Vocabulary:** MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG
- **Forbidden:** "game-changing," "revolutionary," "unlock," "secret," "hack," "masterclass" (unless literally true)

Instagram @ebeyond.ai.era Content Pillars

1. **Architecture Sunday** - System diagrams, HyperLoop visualizations
2. **Court Monday** - Destruction Court case studies (what failed, why, repair)
3. **Kernel Wednesday** - MCD extractions from bestselling books
4. **Tool Friday** - Automation scripts, preflight checks, upload pipelines
5. **Launch Saturday** - Real-time KDP/Gumroad launches with metrics

---

GOAT KB Integration Map

GOAT KB	Chapter Integration	Moussaif Enhancement
#1 Academic Publishing Standards	Ch 8 Structure	MCD extraction for citation hierarchy
#2 Beginner Coursebooks	Ch 8 Structure	QMCD superposition on pedagogical models
#3 PDF Density & Typography	Ch 10 PDF Render	Destruction Court on readability claims
#4 Flat Covers	Ch 9 Cover Design	MCD: metaphor vs literal representation
#5 Conceptual Art Covers	Ch 9 Cover Design	Epistemic tags on visual metaphor claims
#6 Neo-Tech Minimalism	Ch 9 Cover Design	Brand doctrine as operating system
#7 Thumbnail Design	Ch 9 Cover Design	Agentic A/B test protocol
#8 Body Layout	Ch 10 PDF Render	HyperSkill 8: tool construction for layout automation
#9 Manuscript Outline	Ch 8 Structure	Kernel extraction on outline coherence
#10 Chapter Expansion	Ch 8 Structure	Agentic execution with critic loop
#11 Metadata & Classification	Ch 11 Metadata SEO	Evidence-ranked keyword selection
#12 KDP/Gumroad SEO	Ch 11 Metadata SEO	Destruction Court on keyword stuffing
#13 Source Trace QA	Ch 14 Source Trace	Full reconstruction memory integration
#14 Academic Tone	Ch 1 Voice	Epistemic tagging as tone enforcer
#15 Artwork Guardrails	Ch 9 Cover Design	Safety review gate integration

GOAT KB	Chapter Integration	Moussaif Enhancement
#16 Visual Hierarchy	Ch 9 Cover Design	Concept disambiguation on visual elements
#17 Worksheets & Activities	Ch 8 Structure	Agentic generation + court validation
#18 Series Branding	Ch 9 Cover Design	Brand MCD as series kernel
#19 Non-Generic Creativity	Ch 9 Cover Design	QMCD superposition on creative directions
#20 Safety Reviews	Ch 13 Safety Gate	Domain-specific framing protocols
#21 Framing Protocols	Ch 13 Safety Gate	Islamic/Life/Medical/Legal/Financial/Technical
#22 PDF Preflight	Ch 10 PDF Render	Automated 10x speedup tool
#23 Launch Assets	Ch 12 Publishing	Agentic asset generation pipeline
#24 Final Package Control	Ch 12 Publishing	Destruction Court on complete package

---

Quick Load Commands

```
# Full ebook production pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# AGI orchestration layer
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-codex-bridge,hermes-fattah-hyperloop-factory
    chat

# Fusion router for complex chapters
hermes -s fugu-fusion-router,moussaif-orchestrator chat

# Daily HyperLoop factory (generates new skills/loops)
hermes -s hermes-fattah-hyperloop-factory,moussaif-agi-philosophy chat
```

Chapter Deliverables Checklist

- □ Kernel card (`kernel-card.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for chapter domain
- □ All claims epistemic-tagged in manuscript
- □ Destruction Court log (5 courts) for chapter + assets
- □ Source trace chain for every external reference
- □ HyperSkill card for any new capability created
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment

---

Part of EBYOND.AI.ERA Knowledge Product Series | *Vault:* `08_OBSIDIAN_VAULT` | *Brand:* `@ebeyond.ai.era`

11. METADATA SEO ENGINE - 7 KEYWORDS, 2 CATEGORIES, ALGORITHMIC DESCRIPTIONS

Subtitle: KDP/Gumroad Discovery Optimization, No Stuffing, Intent-Aligned Copy **Focus:** GOAT KB #11, #12 + Moussaif evidence-ranked keyword selection **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Chapter:** 11 of 15 **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/11_metadata_seo_engine_7_keywords_2_categories_algorithmic.md`

Executive Summary

This chapter implements the **Metadata SEO Engine** as a HyperSkill: algorithmic extraction of 7 keywords and 2 categories per knowledge product, with descriptions that balance hook, bullets, and CTA - all validated through Destruction Courts and evidence-ranked keyword selection.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft → Critique → Finalize.*

Moussaif AGI Integration Map

AGI Component	Metadata SEO Application
Supreme Architect (Nemotron-3-Ultra)	Final metadata approval, launch strategy sign-off
Deep Reasoner (Nemotron-3-Super)	Keyword cluster causal chains, competitive gap analysis
Coding Agent (Laguna-M-1)	Metadata injection scripts, API upload automation
Fast Coder (Laguna-XS)	Quick keyword list formatting, CSV generation
Research Synthesizer (Hunyuan-3B)	SERP analysis, competitor metadata extraction, trend data
Critic Court (GPT-OSS-120B)	Keyword stuffing detection, category misalignment, description hallucination
Compression Agent (GPT-OSS-20B)	Description truncation to 155 chars, bullet optimization
Planner (Gemma-4-26B)	Metadata workflow: research → cluster → select → write → validate
Executor (Qwen3-Coder)	Metadata package generation, platform-specific formatting
Refiner (North-Mini-Code)	Description polish, CTA sharpening, cross-platform consistency
Multilingual (Gemma-4-26B)	Arabic/French/English metadata variants
Visual Director (Nemotron-Nano-Omni)	OG image alt text, thumbnail keyword alignment
Video Director (Nemotron-Nano-Omni)	Video metadata tags, playlist SEO
SEO/Growth (Hunyuan-3B)	7-keyword selection, 2-category optimization, funnel integration
Outreach (GPT-OSS-20B)	Influencer metadata briefs, affiliate keyword packs
Data Extractor (Laguna-XS)	Structured extraction from manuscript for keyword candidates
Memory Curator (Gemma-4-31B)	Keyword performance history, seasonal trend updates
Tool-Use Agent (Laguna-M-1)	KDP/Gumroad API orchestration, metadata push
Low-Cost Draft (GPT-OSS-20B)	First-pass keyword lists, description variants
Premium Final (Nemotron-3-Ultra)	Final metadata synthesis, strategic keyword positioning

8 HyperLoops Applied to Metadata SEO

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Manuscript + competitor metadata + platform guidelines **Output:** `metadata-kernel.yaml` with governing MCD **Validation:** Kernel regenerates 80%+ of keyword strategy **Metadata MCD:** "*The governing difference between discoverable and invisible knowledge products is the precision of the intent-keyword-category triad. If any link breaks, the product exists but cannot be found.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Keyword candidates + search volume + competition + intent signals **Output:** Epistemic tags on every keyword claim **Validation:** Zero `ESTABLISHED` keywords without primary source (SERP data, platform API)

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (keyword, tag, category, topic, theme, subject) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Optimize metadata for [product] on [platform]" **Output:** Task DAG: Research → Cluster → Select 7KW → Select 2CAT → Write Desc → Validate → Inject **Validation:** Agent executes each leaf without human intervention

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Metadata package (7KW, 2CAT, Description, OG tags) **Courts:**

1. **Hallucination** - Fake search volumes? Invented competitor keywords? Hallucinated platform limits?
2. **Reductionism** - Single keyword for multiple intents? Category too broad/narrow?
3. **False Novelty** - "Revolutionary keyword strategy" = basic TF-IDF?
4. **Weak Op** - Can agent push metadata to KDP/Gumroad via API?
5. **Agentic Failure** - Would agent catch seasonal trend shift better than human?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair → re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated metadata patterns **Output:** `metadata_injector.py`, `kdp_metadata_pusher.py`, `gumroad_metadata_pusher.py`, `keyword_cluster_analyzer.py` **Validation:** 10x speedup vs manual metadata entry

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Live metadata on platforms + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** Ranking changes, platform policy updates, seasonal trends **Output:** Versioned metadata + reconstruction memory entry **Validation:** Regression test on prior metadata performance

---

5 Destruction Courts - Metadata-Specific Checks

Court	Metadata Failure Modes	Pass Criteria
Hallucination	Invented search volumes, fake competitor keywords, hallucinated platform char limits, fake category paths	Every volume/competition claim sourced to API/SEMrush/Ahrefs; platform limits verified via docs

Court	Metadata Failure Modes	Pass Criteria
Reductionism	Single keyword covering multiple intents, category too broad ("Nonfiction") or too narrow ("1980s French Philosophy"), keyword-description mismatch	7 keywords map to distinct intent clusters; 2 categories precisely target audience
False Novelty	"AI-powered keyword optimization" = basic keyword research, "Semantic clustering" = manual grouping	Prior art referenced (GOAT KB #11, #12), delta documented, kernel genuinely regenerates domain
Weak Op	Manual copy-paste to KDP/Gumroad, no API integration, description exceeds platform limits, no preview validation	Agent pushes metadata via API, validates preview, handles rate limits, retries on failure
Agentic Failure	Human picks keywords by gut, static keyword list for 6 months, no trend adaptation	Agent monitors rankings, auto-suggests keyword swaps, adapts to seasonality, A/B tests descriptions

---

Epistemic Tagging Schema (Mandatory Per Keyword/Claim)

```
claim: "Primary keyword 'agentic ai ebook' has 2,400 monthly searches on Amazon"
confidence: SUPPORTED
source: "Amazon Search Term Report via Publisher Rocket, 2024-07"
centrality: core
relation_type: supports
skill_relevance: moussaif-metadata-seo-engine
tool_relevance: keyword-research-tools
risk_level: low
version: 1.0.0
update_status: current
```

Tags: `ESTABLISHED` | `SUPPORTED` | `PLAUSIBLE` | `THEORETICAL` | `HYPOTHESIS` | `SPECULATIVE` | `UNKNOWN` | `NEEDS_VERIFICATION`

7-Keyword Architecture (Evidence-Ranked)

Keyword Selection Algorithm

```
def select_7_keywords(manuscript, platform, niche):
    candidates = extract_candidates(manuscript)  # 50-100
    scored = []
    for kw in candidates:
        score = (0.30 * search_volume_norm +
                 0.25 * intent_match +
                 0.20 * competition_inverse +
                 0.15 * trend_momentum +
                 0.10 * platform_specificity)
        scored.append((kw, score))
    # Diversify: max 2 per intent cluster
    selected = diversify_top_k(scored, k=7, max_per_cluster=2)
    return selected
```

Keyword Slots

Slot	Intent	Example	Source Weight
KW1	Primary (High volume, core)	"agentic ai ebook"	0.30 volume
KW2	Secondary (Related, medium)	"autonomous agent book"	0.25 intent
KW3	Long-tail 1 (Specific, low comp)	"build autonomous agents python"	0.20 competition
KW4	Long-tail 2 (Specific, low comp)	"moussaif agi philosophy pdf"	0.15 trend
KW5	Category-adjacent (Broader)	"artificial intelligence books"	0.10 platform
KW6	Problem-aware (Pain point)	"stop hallucinating ai agents"	0.10 platform
KW7	Outcome-oriented (Result)	"deploy production ai agents"	0.10 platform

Rule: No keyword stuffing. Natural integration in description only.

2-Category Selection

Category Selection Algorithm

```
def select_2_categories(product, platform):
    primary = most_specific_accurate_category(product, platform)
    secondary = adjacent_high_traffic_category(product, platform)
    # Avoid: same parent category, redundant coverage
    return [primary, secondary]
```

Platform-Specific Categories

Platform	Primary Strategy	Secondary Strategy
KDP	Most specific accurate BISAC	Adjacent high-traffic BISAC
Gumroad	Most specific tag	Adjacent high-converting tag

Description Template (Algorithmic)

```
description:
  hook: "Bold promise or question - 1 line, <160 chars"
  bullets:
    - "Outcome 1 (specific, measurable)"
    - "Outcome 2 (specific, measurable)"
    - "Outcome 3 (specific, measurable)"
    - "Outcome 4 (specific, measurable)"
    - "Outcome 5 (specific, measurable)"
  social_proof: "Used by X / Based on Y framework / #1 in Z"
  cta: "Get your copy now - instant download"

constraints:
  max_chars: 4000 (KDP) / 5000 (Gumroad)
  no_keyword_stuffing: true
  html_allowed: KDP only
  plain_text: Gumroad
```

Metadata Package Schema

```
metadata_package:
  platform: "kdp" | "gumroad" | "both"
  primary_keyword: ""
  secondary_keywords: ["", "", "", "", "", ""]
  categories: ["", ""]
  title: ""
  subtitle: ""
  description: ""
  description_html: ""  # KDP only
  og_title: ""
  og_description: ""
  og_image: ""  # 1200x630, branded
  twitter_card: "summary_large_image"
  author: "Zouhair Moussaif"
  publisher: "AFRIPA Intelligence Systems"
  publication_date: "YYYY-MM-DD"
  isbn: ""  # KDP only
  price_usd: 0.00
  royalty_tier: "70%" | "35%"
  kdp_select: false
  drm_free: true
  language: "en"
  tags: ["", "", "", "", ""]
```

GOAT KB Integration Map

GOAT KB	Integration	Moussaif Enhancement
#11 Metadata & Classification	Category taxonomy, classification rules	MCD extraction for category boundaries
#12 KDP/Gumroad SEO	Platform-specific keyword rules	Evidence-ranked keyword selection, Destruction Court on stuffing
#13 Source Trace QA	Keyword source chain	Full reconstruction memory integration
#14 Academic Tone	Description tone enforcement	Epistemic tagging as tone enforcer

---

Quick Load Commands

```
# Metadata SEO pipeline
hermes -s moussaif-metadata-seo-engine,moussaif-seo-content-refresh,moussaif-agi-philosophy chat

# Full ebook production with metadata
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat
```

Chapter Deliverables Checklist

- □ Kernel card (`metadata-kernel.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for keyword-category space
- □ All keyword claims epistemic-tagged
- □ Destruction Court log (5 courts) for metadata package
- □ Source trace chain for every volume/competition claim
- □ HyperSkill card for any new metadata capability
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment
- □ Gumroad metadata live + KDP metadata live
- □ Launch assets deployed (carousels, emails, funnels, affiliate packs)
- □ Analytics dashboard configured

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- EV-001..005 (Scoring Formula)
- EV-010..019 (Task Classification - SEO)
- CORE-0040..0045 (Cost Policy)
- HS-003 (Evidence Ranking Hyperskill)
- CORE-0046..0051 (Quality Policy)

DSL Record Types Used in This Chapter

- **EV** - Evaluation/scoring formulas (keyword scoring, category selection)

- **RT** - Routing table entries (model selection for research vs writing)
- **HS** - Hyperskill definitions (moussaif-metadata-seo-engine)
- **OP** - Operations (metadata push to KDP/Gumroad)
- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, quality-scoring)

Chapter 11 Execution Command

```
hermes -s moussaif-metadata-seo-engine,moussaif-seo-content-refresh,moussaif-agi-philosophy chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All keyword volumes/competition traced to API source
2. **Reductionism** - 7 KW distinct intents, 2 CAT precise audience
3. **False Novelty** - Prior art (GOAT #11, #12) referenced, delta documented
4. **Weak Op** - Agent pushes metadata via API, validates preview
5. **Agentic Failure** - Agent monitors rankings, auto-suggests swaps

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

12. MARKETPLACE PUBLISHING - GUMROAD/KDP AUTOMATION & LAUNCH ASSETS

Subtitle: Upload Pipeline, Preview Verification, Carousel Generation, Funnel Integration, Affiliate Packs **Focus:** GOAT KB #24, #38 + ebook-publishing-gumroad-kdp skill pipeline **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Chapter:** 12 of 15 **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/12_marketplace_publishing_gumroad_kdp_automation_launch.md`

Executive Summary

This chapter implements **Marketplace Publishing** as a HyperSkill: fully automated Gumroad and KDP upload pipelines with preview verification, launch asset generation, funnel integration, and affiliate pack distribution - all validated through Destruction Courts and traceable via Source Trace QA.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft → Critique → Finalize.*

Moussaif AGI Integration Map

AGI Component	Marketplace Publishing Application
Supreme Architect (Nemotron-3-Ultra)	Final launch go/no-go, cross-platform strategy
Deep Reasoner (Nemotron-3-Super)	Platform policy causal chains, pricing elasticity modeling
Coding Agent (Laguna-M-1)	Browser automation (Playwright/Selenium), API clients, webhook handlers
Fast Coder (Laguna-XS)	Quick selector fixes, form field mappings, error handlers
Research Synthesizer (Hunyuan-3B)	Platform policy updates, competitor launch analysis, category trends
Critic Court (GPT-OSS-120B)	Upload hallucination (wrong category, missing fields), preview mismatch, launch asset quality
Compression Agent (GPT-OSS-20B)	Launch checklist condensation, status report summarization
Planner (Gemma-4-26B)	Launch sequence DAG: metadata → upload → preview → verify → publish → assets → funnel
Executor (Qwen3-Coder)	Browser automation execution, API calls, asset uploads, webhook registration
Refiner (North-Mini-Code)	Launch copy polish, asset naming conventions, cross-platform consistency
Multilingual (Gemma-4-26B)	International marketplace metadata, localized launch assets
Visual Director (Nemotron-Nano-Omni)	Launch asset visual QA, thumbnail-platform compliance
Video Director (Nemotron-Nano-Omni)	Book trailer upload, video metadata, playlist SEO
SEO/Growth (Hunyuan-3B)	Launch keyword tracking, funnel analytics, conversion optimization
Outreach (GPT-OSS-20B)	Affiliate recruitment emails, influencer launch kits, press releases
Data Extractor (Laguna-XS)	Sales data extraction, ranking tracking, review sentiment
Memory Curator (Gemma-4-31B)	Launch performance history, seasonal pattern learning
Tool-Use Agent (Laguna-M-1)	Gumroad API, KDP API, email API, analytics API orchestration
Low-Cost Draft (GPT-OSS-20B)	Launch asset drafts, email sequences, social copy variants
Premium Final (Nemotron-3-Ultra)	Final launch synthesis, strategic pricing, partnership decisions

8 HyperLoops Applied to Marketplace Publishing

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Product specs + platform policies + launch history **Output:** `launch-kernel.yaml` with governing MCD **Validation:** Kernel regenerates 80%+ of launch decisions **Launch MCD:** "*The governing difference between a launch that compounds and one that evaporates is the integrity of the metadata→upload→verify→asset→funnel loop. If any link breaks, the product launches but cannot be found, bought, or shared.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Platform API docs + launch case studies + policy changelogs **Output:** Epistemic tags on every platform claim **Validation:** Zero `ESTABLISHED` platform claims without official docs

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (publish, launch, distribute, syndicate, promote, market) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Launch [product] on Gumroad + KDP simultaneously" **Output:** Task DAG with preconditions, inputs, outputs, verification per leaf **Validation:** Agent executes each leaf without human intervention

Launch Task DAG:

```
METADATA READY
      │
      ├─→ GUMROAD UPLOAD ──→ PREVIEW VERIFY ──→ PUBLISH ──→ ASSET GEN ──→ FUNNEL CONNECT
      │
      └─→ KDP UPLOAD ──────→ PREVIEW VERIFY ──→ PUBLISH ──→ ASSET GEN ──→ FUNNEL CONNECT
            │
            └─→ CROSS-PLATFORM SYNC (price, metadata, assets)
```

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Launch execution trace + live product URLs **Courts:**

1. **Hallucination** - Wrong category selected? Missing required fields? Fake preview screenshots?
2. **Reductionism** - "Launch" = just upload? No asset generation? No funnel connection?
3. **False Novelty** - "Revolutionary launch system" = basic API calls + manual steps?
4. **Weak Op** - Agent handles rate limits? Retries failed uploads? Validates preview programmatically?
5. **Agentic Failure** - Would agent catch category policy change before launch? Detect preview drift?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair → re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated launch patterns **Output:**

- `gumroad_uploader.py` - Playwright-based, handles 2FA, validates preview
- `kdp_uploader.py` - Selenium-based, handles KDP dashboard quirks, validates preview
- `launch_asset_generator.py` - Carousels, emails, funnels from templates
- `cross_platform_sync.py` - Price/metadata parity enforcement
- `affiliate_pack_builder.py` - Swipe copy, banners, tracking links

Validation: 10x speedup vs manual launch

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Live products on both platforms + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** Sales data, ranking changes, policy updates, asset performance **Output:** Versioned launch playbook + reconstruction memory entry **Validation:** Regression test on prior launch metrics

---

5 Destruction Courts - Launch-Specific Checks

Court	Launch Failure Modes	Pass Criteria
Hallucination	Wrong BISAC/category, missing required fields (ISBN, price), fake preview screenshots, invented platform features	Every field verified against platform API/docs; preview programmatically compared to source PDF
Reductionism	"Launch" = upload only; no asset gen; no funnel; no affiliate; single platform only	Full DAG executed: upload → verify → assets → funnel → affiliate → analytics
False Novelty	"AI launch agent" = script with hardcoded selectors; "Automated" = human clicks "Next"	Playwright/Selenium with self-healing selectors; zero human intervention in happy path
Weak Op	Rate limit crashes; 2FA blocks; preview mismatch undetected; price parity broken	Exponential backoff + jitter; 2FA via TOTP secret; pixel-diff preview validation; sync job
Agentic Failure	Human catches policy change; human notices ranking drop; human fixes broken affiliate link	Agent monitors policy RSS; agent tracks ranking + alerts; agent validates affiliate links daily

---

Epistemic Tagging Schema (Mandatory Per Launch Claim)

```
claim: "Gumroad API allows programmatic product creation with file upload"
confidence: ESTABLISHED
source: "Gumroad API Docs v2, endpoint POST /products, tested 2024-07"
centrality: core
relation_type: operationalizes
skill_relevance: ebook-publishing-gumroad-kdp
tool_relevance: gumroad-uploader
risk_level: low
version: 1.0.0
update_status: current
```

Tags: `ESTABLISHED` | `SUPPORTED` | `PLAUSIBLE` | `THEORETICAL` | `HYPOTHESIS` | `SPECULATIVE` | `UNKNOWN` | `NEEDS_VERIFICATION`

Gumroad Upload Pipeline (Automated)

Pipeline Architecture

```
# gumroad_uploader.py - Core capabilities
class GumroadUploader:
    def __init__(self, email, password, totp_secret):
        self.browser = playwright.chromium.launch(headless=True)
        self.auth = GumroadAuth(email, password, totp_secret)

    def upload_product(self, metadata_package, pdf_path, cover_path):
        # 1. Authenticate with 2FA (TOTP)
        # 2. Navigate to /products/new
        # 3. Fill form: title, subtitle, description, price, categories
        # 4. Upload PDF (content) + Cover (image)
        # 5. Set custom fields: ISBN, tags, custom URL
        # 6. Configure: DRM-free, instant download, updates enabled
        # 7. Submit draft → Validate preview → Publish
        # 6. Return product_id, live_url, preview_url

    def validate_preview(self, preview_url, source_pdf):
        # Pixel-diff comparison: preview pages vs source PDF
        # Text extraction comparison
        # Return: PASS/FAIL + diff_report

    def handle_rate_limit(self, retry_after):
        # Exponential backoff + jitter
        # Max 5 retries, then escalate
```

Gumroad Field Mapping

Metadata Field	Gumroad Field	Validation
title	Product Name	Required, ≤255 chars
subtitle	Subtitle	Optional, ≤255 chars
description	Description	HTML allowed, ≤5000 chars
price_usd	Price	Required, ≥$0.01
categories	Categories	Max 2, from Gumroad taxonomy
tags	Tags	Max 10, from metadata.tags
pdf_path	Content File	Required, PDF only
cover_path	Cover Image	Required, 2560px wide min
isbn	Custom Field	Optional, validated format
metadata_package	Custom Fields	JSON blob for traceability

KDP Upload Pipeline (Automated)

Pipeline Architecture

```
# kdp_uploader.py - Core capabilities
class KDPUploader:
    def __init__(self, email, password, marketplace_id):
        self.browser = selenium.webdriver.Chrome(options=headless_options)
        self.auth = KDPAuth(email, password)
        self.marketplace = marketplace_id

    def upload_ebook(self, metadata_package, manuscript_path, cover_path):
        # 1. Login with 2FA (SMS/Authenticator)
        # 2. Navigate to Bookshelf → Create New Title → Kindle eBook
        # 3. Enter Details: title, subtitle, series, edition, author, contributors
        # 4. Enter Publishing Rights, Keywords (7), Categories (2)
        # 5. Upload Manuscript (DOCX/PDF) + Cover (JPG/PNG)
        # 6. Preview: Launch Kindle Previewer → Validate all devices
        # 7. Set Pricing: Territories, Royalty (70%/35%), Price, KDP Select
        # 8. Publish → Wait for "Live" status → Capture ASIN
        # 9. Return: asin, live_url, preview_url

    def validate_preview(self, asin, source_pdf):
        # Launch Kindle Previewer programmatically
        # Capture screenshots per device (Phone, Tablet, Kindle, Desktop)
        # Compare text extraction vs source
        # Return: PASS/FAIL + device_screenshots

    def handle_kdp_quirks(self):
        # Category selection: search + click (not dropdown)
        # Keyword entry: 7 separate fields, char limits
        # Previewer: iframe handling, wait for render
        # Price: per-territory matrix, VAT handling
```

KDP Field Mapping

Metadata Field	KDP Field	Validation
title	Book Title	Required, ≤200 chars
subtitle	Subtitle	Optional, ≤200 chars
author	Author Name	Required
description	Book Description	HTML allowed, ≤4000 chars
primary_keyword + 6 secondary	7 Keyword Fields	Each ≤50 chars, no duplicates
categories (2)	Browse Categories	Search + select, not free text
manuscript_path	Manuscript Upload	DOCX/PDF/EPUB, validated
cover_path	Cover Upload	JPG/PNG, 2560x1600 min
price_usd	List Price (US)	≥$2.99 for 70% royalty
kdp_select	KDP Select Enrollment	Boolean
drm_free	DRM	Boolean

Preview Verification (Automated)

Verification Protocol

```
def verify_preview(platform, product_id, source_pdf):
    if platform == "gumroad":
        return verify_gumroad_preview(product_id, source_pdf)
    elif platform == "kdp":
        return verify_kdp_preview(product_id, source_pdf)

def verify_gumroad_preview(product_id, source_pdf):
    # 1. Fetch preview pages via Gumroad preview URL
    # 2. Render each page to image (headless Chrome)
    # 3. Extract text from preview (OCR + PDF text)
    # 4. Compare vs source PDF text extraction
    # 5. Check: page count, text coverage >95%, images present
    # 6. Return: {pass: bool, coverage: float, diff_pages: [], screenshots: []}

def verify_kdp_preview(asin, source_pdf):
    # 1. Launch Kindle Previewer for ASIN
    # 2. Iterate devices: Phone, Tablet, Kindle, Desktop
    # 3. Per device: capture screenshots, extract text
    # 4. Check: TOC navigation, hyperlinks, images, tables, fonts
    # 5. Return: {pass: bool, device_results: {}, screenshots: {}}
```

Acceptance Criteria

Check	Gumroad	KDP
Page count match	□	□
Text coverage	≥95%	≥95%
Images rendered	□	□
Font embedding	□	□
TOC navigation	N/A	□
Hyperlinks	□	□
Table rendering	□	□
No truncation	□	□

Launch Asset Generator (GOAT KB #24, #38)

Assets Generated Per Launch

```
launch_assets:
  carousels:
    - platform: instagram
      slides: 5-7
      format: 1080x1080
      template: launch_carousel_v1
    - platform: linkedin
      slides: 5-7
      format: 1080x1080
      template: linkedin_carousel_v1
    - platform: facebook
      slides: 5-7
      format: 1080x1080
      template: fb_carousel_v1
  thread_visuals:
    - platform: x_twitter
      tweets: 10-15
      images: 10-15
      template: thread_visual_v1
  email_sequence:
    - email_1: launch_announcement (immediate)
    - email_2: behind_the_scenes (day_2)
    - email_3: social_proof (day_4)
    - email_4: objection_handling (day_7)
    - email_5: urgency_scarcity (day_10)
  funnel_map:
    - lead_magnet: chapter_1_pdf
    - tripwire: workbook_bundle ($7)
    - core: main_ebook ($27-97)
    - premium: course_bundle ($197-497)
    - continuity: membership ($27/mo)
  cross_sell_bundles:
    - bundle_1: ebook + workbook
    - bundle_2: ebook + audiobook
    - bundle_3: complete_series
  affiliate_pack:
    - swipe_copy: email, social, blog
    - banners: 300x250, 728x90, 1080x1080, 1200x628
    - tracking_links: per affiliate
    - commission_tiers: 30%/40%/50%
  press_kit:
    - one_pager: pdf
    - quotes: 5-10
    - images: cover, author, lifestyle
    - bio: short, medium, long
```

Asset Generation Pipeline

```
class LaunchAssetGenerator:
    def __init__(self, metadata_package, brand_kit):
        self.metadata = metadata_package
        self.brand = brand_kit  # colors, fonts, logos, templates

    def generate_all(self):
        assets = {}
        assets['carousels'] = self.generate_carousels()
        assets['threads'] = self.generate_thread_visuals()
        assets['emails'] = self.generate_email_sequence()
        assets['funnel'] = self.generate_funnel_map()
        assets['bundles'] = self.generate_cross_sell_bundles()
        assets['affiliate'] = self.generate_affiliate_pack()
        assets['press'] = self.generate_press_kit()
        return assets

    def generate_carousels(self):
        # Use faceless-carousel-creator, faceless-linkedin-carousel-creator
        # Template: Hook → Problem → MCD → Proof → Application → CTA
        # Brand compliance: colors, fonts, logo position, series mark
        pass
```

Funnel Integration

Funnel Architecture

```
LAUNCH TRAFFIC
      │
      ▼
LEAD MAGNET (Ch 1 PDF) ──► EMAIL CAPTURE
      │                          │
      ▼                          ▼
TRIPWIRE ($7 Workbook)    NURTURE SEQUENCE
      │                          │
      ▼                          ▼
CORE OFFER ($27-97 Ebook) ──► UPSELL (Course $197)
      │                          │
      ▼                          ▼
CROSS-SELL (Bundles)      CONTINUITY (Membership)
      │
      ▼
AFFILIATE PROGRAM ──► RECURRING REVENUE
```

Funnel Automation

```python
class FunnelIntegrator:
    def __init__(self, email_provider, payment_provider, analytics):
        self.email = email_provider  # ConvertKit, ActiveCampaign, etc.
        self.payment = payment_provider  # Stripe, Gumroad, etc.
        self.analytics = analytics  # Mixpanel, GA4, etc.

    def setup_launch_funnel(self, product_id, metadata_package):
        # 1. Create lead magnet delivery (Ch 1 PDF)
        # 2. Build tripwire page (workbook $7)
        # 3. Configure core offer page (ebook)
        # 4. Set up upsell (course bundle)
        # 5. Configure cross-sell bundles
        # 6. Create email sequences with tags/triggers
        # 7. Set up affiliate program (Gumroad/ThriveCart)
        # 8. Configure analytics events per funnel step
        pass
```

Affiliate Pack Distribution

Affiliate Pack Contents

```yaml
affiliate_pack:
  swipe_copy:
    email:
      - subject_lines: [5 variants]
      - body_templates: [3 variants: story, logic, urgency]
    social:
      - twitter_threads: [3 variants]
      - linkedin_posts: [3 variants]
      - instagram_stories: [5 frames]
    blog:
      - review_template: [structure + key points]
      - comparison_template: [vs competitors]
  visual_assets:
    banners:
      - 300x250 (rectangle)
      - 728x90 (leaderboard)
      - 1080x1080 (square)
      - 1200x628 (link share)
    badges:
      - "As seen in"
      - "Bestseller"
      - "Top rated"
  tracking:
    base_url: "https://gumroad.com/l/product-id"
    utm_template: "?utm_source=affiliate&utm_medium={medium}&utm_campaign=launch&utm_content={affiliate_id}"
    commission_tiers:
      - tier_1: 30% (0-10 sales)
      - tier_2: 40% (11-50 sales)
      - tier_3: 50% (51+ sales)
  compliance:
    ftc_disclosure: "Required: #ad or #affiliate"
    brand_guidelines: "No false claims, use approved copy only"
    payment_terms: "Net-30 after refund period"
```

Cross-Platform Sync (Price/Metadata Parity)

Sync Protocol

```
class CrossPlatformSync:
    def __init__(self, gumroad_uploader, kdp_uploader):
        self.gumroad = gumroad_uploader
        self.kdp = kdp_uploader

    def enforce_parity(self, product_id):
        # 1. Fetch live metadata from both platforms
        gumroad_meta = self.gumroad.get_product(product_id)
        kdp_meta = self.kdp.get_book(product_id)

        # 2. Compare critical fields
        checks = {
            'title': gumroad_meta.title == kdp_meta.title,
            'price_usd': abs(gumroad_meta.price - kdp_meta.price) < 0.01,
            'description_similarity': similarity(gumroad_meta.desc, kdp_meta.desc) > 0.9,
            'categories_aligned': self.check_category_mapping(gumroad_meta.cats, kdp_meta.cats),
            'cover_identical': hash(gumroad_meta.cover) == hash(kdp_meta.cover),
        }

        # 3. Auto-correct drift
        if not all(checks.values()):
            self.correct_drift(product_id, checks)

        return checks
```

GOAT KB Integration Map

GOAT KB	Launch Integration	Moussaif Enhancement
#24 Final Package Control	Pre-launch checklist, package validation	Destruction Court on complete package
#38 Launch Assets	Carousel, email, funnel, affiliate generation	Agentic generation + court validation
#22 PDF Preflight	Pre-upload validation	Automated pixel-diff preview verification
#23 Launch Assets	Asset templates	Brand compliance + court validation
#12 KDP/Gumroad SEO	Metadata optimization	Evidence-ranked keywords + court validation

Quick Load Commands

```
# Full launch pipeline
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat

# Launch day orchestration
hermes -s fugu-fusion-router,moussaif-orchestrator chat
```

Chapter Deliverables Checklist

- □ Kernel card (`launch-kernel.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for launch domain
- □ All launch claims epistemic-tagged
- □ Destruction Court log (5 courts) for launch execution
- □ Source trace chain for every platform claim
- □ HyperSkill card for any new launch capability
- □ Reconstruction memory entry

- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry, daily_generation_report)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment
- □ Gumroad product live + KDP product live
- □ Preview verification PASS on both platforms
- □ Launch assets deployed (carousels, emails, funnels, affiliate packs)
- □ Cross-platform parity enforced
- □ Analytics dashboard configured
- □ Affiliate program live with tracking

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0026..0032 (Image/Video Orchestration - launch assets)
- CORE-0033..0039 (Video Orchestration - book trailers)
- CORE-0040..0045 (Cost Policy - launch budget management)
- CORE-0046..0051 (Quality Policy - launch quality dimensions)
- HS-003 (Evidence Ranking Hyperskill - platform claims)
- OP-001..120 (Operations - upload, verify, asset generation, funnel)

DSL Record Types Used in This Chapter

- **RT** - Routing table (model selection for browser automation vs API)
- **HS** - Hyperskill definitions (ebook-publishing-gumroad-kdp, moussaif-launch-asset-generator)
- **OP** - Operations (gumroad_upload, kdp_upload, preview_verify, asset_generate, funnel_connect)
- **CORE** - Subsystem mandates (tool-security, credit-accounting, quality-scoring, latency-control)
- **EV** - Evaluation (launch success metrics scoring)

Chapter 12 Execution Command

```
hermes -s ebook-publishing-gumroad-kdp,moussaif-cover-design-system,moussaif-metadata-seo-engine,moussaif-pdf-re
    nder-qa,moussaif-launch-asset-generator,moussaif-source-trace-qa,moussaif-safety-review-gate chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - All platform fields/API behaviors traced to official docs
2. **Reductionism** - Full DAG executed: upload → verify → assets → funnel → affiliate
3. **False Novelty** - Prior art (GOAT #24, #38) referenced, delta documented
4. **Weak Op** - Agent handles 2FA, rate limits, preview diff, price sync
5. **Agentic Failure** - Agent monitors policy, rankings, affiliate links

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

13. SAFETY REVIEW GATE - ISLAMIC/LIFE/MEDICAL/LEGAL/FINANCIAL/TECHNICAL FRAMING

Subtitle: Compliance Without Compromise: Epistemic Tagging Meets Regulatory Boundaries **Focus:** GOAT KB #21, #22 + moussaif-safety-review-gate skill **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Chapter:** 13 of 15 **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/13_safety_review_gate_islamic_life_medical_legal_financial.md`

Executive Summary

This chapter implements the **Safety Review Gate** as a HyperSkill: domain-specific framing protocols that ensure every knowledge product complies with Islamic principles, life/medical safety, legal accuracy, financial regulation, and technical correctness - without diluting the product's value or claiming expertise it doesn't possess. All outputs pass Destruction Courts and carry full epistemic traceability.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft → Critique → Finalize.*

Moussaif AGI Integration Map

AGI Component	Safety Review Application
Supreme Architect (Nemotron-3-Ultra)	Final safety clearance, precedent-setting edge cases
Deep Reasoner (Nemotron-3-Super)	Multi-jurisdictional legal analysis, fatwa reconciliation
Coding Agent (Laguna-M-1)	Safety rule engine, automated flagging, compliance dashboards
Fast Coder (Laguna-XS)	Quick rule additions, pattern updates, regex refinements
Research Synthesizer (Hunyuan-3B)	Regulatory monitoring, scholarly source verification, precedent tracking
Critic Court (GPT-OSS-120B)	Hallucinated legal advice, missing disclaimers, overreach detection
Compression Agent (GPT-OSS-20B)	Disclaimer condensation, warning label optimization
Planner (Gemma-4-26B)	Safety workflow: scan → flag → route → review → clear → document
Executor (Qwen3-Coder)	Automated scanning, tag injection, disclaimer insertion, report generation
Refiner (North-Mini-Code)	Framing polish, tone consistency, cross-domain alignment
Multilingual (Gemma-4-26B)	Arabic fatwa sources, French legal codes, multi-jurisdiction
Visual Director (Nemotron-Nano-Omni)	Safety icon system, warning label design, compliance badges
Video Director (Nemotron-Nano-Omni)	Safety briefing videos, compliance walkthroughs
SEO/Growth (Hunyuan-3B)	Safe keyword selection, compliant funnel language
Outreach (GPT-OSS-20B)	Partner compliance verification, affiliate safety briefings
Data Extractor (Laguna-XS)	Regulatory change extraction, citation harvesting
Memory Curator (Gemma-4-31B)	Precedent database, ruling evolution, scholar opinion tracking
Tool-Use Agent (Laguna-M-1)	Legal API orchestration, fatwa database queries, regulatory feeds
Low-Cost Draft (GPT-OSS-20B)	First-pass safety scans, disclaimer drafts, framing variants
Premium Final (Nemotron-3-Ultra)	Final safety synthesis, edge case resolution, scholarly sign-off

8 HyperLoops Applied to Safety Review

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Product domain + regulatory landscape + scholarly sources **Output:** `safety-kernel.yaml` with governing MCD **Validation:** Kernel regenerates 80%+ of safety decisions **Safety MCD:** *"The governing difference between responsible knowledge transfer and harmful misinformation is the precision of the domain-boundary-disclaimer triad. If any boundary is blurred, the product becomes liability instead of asset."*

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Regulatory texts + scholarly opinions + precedent cases + platform policies **Output:** Epistemic tags on every safety claim **Validation:** Zero `ESTABLISHED` safety claims without primary source (statute, fatwa, guideline)

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (advice, guidance, information, education, opinion, fatwa, ruling, disclaimer, warning) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Apply safety framing to [product] in [domain]" **Output:** Task DAG: Scan → Identify Domain → Select Framing → Insert Disclaimers → Verify Boundaries → Document **Validation:** Agent executes each leaf without human intervention

Safety Task DAG:

```
DOMAIN IDENTIFICATION
       │
       ├─► ISLAMIC ──► SCHOLARLY SOURCE → FATWA CHECK → FRAMING → DISCLAIMER
       │
       ├─► MEDICAL/LIFE ──► LICENSE CHECK → EVIDENCE GRADE → FRAMING → DISCLAIMER
       │
       ├─► LEGAL ──► JURISDICTION → STATUTE/CASE → FRAMING → DISCLAIMER
       │
       ├─► FINANCIAL ──► REGULATOR → RULE/EXEMPTION → FRAMING → DISCLAIMER
       │
       └─► TECHNICAL ──► STANDARD → SPEC/TEST → FRAMING → DISCLAIMER
```

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Framed content + disclaimers + boundary markers **Courts:**

1. **Hallucination** - Invented statutes? Fake fatwas? Misquoted guidelines? Hallucinated case law?
2. **Reductionism** - "Consult professional" without domain specificity? Conflated legal/medical/financial?
3. **False Novelty** - "Revolutionary safety framework" = standard disclaimer + domain label?
4. **Weak Op** - Can agent detect missing disclaimer? Route to correct domain reviewer?
5. **Agentic Failure** - Would agent catch jurisdiction mismatch better than human?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair → re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated framing patterns **Output:**

- `safety_scanner.py` - Regex + LLM hybrid for domain detection
- `disclaimer_injector.py` - Context-aware disclaimer insertion
- `compliance_checker.py` - Rule engine for each domain
- `fatwa_query.py` - Scholarly database interface
- `regulatory_monitor.py` - RSS/API feeds for rule changes

Validation: 10x speedup vs manual review

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Framed content + safety report + audit trail **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New rulings, regulatory changes, scholarly opinions, platform policy updates **Output:** Versioned framing rules + reconstruction memory entry **Validation:** Regression test on prior safety decisions

---

5 Destruction Courts - Safety-Specific Checks

Court	Safety Failure Modes	Pass Criteria
Hallucination	Invented Quranic verses, fake hadith, fabricated statutes, invented case citations, hallucinated FDA/EMA guidelines, fake fatwa references	Every religious/legal/medical/financial claim traced to primary source with URL/accession number
Reductionism	"Not legal advice" on medical content, "Consult scholar" on financial advice, conflated jurisdictions, universal disclaimer	Domain-specific framing with precise boundary: "This is educational Islamic finance information, not a fatwa. Consult a qualified mufti for personal rulings."
False Novelty	"AI-powered compliance" = keyword filter, "Scholar-verified" = keyword match, "Regulatory intelligence" = RSS feed	Prior art referenced (GOAT #21, #22), delta documented, kernel genuinely regenerates domain
Weak Op -	Agent misses missing disclaimer, routes medical to legal reviewer, cannot detect jurisdiction mismatch	Agent detects domain, selects correct framing template, inserts disclaimer, flags boundary edge cases
Agentic Failure	Human catches missed disclaimer, human detects outdated ruling, human spots jurisdiction error	Agent monitors regulatory feeds, tracks scholarly opinion evolution, flags jurisdiction drift

---

Epistemic Tagging Schema (Mandatory Per Safety Claim)

```
claim: "This chapter describes zakat calculation principles per Hanafi school"
confidence: ESTABLISHED
source: "Al-Mabsut, Al-Sarakhsi, Vol 3, Kitab al-Zakat; verified via Makkah API 2024"
centrality: core
relation_type: supports
skill_relevance: moussaif-safety-review-gate
tool_relevance: islamic-finance-tools
risk_level: medium  # Religious ruling - requires scholar verification for personal use
version: 1.0.0
update_status: current
domain: islamic
school: hanafi
verification_required: true  # Personal application needs mufti
```

Tags: `ESTABLISHED` | `SUPPORTED` | `PLAUSIBLE` | `THEORETICAL` | `HYPOTHESIS` | `SPECULATIVE` | `UNKNOWN` | `NEEDS_VERIFICATION`

Domain-Specific Framing Protocols

1. ISLAMIC FRAMING (Sharia Compliance)

Governing Principle

> **All content touching Islamic law, finance, worship, ethics, or history must carry precise school attribution, scholarly chain, and personal-use disclaimer.**

Framing Template

```
> **Islamic Framing Notice:** This content presents [topic] according to the [School:
    Hanafi/Maliki/Shafi'i/Hanbali/Ja'fari] school as derived from [Primary Source: Quran/Quranic verse/Hadith
    collection/Fiqh text]. It represents scholarly interpretation, not a binding fatwa. For personal religious
    rulings, consult a qualified mufti who can assess your specific circumstances. The author and publisher are
    not responsible for actions taken based on this educational content.
```

Scholarly Source Hierarchy

Tier	Sources	Confidence
1 - Primary	Quran (verified via Tanzil.net), Sahih Bukhari/Muslim (via Sunnah.com), Muwatta Malik	ESTABLISHED
2 - Classical Fiqh	Al-Mabsut, Al-Umm, Al-Mughni, Bada'i al-Sana'i, Minhaj al-Talibin	ESTABLISHED
3 - Fatwa Bodies	Al-Azhar, Islamqa.info, SeekersGuidance, Dar al-Ifta (verified via API)	SUPPORTED
4 - Contemporary Scholars	Verified living scholars with ijazah chains	SUPPORTED/PLAUSIBLE
5 - Academic	Peer-reviewed Islamic studies journals	SUPPORTED
6 - Unverified	Blogs, forums, social media	SPECULATIVE/UNKNOWN

Domain-Specific Rules

Domain	Framing Requirement
Islamic Finance	School + AAOIFI standard + "Not a fatwa - consult mufti for transactions"
Worship (Ibadah)	School + evidence (dalil) + "Validity depends on conditions - verify with scholar"
Family Law	School + jurisdiction + "Court ruling supersedes - consult lawyer"
Ethics/Akhlaq	Source + "General guidance - personal application varies"
History/Seerah	Primary source chain (isnad) + "Historical account - not prescriptive"

---

2. MEDICAL/LIFE SCIENCES FRAMING

Governing Principle

> **No content may be construed as medical advice, diagnosis, treatment recommendation, or substitute for licensed professional care.**

Framing Template

```
> **Medical Disclaimer:** This content is for educational and informational purposes only. It does not
    constitute medical advice, diagnosis, or treatment. The information presented is based on [source:
    peer-reviewed literature/clinical guidelines/textbooks] as of [date]. Medical knowledge evolves rapidly.
    Always consult a licensed physician or qualified healthcare provider for any health concerns, before
    starting any treatment, or before making changes to existing treatment. The author and publisher are not
    liable for any adverse outcomes resulting from use of this information.
```

Evidence Grading (Adapted from GRADE)

Grade	Source	Framing
A	Systematic reviews, RCTs, major society guidelines (AHA, ACC, ADA, WHO)	"Strong evidence supports..."
B	Cohort studies, case-control, minor society guidelines	"Evidence suggests..."
C	Case series, expert opinion, consensus statements	"Expert opinion indicates..."
D	Preclinical, theoretical, anecdotal	"Preliminary data suggests... (not clinically validated)"

Domain-Specific Rules

Domain	Framing Requirement
Nutrition/Supplements	"Not evaluated by FDA. Not intended to diagnose, treat, cure, prevent disease."

Domain	Framing Requirement
Mental Health	"Not a substitute for therapy. Crisis: 988 (US) / local emergency."
Fitness/Exercise	"Consult physician before starting. Risk of injury exists."
First Aid/Emergency	"Not a substitute for EMS. Call emergency services immediately."
Pharmacology	"Prescription only. Dosing requires physician. Interactions possible."

---

3. LEGAL FRAMING

Governing Principle

> **No content may constitute legal advice, create attorney-client relationship, or substitute for licensed counsel in the relevant jurisdiction.**

Framing Template

```
> **Legal Disclaimer:** This content provides general legal information based on [jurisdiction: US
    Federal/State/Country] law as of [date]. It does not constitute legal advice, does not create an
    attorney-client relationship, and should not be relied upon for legal decisions. Laws vary by jurisdiction
    and change frequently. For any specific legal matter, consult a licensed attorney in the relevant
    jurisdiction. The author and publisher are not liable for actions taken based on this information.
```

Jurisdiction Matrix

Jurisdiction	Primary Sources	Update Frequency
US Federal	USC, CFR, Federal Register, Supreme Court opinions	Real-time via CourtListener
US State (50)	State statutes, admin codes, appellate opinions	Weekly per state
EU/UK	EUR-Lex, UK Legislation, ECJ/UKSC opinions	Weekly
International	UN treaties, WTO, ICC, ICJ	Monthly

Domain-Specific Rules

Domain	Framing Requirement
Contracts	"Template only. Requires attorney review for enforceability in your jurisdiction."
IP/Copyright	"Fair use is fact-specific. Register copyrights. Consult IP attorney."
Business Formation	"State-specific. Tax implications vary. Consult CPA + attorney."
Employment	"Federal + state + local laws apply. At-will exceptions vary."
Privacy/GDPR	"DPO required for certain processors. DPIA for high-risk. Legal review essential."

---

4. FINANCIAL FRAMING

Governing Principle

> **No content may constitute investment advice, financial advice, tax advice, or solicitation. All financial content is educational.**

Framing Template

```
> **Financial Disclaimer:** This content is for educational purposes only and does not constitute investment
    advice, financial advice, tax advice, or a solicitation to buy/sell any security or financial product. All
    investments involve risk, including loss of principal. Past performance does not guarantee future results.
    Consult a licensed financial advisor, CPA, or tax attorney for advice specific to your situation. The author
    and publisher are not registered investment advisors and are not liable for financial decisions made based
    on this content.
```

Regulatory Frameworks

Regulator	Domain	Key Rules
SEC	Securities	No unregistered offers, no guarantees, accurate performance
FINRA	Broker-dealer	Fair, balanced, not misleading
CFTC	Derivatives	Leverage disclosure, risk warnings
IRS	Tax	Not tax advice. Consult CPA/EA.
State	Insurance/Banking	Licensing required for advice

Domain-Specific Rules

Domain	Framing Requirement
Investing/Trading	"Not investment advice. High risk. Can lose entire investment."
Crypto/DeFi	"Unregulated. High volatility. Smart contract risk. Not FDIC insured."
Tax Strategy	"General principles. Your facts determine outcome. Consult CPA."
Retirement	"Rules change. Age limits apply. Penalty for early withdrawal."
Real Estate	"Market varies. Leverage risk. Consult agent + attorney + CPA."

---

5. TECHNICAL FRAMING

Governing Principle

> **Technical content must distinguish between: established standards, vendor specifications, theoretical capabilities, and verified implementations.**

Framing Template

```
> **Technical Notice:** This content describes [technology/standard/protocol] as specified in [source:
    RFC/IEEE/ISO/vendor doc] version [x.y] as of [date]. Implementation details may vary by platform, version,
    and configuration. Performance claims are based on [benchmarks/specifications/theoretical maximums] and may
    not reflect real-world results. Always test in your environment. The author and publisher are not
    responsible for system failures, data loss, or security incidents resulting from implementation of this
    information.
```

Specification Hierarchy

Tier	Sources	Confidence
1 - Standards	RFC (IETF), IEEE, ISO, W3C, OASIS	ESTABLISHED
2 - Vendor Specs	Official docs (AWS, Azure, GCP, Kubernetes, Linux kernel)	SUPPORTED
3 - White Papers	Vendor architecture papers, reference architectures	PLAUSIBLE
4 - Community	GitHub issues, Stack Overflow, blogs, forums	SPECULATIVE/UNKNOWN

Domain-Specific Rules

Domain	Framing Requirement
Security/Crypto	"Not a security audit. Use audited libraries. Rotate keys. Threat model first."
AI/ML	"Model limitations: bias, hallucination, drift. Validate on your data. Human-in-loop."
Cloud/DevOps	"Default configs insecure. Least privilege. Test disaster recovery. Cost monitoring."
Data/Privacy	"PII handling per GDPR/CCPA. Encryption at rest/transit. Retention policies."
Blockchain/Smart Contracts	"Unaudited code = risk. Testnet first. Formal verification recommended. Gas optimization ≠ security."

Cross-Domain Safety Matrix

Content Touches	Required Framings	Reviewer
Islamic + Finance	Islamic + Financial	Mufti + CFA
Medical + Legal	Medical + Legal	MD + JD
Financial + Legal	Financial + Legal	CFA + JD
Technical + Legal	Technical + Legal	PE + JD
Islamic + Medical	Islamic + Medical	Mufti + MD

Rule: When domains overlap, ALL applicable framings apply. Most restrictive standard wins.

GOAT KB Integration Map

GOAT KB	Safety Integration	Moussaif Enhancement
#21 Safety Reviews	Domain-specific review protocols	Epistemic tagging as framing enforcer
#22 Islamic/Life/Medical/Legal/Financial/Technical Framing	Detailed framing templates per domain	Destruction Court on framing completeness
#23 PDF Preflight	Safety metadata injection	Automated disclaimer injection + court validation
#13 Source Trace QA	Safety source chain	Full reconstruction memory for rulings

---

Quick Load Commands

```
# Safety review pipeline
hermes -s moussaif-safety-review-gate,moussaif-agi-philosophy,moussaif-orchestrator chat

# Domain-specific safety
hermes -s moussaif-safety-review-gate,moussaif-fiqh-compliance,moussaif-medical-framing,moussaif-legal-framing,m
    oussaif-financial-framing,moussaif-technical-framing chat
```

Chapter Deliverables Checklist

- □ Kernel card (`safety-kernel.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for safety domain boundaries
- □ All safety claims epistemic-tagged with domain + source
- □ Destruction Court log (5 courts) for framing completeness
- □ Source trace chain for every religious/legal/medical/financial/technical claim
- □ HyperSkill card for any new framing capability
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment
- □ All 5 domain templates implemented and tested
- □ Cross-domain matrix validated
- □ Scholarly/regulatory source database current
- □ Disclaimer injection automated and verified

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0013..0015 (Self-Improvement - safety rule updates)
- CORE-0046..0051 (Quality Policy - multi-dimensional safety quality)
- CORE-0145..0178 (AGI Fattah Orientation - safety as blocked problem space)
- HS-001..004 (Hyperskill Invariant - safety gate as capability)
- MEM-001..007 (Memory Architecture - negative memory for safety failures)
- CORE-0145..0178 (Image/Video direction - safety visual language)

DSL Record Types Used in This Chapter

- **HS** - Hyperskill definitions (moussaif-safety-review-gate, domain-specific framings)
- **CORE** - Subsystem mandates (secret-redaction, tool-security, quality-scoring, incident-response)
- **EV** - Evaluation (safety scoring, risk level assessment)
- **OP** - Operations (scan, flag, route, review, clear, document)
- **MEM** - Memory (negative memory for safety failures, procedural for framing rules)

Chapter 13 Execution Command

```
hermes -s moussaif-safety-review-gate,moussaif-agi-philosophy,moussaif-orchestrator chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - Every religious/legal/medical/financial/technical claim traced to primary source
2. **Reductionism** - Domain-specific framings, not universal disclaimers; boundaries precise
3. **False Novelty** - Prior art (GOAT #21, #22) referenced, delta documented
4. **Weak Op** - Agent detects domain, selects framing, inserts disclaimer, flags edges
5. **Agentic Failure** - Agent monitors regulatory feeds, scholarly opinions, jurisdiction drift

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

14. SOURCE TRACE QA - ORIGINAL → CLEANED → TOPIC → USAGE CHAIN

Subtitle: Full Provenance from Raw Knowledge to Published Asset with Reconstruction Memory **Focus:** GOAT KB #14 + HyperKB reconstruction memory integration **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Chapter:** 14 of 15 **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/14_source_trace_qa_original_cleaned_topic_usage_chain.md`

Executive Summary

This chapter implements **Source Trace QA** as a HyperSkill: an unbroken chain of provenance from raw source ingestion through cleaning, topic mapping, usage logging, to reconstruction memory - enabling any published claim to be traced to its origin, every edit to be audited, and every version to be reconstructed. This is the immune system of the knowledge product.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft → Critique → Finalize.*

Moussaif AGI Integration Map

AGI Component	Source Trace Application
Supreme Architect (Nemotron-3-Ultra)	Final traceability audit, schema evolution decisions
Deep Reasoner (Nemotron-3-Super)	Causal chain analysis: source → decision → outcome
Coding Agent (Laguna-M-1)	Trace pipeline: ingestion, cleaning, chunking, linking, logging
Fast Coder (Laguna-XS)	Quick schema fixes, index updates, query optimizations
Research Synthesizer (Hunyuan-3B)	Source verification, citation harvesting, authority scoring
Critic Court (GPT-OSS-120B)	Broken chains, orphaned chunks, circular references, missing provenance
Compression Agent (GPT-OSS-20B)	Trace log condensation, summary generation, diff compression
Planner (Gemma-4-26B)	Trace workflow: ingest → clean → chunk → tag → link → log → monitor
Executor (Qwen3-Coder)	Pipeline execution, index updates, alert triggers, report generation
Refiner (North-Mini-Code)	Query optimization, UI polish, cross-reference validation
Multilingual (Gemma-4-26B)	Multi-language source tracking, translation provenance
Visual Director (Nemotron-Nano-Omni)	Trace visualization: lineage graphs, impact maps, version trees
Video Director (Nemotron-Nano-Omni)	Trace audit walkthroughs, reconstruction demos
SEO/Growth (Hunyuan-3B)	Source quality metrics, citation impact, trust signals
Outreach (GPT-OSS-20B)	Contributor attribution, collaborator credit, citation exports
Data Extractor (Laguna-XS)	Structured provenance extraction: DOIs, ORCIDs, handles, versions
Memory Curator (Gemma-4-31B)	Reconstruction memory: kernel evolution, concept drift, version genealogy
Tool-Use Agent (Laguna-M-1)	Source APIs (Crossref, PubMed, arXiv, GitHub, legal DBs), citation managers
Low-Cost Draft (GPT-OSS-20B)	First-pass trace logs, provenance summaries, gap reports
Premium Final (Nemotron-3-Ultra)	Final traceability certification, schema governance, audit sign-off

8 HyperLoops Applied to Source Trace QA

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Source corpus + trace requirements + compliance needs **Output:** `trace-kernel.yaml` with governing MCD **Validation:** Kernel regenerates 80%+ of trace decisions **Trace MCD:** *"The governing difference between a knowledge product that can be audited and one that cannot is the continuity of the original→cleaned→topic→usage chain. If any link breaks, the product becomes unverifiable and untrustworthy."*

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Source metadata + cleaning logs + topic assignments + usage records **Output:** Epistemic tags on every trace link **Validation:** Zero `ESTABLISHED` trace links without verifiable source artifact

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (source, origin, provenance, lineage, pedigree, audit trail, chain of custody, version history, change log, diff, blame, annotation) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Establish full provenance for [knowledge product]" **Output:** Task DAG: Ingest → Clean → Chunk → Tag → Link → Log → Monitor → Reconstruct **Validation:** Agent executes each leaf without human intervention

Trace Task DAG:



Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Trace pipeline execution + trace database **Courts:**

1. **Hallucination** - Invented sources? Fake DOIs? Fabricated cleaning logs? Phantom usage records?
2. **Reductionism** - Provenance = just filename? Lost cleaning context? Flattened topic hierarchy?
3. **False Novelty** - "Blockchain provenance" = appended hash? "AI traceability" = filename + timestamp?
4. **Weak Op** - Agent cannot reconstruct claim from trace? Cannot find orphaned chunks?
5. **Agentic Failure** - Would agent detect silent source corruption better than human?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair → re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated trace patterns **Output:**

- `source_ingestion.py` - Multi-source fetcher with hash verification

- `cleaning_pipeline.py` - Normalize, dedupe, redact, schema-validate with diff logging
- `chunker_tagger.py` - Semantic chunker + epistemic tagger + GraphRAG linker
- `topic_mapper.py` - Concept assignment + MCD linking + skill mapping
- `usage_logger.py` - Fine-grained usage: chapter, paragraph, claim, skill, loop
- `trace_monitor.py` - Staleness alerts, drift detection, orphan finder, conflict detector
- `reconstructor.py` - Kernel evolution, concept drift, version genealogy, blame

Validation: 10x speedup vs manual trace management

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Complete trace database + monitoring alerts + reconstruction reports **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** New sources, cleaning improvements, topic refinements, usage patterns, kernel updates **Output:** Versioned trace schema + reconstruction memory entry + schema migration **Validation:** Regression test on prior trace queries

---

5 Destruction Courts - Trace-Specific Checks

Court	Trace Failure Modes	Pass Criteria
Hallucination	Invented source URLs, fake DOIs/ISBNs, fabricated cleaning operations, phantom usage logs, invented chunk IDs	Every trace link resolves to verifiable artifact (file, API response, log entry, hash)
Reductionism	Provenance = filename only; cleaning = "cleaned"; topic = single tag; usage = "used in book"	Full chain: source → hash → cleaning ops → chunks → tags → concepts → MCDs → skills → usage locations
False Novelty	"Immutable provenance" = appended hash; "AI traceability" = timestamp + filename; "Blockchain audit" = SHA256 in filename	Prior art (Git, DVC, MLflow, DataLineage) referenced, delta documented, kernel genuinely regenerates trace
Weak Op	Agent cannot answer "Where did this claim come from?"; cannot find all usages of a source; cannot reconstruct v1.0	Agent answers provenance queries, finds all usages, reconstructs any version, detects drift
Agentic Failure	Human finds broken link, human detects silent source change, human notices orphaned chunk	Agent monitors source hashes, alerts on drift, finds orphans, suggests repairs, predicts staleness

---

Epistemic Tagging Schema (Mandatory Per Trace Link)

```
trace_link:
  chunk_id: "afripa-kb-2025-001-042"
  source_ref: "02_SERVICE_PORTFOLIO.md#content-engine"
  source_hash: "sha256:a1b2c3d4..."
  cleaning_ops: ["normalize_whitespace", "remove_headers", "dedupe_paragraphs"]
  cleaning_hash: "sha256:e5f6g7h8..."
  chunks:
    - id: "afripa-kb-2025-001-042-001"
      content: "The governing difference between a content engine that compounds..."
      epistemic_tag: ESTABLISHED
      domain: content-engine
      concept: MCD-content-engine
      centrality: core
      relations:
        - {target: "MCD-funnel", type: "supports"}
        - {target: "seo-topic-cluster", type: "generates"}
      skills: ["moussaif-seo-content-refresh", "faceless-content-calendar-automator"]
      tools: ["content-engine-tools"]
      risk_level: low
      version: "1.0.0"
      extracted: "2025-07-10T14:30:00Z"
      extractor: "kb_extraction_sop"
      update_status: current
  usage_log:
    - {chapter: "04_CHANNELS_SOCIAL/01_FACELESS_CONTENT_OS.md", location: "paragraph 3", claim: "MCD statement",
       skill: "faceless-content-os"}
    - {skill: "moussaif-seo-content-refresh", usage: "keyword clustering", timestamp: "2025-07-10T15:00:00Z"}
    - {loop: "hyperloop-seo-keyword-cluster-expansion-v001", usage: "evidence", timestamp:
       "2025-07-10T15:30:00Z"}
```

Source Trace Chain Architecture

Four-Link Chain

```
ORIGINAL SOURCE
      │
      ▼ (ingest + hash + verify)
CLEANED SOURCE
      │
      ▼ (chunk + tag + link)
TOPIC-MAPPED CHUNKS
      │
      ▼ (usage log + monitor)
USAGE-LOGGED ASSETS
```

Link 1: ORIGINAL → CLEANED

Artifact	Schema
Source Manifest	`source_id, source_type, url/path, fetched_at, fetched_by, hash_sha256, signature, metadata`
Cleaning Log	`source_id, operation, agent, timestamp, params, input_hash, output_hash, diff_summary`
Cleaned Artifact	`cleaned_id, source_id, content, cleaning_hash, schema_version`

Link 2: CLEANED → TOPIC-MAPPED

Artifact	Schema
Chunk Index	`chunk_id, cleaned_id, content, token_count, start_char, end_char, headers_preserved`
Epistemic Tags	`chunk_id, tag (ESTABLISHED..NEEDS_VERIFICATION), confidence_score, tagger, timestamp`
Concept Map	`concept_id, name, definition, necessary_conditions, sufficient_conditions, domain`
GraphRAG Edges	`source_chunk, target_chunk, relation_type (supports, contradicts, refines, generates, operationalizes, tests, trains, retrieves, updates), weight, evidence`
MCD Links	`chunk_id, mcd_id, role (kernel, evidence, application, boundary)`

Link 3: TOPIC-MAPPED → USAGE-LOGGED

Artifact	Schema
Usage Log	`usage_id, chunk_id, context (chapter, skill, loop, asset, claim), location, timestamp, agent, purpose`
Asset Manifest	`asset_id, asset_type (chapter, cover, funnel, email, carousel), chunks_used, skills_used, loops_used, version`
Skill Manifest	`skill_id, chunks_used, concepts_used, mcds_used, version, last_validated`
Loop Manifest	`loop_id, chunks_used, skills_used, evidence_used, version, last_executed`

Reconstruction Memory (The "Time Machine")

Purpose

Track how knowledge evolves so any version can be reconstructed and reasoning traced.

Entry Schema

```
## Reconstruction Memory: 2025-07-10

### Kernels Updated
- `content-engine`: Refined MCD to emphasize loop integrity over output volume
- `seo-topic-cluster`: Added "intent drift" as boundary condition

### Concepts Refined
- `faceless-content-os`: Split into core OS + platform adapters
- `ebook-factory`: Separated PDF render from metadata engine

### Tools Versioned
- `pdf_preflight.py` → v1.3.0 (added transparency flattening check)
- `keyword_cluster.py` → v2.1.0 (supports 7 KW + 2 cat schema)

### Patterns Learned
- Cover thumbnail test protocol reduces KDP rejection by 40%
- Agent dry-run catches 80% of Weak Op failures pre-court

### Failed Patterns (Logged to failure_registry)
- QMCD superposition on highly subjective domains (brand voice) → low agreement
- GraphRAG on sparse domains (<100 chunks) → noise dominates

### Next Ingestion Priorities
1. Client deliverables from Q2 projects
2. Updated platform API docs (TikTok, YouTube, Instagram)
3. New GOAT KB releases (monthly)
```

Reconstruction Capabilities

Query	Capability
"What was the MCD for content-engine in v1.2?"	Exact kernel card retrieval
"Show all chunks that contributed to chapter 4 in v1.0"	Full usage trace + chunk content
"Why did the SEO keyword for 'AI agents' change in v2.1?"	Evidence log + drift detection
"Reconstruct the exact skill set for faceless-content-os v1.5"	Skill manifest + chunk dependencies
"What would break if we update the PDF preflight tool?"	Impact analysis via usage graph

---

Trace Database Schema

Tables

```
-- Sources
CREATE TABLE sources (
    source_id TEXT PRIMARY KEY,
    source_type TEXT, -- pdf, api, web, manual, transcript
    url_or_path TEXT,
    fetched_at TIMESTAMP,
    fetched_by TEXT,
    hash_sha256 TEXT,
    signature TEXT,
    metadata JSONB,
    status TEXT -- active, deprecated, superseded
);

-- Cleaning Operations
CREATE TABLE cleaning_ops (
    op_id TEXT PRIMARY KEY,
    source_id TEXT REFERENCES sources(source_id),
    operation TEXT, -- normalize, dedupe, redact, validate, restructure
    agent TEXT, -- human, script:name, llm:model
    timestamp TIMESTAMP,
    params JSONB,
    input_hash TEXT,
    output_hash TEXT,
    diff_summary TEXT
);

-- Cleaned Artifacts
CREATE TABLE cleaned_artifacts (
    cleaned_id TEXT PRIMARY KEY,
    source_id TEXT REFERENCES sources(source_id),
    content_hash TEXT,
    cleaning_hash TEXT,
    schema_version TEXT,
    created_at TIMESTAMP
);

-- Chunks
CREATE TABLE chunks (
    chunk_id TEXT PRIMARY KEY,
    cleaned_id TEXT REFERENCES cleaned_artifacts(cleaned_id),
    content TEXT,
    token_count INT,
    start_char INT,
    end_char INT,
    headers JSONB,
    created_at TIMESTAMP
);

-- Epistemic Tags
CREATE TABLE epistemic_tags (
    tag_id TEXT PRIMARY KEY,
    chunk_id TEXT REFERENCES chunks(chunk_id),
    tag TEXT, -- ESTABLISHED, SUPPORTED, PLAUSIBLE, THEORETICAL, HYPOTHESIS, SPECULATIVE, UNKNOWN,
        NEEDS_VERIFICATION
```

```
    confidence_score FLOAT,
    tagger TEXT,
    timestamp TIMESTAMP,
    evidence_refs JSONB
);

-- Concepts
CREATE TABLE concepts (
    concept_id TEXT PRIMARY KEY,
    name TEXT,
    definition TEXT,
    necessary_conditions JSONB,
    sufficient_conditions JSONB,
    domain TEXT,
    created_at TIMESTAMP
);

-- GraphRAG Edges
CREATE TABLE graph_edges (
    edge_id TEXT PRIMARY KEY,
    source_chunk_id TEXT REFERENCES chunks(chunk_id),
    target_chunk_id TEXT REFERENCES chunks(chunk_id),
    relation_type TEXT, -- supports, contradicts, refines, generates, operationalizes, tests, trains, retrieves,
        updates
    weight FLOAT,
    evidence JSONB
);

-- MCD Links
CREATE TABLE mcd_links (
    link_id TEXT PRIMARY KEY,
    chunk_id TEXT REFERENCES chunks(chunk_id),
    mcd_id TEXT,
    role TEXT, -- kernel, evidence, application, boundary
    strength FLOAT
);

-- Usage Log
CREATE TABLE usage_log (
    usage_id TEXT PRIMARY KEY,
    chunk_id TEXT REFERENCES chunks(chunk_id),
    context_type TEXT, -- chapter, skill, loop, asset, claim
    context_id TEXT,
    location TEXT,
    timestamp TIMESTAMP,
    agent TEXT,
    purpose TEXT
);

-- Assets
CREATE TABLE assets (
    asset_id TEXT PRIMARY KEY,
    asset_type TEXT, -- chapter, cover, funnel, email, carousel, skill, loop
    chunks_used JSONB, -- [chunk_id]
    skills_used JSONB, -- [skill_id]
    loops_used JSONB, -- [loop_id]
    version TEXT,
    created_at TIMESTAMP,
    status TEXT -- draft, published, archived
);

-- Reconstruction Memory
CREATE TABLE reconstruction_memory (
    entry_id TEXT PRIMARY KEY,
    date DATE,
    kernels_updated JSONB,
    concepts_refined JSONB,
    tools_versioned JSONB,
    patterns_learned JSONB,
    failed_patterns JSONB,
    next_priorities JSONB
);
```

GOAT KB Integration Map

GOAT KB	Trace Integration	Moussaif Enhancement
#14 Source Trace QA	Original→Cleaned→Topic→Usage chain	Full reconstruction memory + DSL trace
#13 Source Trace QA	Source chain methodology	Epistemic tagging as trace enforcer
#6 KB Engineering	KB build pipeline	Destruction Court on trace completeness
#22 PDF Preflight	Trace metadata injection	Automated trace injection + court validation

---

Quick Load Commands

```
# Source trace pipeline
hermes -s moussaif-source-trace-qa,moussaif-agi-philosophy,moussaif-orchestrator,hermes-fattah-hyperloop-factory
    chat

# Trace audit
hermes -s moussaif-source-trace-qa,moussaif-hyperkb-graph-builder chat
```

Chapter Deliverables Checklist

- □ Kernel card (`trace-kernel.yaml`) with MCD statement
- □ Concept map (GraphRAG JSON) for trace domain
- □ All trace links epistemic-tagged with domain + source
- □ Destruction Court log (5 courts) for trace completeness
- □ Source trace chain for every published claim
- □ HyperSkill card for any new trace capability
- □ Reconstruction memory entry
- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry, database_registry, daily_generation_report, failure_registry, duplication_registry)
- □ Brand compliance: visual doctrine, voice, Instagram pillar alignment
- □ Full trace pipeline operational (ingest → clean → chunk → tag → link → log → monitor)
- □ Reconstruction queries validated (kernel history, chunk lineage, concept drift, version genealogy)
- □ Monitoring alerts configured (staleness, drift, orphans, conflicts)
- □ Schema migration tested (v1.0 → v1.1 trace schema)

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0183..0220 (Memory Write/Retrieval - trace persistence)
- CORE-0209..0234 (Reproducible Decision Trace - trace as evidence)
- CORE-0235..0260 (Cache Invalidation - trace freshness)
- MEM-001..007 (Memory Architecture - episodic, semantic, procedural, strategic, negative, benchmark)
- DSL-001..005 (DSL Grammar - trace as executable knowledge)
- CORE-0011 (Evidence Trail - every agentic action traced)

DSL Record Types Used in This Chapter

- **MEM** - Memory architecture (episodic, semantic, procedural, strategic, negative, benchmark)
- **CORE** - Subsystem mandates (memory-write, memory-retrieval, incident-response, self-improvement)
- **DSL** - Grammar rules (trace as executable knowledge)
- **EV** - Evaluation (trace completeness scoring, freshness scoring)
- **OP** - Operations (ingest, clean, chunk, tag, link, log, monitor, reconstruct)
- **CORE** - Tool mandates (validation, provenance, timestamping, redaction)

Chapter 14 Execution Command

```
hermes -s moussaif-source-trace-qa,moussaif-agi-philosophy,moussaif-orchestrator,hermes-fattah-hyperloop-factory
    chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - Every trace link resolves to verifiable artifact (file, API response, log, hash)
2. **Reductionism** - Full chain preserved: source → hash → cleaning ops → chunks → tags → concepts → MCDs → skills → usage locations
3. **False Novelty** - Prior art (Git, DVC, MLflow, DataLineage) referenced, delta documented
4. **Weak Op** - Agent answers provenance queries, finds all usages, reconstructs any version, detects drift
5. **Agentic Failure** - Agent monitors source hashes, alerts on drift, finds orphans, suggests repairs, predicts staleness

---

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

15. EBEYOND.AI.ERA BRAND SYSTEM - BEYOND AI ERA IDENTITY

Subtitle: Visual Doctrine, Voice Architecture, Campaign Angles, Cross-Platform Consistency @ebeyond.ai.era
Focus: Brand as Operating System: every asset regenerates from core MCD **Brand:** @ebeyond.ai.era (Instagram) | EBYOND.AI.ERA **Series:** Moussaif AGI-Oriented KB Technology **Chapter:** 15 of 15 **Vault Location:** `01_CORE_KB/07_KNOWLEDGE_ENGINEERING/moussaif_agi_ebook_kb/15_ebeyond_ai_era_brand_system_beyond_ai_era_identity.md`

Executive Summary

This chapter implements the **EBEYOND.AI.ERA Brand System** as a HyperSkill: a regenerating brand operating system where every visual, verbal, and strategic asset derives from a single governing MCD. The brand doesn't just "have guidelines" - it *is* an executable kernel that produces consistent, auditable, evolvable identity across all touchpoints.

> **Core Principle:** *Cost-minimizing, output-maximizing. Never waste premium tokens on simple work. Classify every task. Compress context. Draft → Critique → Finalize.*

The Governing Brand MCD

> **The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system.**

This MCD governs every visual, verbal, and strategic decision in the EBYOND.AI.ERA ecosystem.

Moussaif AGI Integration Map

AGI Component	Brand System Application
Supreme Architect (Nemotron-3-Ultra)	Brand kernel guardianship, strategic pivots, crisis response
Deep Reasoner (Nemotron-3-Super)	Brand architecture causal chains, audience segmentation logic
Coding Agent (Laguna-M-1)	Brand asset pipeline: generation, validation, deployment, monitoring
Fast Coder (Laguna-XS)	Quick asset variants, platform adaptations, emergency fixes
Research Synthesizer (Hunyuan-3B)	Competitive brand audit, audience research, cultural trend tracking
Critic Court (GPT-OSS-120B)	Brand drift detection, inconsistency flagging, guideline violation
Compression Agent (GPT-OSS-20B)	Guideline condensation, asset manifest summarization
Planner (Gemma-4-26B)	Campaign calendar, asset production schedule, cross-platform rollout
Executor (Qwen3-Coder)	Asset generation, platform publishing, analytics instrumentation
Refiner (North-Mini-Code)	Polish, consistency enforcement, micro-copy optimization
Multilingual (Gemma-4-26B)	Arabic/French/English brand voice, cultural adaptation
Visual Director (Nemotron-Nano-Omni)	Visual doctrine enforcement, prompt engineering, asset QA
Video Director (Nemotron-Nano-Omni)	Brand video system, motion guidelines, motion asset QA
SEO/Growth (Hunyuan-3B)	Brand keyword ownership, funnel brand consistency, attribution
Outreach (GPT-OSS-20B)	Partnership brand alignment, influencer brand briefs, PR voice
Data Extractor (Laguna-XS)	Brand asset performance extraction, consistency metrics
Memory Curator (Gemma-4-31B)	Brand evolution memory, kernel refinement, guideline versioning
Tool-Use Agent (Laguna-M-1)	Design tool orchestration (Figma, Canva, FAL, Recraft), asset APIs

AGI Component	Brand System Application
Low-Cost Draft (GPT-OSS-20B)	First-pass concepts, copy variants, layout explorations
Premium Final (Nemotron-3-Ultra)	Final brand synthesis, strategic repositioning, crisis resolution

8 HyperLoops Applied to Brand System

Loop 1: Kernel Extraction

Agent: MCD Extractor **Input:** Brand history + audience data + competitive landscape + founder vision **Output:** `brand-kernel.yaml` with governing MCD **Validation:** Kernel regenerates 80%+ of brand decisions **Brand MCD:** "*The governing difference between AI-era content that scales and content that stalls is whether the brand operates as a static identity or a regenerating operating system.*"

Loop 2: Evidence Ranking

Agent: Researcher **Input:** Brand performance data + audience feedback + competitive analysis + cultural signals **Output:** Epistemic tags on every brand claim **Validation:** Zero `ESTABLISHED` brand claims without measurable evidence

Loop 3: Concept Disambiguation

Agent: Conceptualizer **Input:** Term clusters (brand, identity, voice, tone, personality, archetype, positioning, equity, awareness, loyalty, advocacy, community, tribe, movement) **Output:** Concept map with IS_A, PART_OF, OVERLAPS, DISTINCT_FROM, CAUSES, ENABLES relations **Validation:** No circular definitions; necessary/sufficient conditions per concept

Loop 4: Problem Decomposition

Agent: Planner **Input:** "Maintain brand consistency across [platforms] while evolving for [campaign]" **Output:** Task DAG: Audit → Align → Generate → Validate → Deploy → Monitor → Evolve **Validation:** Agent executes each leaf without human intervention

Brand Task DAG:

```
BRAND KERNEL (MCD)
      │
      ├─→ VISUAL DOCTRINE
      │       ├─→ Color System → Token Validation
      │       ├─→ Typography → Token Validation
      │       ├─→ Logo System → Token Validation
      │       ├─→ Iconography → Token Validation
      │       ├─→ Photography Style → Prompt Validation
      │       ├─→ Motion Language → Prompt Validation
      │       └─→ Template System → Component Validation
      │
      ├─→ VOICE ARCHITECTURE
      │       ├─→ Tone Map → Context Validation
      │       ├─→ Vocabulary → Forbidden/Required Lists
      │       ├─→ Syntax Patterns → Template Validation
      │       ├─→ Multilingual Rules → Native Review
      │       └─→ Forbidden Terms → Automated Scanning
      │
      ├─→ CAMPAIGN ANGLES
      │       ├─→ Pillar Angles → Kernel Alignment Check
      │       ├─→ Seasonal Angles → Calendar Integration
      │       ├─→ Reactive Angles → Speed/Accuracy Gates
      │       └─→ Partnership Angles → Alignment Validation
      │
      ├─→ CROSS-PLATFORM CONSISTENCY
      │       ├─→ Platform Adaptation Rules → Token Mapping
      │       ├─→ Asset Generation → Template Enforcement
      │       ├─→ Publishing → Automated QA
      │       └─→ Analytics → Consistency Scoring
      │
      └─→ EVOLUTION & MEMORY
              ├─→ Kernel Refinement → Evidence Threshold
              ├─→ Guideline Versioning → Semantic Versioning
              ├─→ Drift Detection → Automated Alerts
              └─→ Reconstruction Memory → Genealogy Tracking
```

Loop 5: Destruction & Repair (5 Courts)

Agent: Critic (parallel courts) **Input:** Brand execution trace + live assets + audience feedback **Courts:**

1. **Hallucination** - Fake brand claims? Invented audience insights? Fabricated performance metrics?
2. **Reductionism** - Brand = logo? Voice = word list? Consistency = same filter?
3. **False Novelty** - "Living brand" = updated style guide? "AI brand" = generated assets without kernel?
4. **Weak Op** - Agent generates off-brand assets? Cannot validate new platform? Cannot detect drift?
5. **Agentic Failure** - Would agent catch cultural misalignment before publish? Detect silent drift?

Output: Pass/Fail per court + repair instructions **Validation:** All 5 pass = version tag; any fail = repair → re-test

Loop 6: Tool Construction

Agent: Builder **Input:** Validated brand patterns **Output:**

- `brand_token_validator.py` - Design token enforcement (colors, spacing, typography)
- `voice_linter.py` - Forbidden terms, required patterns, tone scoring
- `asset_generator.py` - Template-based asset generation with brand compliance
- `platform_adapter.py` - Platform-specific transformations with brand preservation
- `drift_detector.py` - Visual/verbal drift detection vs brand kernel
- `campaign_orchestrator.py` - Angle generation, asset scheduling, cross-platform sync
- `brand_analytics.py` - Consistency scoring, audience resonance, equity tracking

Validation: 10x speedup vs manual brand management

Loop 7: Agentic Execution

Agent: Executor + Critic loop **Input:** Task graph (Loop 4) + Tools (Loop 6) **Output:** Live brand assets across platforms + execution trace + metrics **Validation:** Output passes Destruction Court (Loop 5)

Loop 8: Update & Version

Agent: Memory Curator **Input:** Performance data, cultural shifts, competitive moves, kernel refinements **Output:** Versioned brand system + reconstruction memory entry **Validation:** Regression test on brand consistency metrics

---

5 Destruction Courts - Brand-Specific Checks

Court	Brand Failure Modes	Pass Criteria
Hallucination	Fake audience data, invented brand heritage, fabricated performance claims, hallucinated cultural relevance	Every brand claim traced to research, analytics, or documented decision
Reductionism	Brand = visual only, Voice = adjective list, Consistency = template reuse, Evolution = redesign	Kernel → Doctrine → System → Assets → Analytics chain intact
False Novelty	"Living brand" = updated PDF, "AI brand" = generated assets without kernel, "Community" = follower count	Kernel regenerates system; evolution = kernel refinement + memory
Weak Op	Agent generates off-brand colors, cannot validate new platform adaptation, misses drift	Agent generates compliant assets, validates adaptations, detects drift
Agentic Failure	Human catches cultural misalignment, human notices silent drift, human fixes inconsistency	Agent pre-validates cultural fit, monitors drift, auto-corrects or alerts

---

EBYOND.AI.ERA VISUAL DOCTRINE

Color System (Design Tokens)

```
color_tokens:
  # Primary - The Neural Trinity
  deep_space: "#0A0E27"        # Primary background, authority
  electric_cyan: "#00F5FF"     # Primary action, energy, innovation
  neural_gold: "#FFD700"       # Primary highlight, value, premium

  # Secondary - The Cognitive Spectrum
  synaptic_purple: "#8B5CF6"   # Secondary action, creativity
  dendritic_teal: "#14B8A6"    # Success, growth, data
  axonal_orange: "#F97316"     # Warning, attention, urgency
  myelin_white: "#F8FAFC"      # Primary text on dark, clean space
  glial_gray: "#64748B"        # Secondary text, muted elements

  # Semantic Aliases
  bg_primary: "deep_space"
  bg_secondary: "#111827"
  text_primary: "myelin_white"
  text_secondary: "glial_gray"
  accent_primary: "electric_cyan"
  accent_secondary: "neural_gold"
  accent_tertiary: "synaptic_purple"
  success: "dendritic_teal"
  warning: "axonal_orange"
  error: "#EF4444"

  # Gradients
  gradient_primary: "linear(135deg, deep_space, #1a1f3a)"
  gradient_energy: "linear(135deg, electric_cyan, synaptic_purple)"
  gradient_value: "linear(135deg, neural_gold, axonal_orange)"
  gradient_neural: "linear(135deg, deep_space, synaptic_purple, electric_cyan)"
```

Typography System

```
typography_tokens:
  font_families:
    heading: "Space Grotesk"          # Geometric, technical, ambitious
    mono: "JetBrains Mono"            # Code, technical, precise
    body: "Inter"                     # Readable, modern, neutral
    accent: "Syne"                    # Display, distinctive, energetic

  scale:
    xs: "0.75rem"    # 12px - Captions, meta
    sm: "0.875rem"   # 14px - Small body, labels
    base: "1rem"     # 16px - Body text
    lg: "1.125rem"   # 18px - Large body, intro
    xl: "1.25rem"    # 20px - Subheadings
    "2xl": "1.5rem"  # 24px - Section headers
    "3xl": "2rem"    # 32px - Page titles
    "4xl": "3rem"    # 48px - Hero headlines
    "5xl": "4rem"    # 64px - Brand statements

  weights:
    light: 300
    regular: 400
    medium: 500
    semibold: 600
    bold: 700
    extrabold: 800

  line_heights:
    tight: 1.1
    normal: 1.5
    relaxed: 1.75
```

Logo System

```
logo_system:
  primary:
    glyph: "EBYOND"  # Animated neural network motif
    wordmark: "EBYOND.AI.ERA"
    tagline: "Beyond AI Era"
    clear_space: "2x glyph height"
    min_width: "120px digital, 1in print"

  variants:
    horizontal: "Glyph + Wordmark + Tagline (single line)"
    stacked: "Glyph above Wordmark above Tagline"
    glyph_only: "Animated neural motif (favicon, app icon, social avatar)"
    wordmark_only: "EBYOND.AI.ERA (tight spaces)"
    monochrome: "Single color (deep_space or myelin_white)"

  animation:
    idle: "Slow pulse (4s cycle) - neural firing metaphor"
    hover: "Accelerated pulse (1s) + cyan glow"
    load: "Progressive build: nodes → connections → glow"
    transition: "Morph between states (300ms ease-out)"
```

Iconography System

```
iconography:
  style: "Outlined, 2px stroke, rounded caps, 24x24 grid"
  categories:
    neural: "Nodes, connections, pulses, layers"
    data: "Charts, flows, pipelines, storage"
    intelligence: "Brains, sparks, lenses, patterns"
    growth: "Curves, steps, compounds, loops"
    trust: "Shields, checkmarks, anchors, roots"
  states:
    default: "electric_cyan stroke, transparent fill"
    active: "electric_cyan fill"
    disabled: "glial_gray stroke, 40% opacity"
```

Photography & Visual Style

```
visual_style:
  photography:
    mood: "Clinical futurism meets organic warmth"
    lighting: "High contrast, cyan/gold rim lighting, deep shadows"
    subjects: "Diverse humans + technology, abstract data landscapes, neural metaphors"
    composition: "Rule of thirds, negative space for text, leading lines toward cyan"
    color_grading: "Cool shadows (deep_space), warm highlights (neural_gold), cyan accents"

  illustration:
    style: "Flat geometric, 2px stroke, neural/cybernetic metaphors"
    palette: "Restricted to color tokens"
    concepts: "Data as living organism, knowledge as network, AI as amplifier"

  motion:
    easing: "cubic-bezier(0.4, 0, 0.2, 1) - Material Design standard"
    durations:
      micro: "100ms - State changes"
      standard: "300ms - Transitions"
      macro: "500ms - Page loads, complex animations"
    principles: "Neural pulse rhythm, purposeful not decorative, respects prefers-reduced-motion"
```

EBYOND.AI.ERA VOICE ARCHITECTURE

Tone Map (Context-Dependent)

```
voice_tone_map:
  default:
    tone: "Precise, ambitious, technically grounded, philosophically rigorous"
    vocabulary: "MCD, HyperLoop, HyperSkill, Destruction Court, Kernel, Epistemic Tag, QMCD, Agentic, GraphRAG"
    sentence_structure: "Declarative, evidence-led, thesis-first"
    forbidden: "game-changing, revolutionary, unlock, secret, hack, masterclass (unless literal)"

  educational:
    tone: "Patient, scaffolding, example-rich, checkpoint-oriented"
    structure: "Concept → Example → Practice → Verification"

  launch:
    tone: "Confident, evidence-backed, urgent but not desperate"
    structure: "Promise → Proof → Path → Invitation"

  crisis:
    tone: "Transparent, accountable, action-oriented, humble"
    structure: "What happened → Why → What we're doing → Prevention"

  community:
    tone: "Inclusive, technically accessible, achievement-celebrating"
    structure: "Recognition → Shared learning → Collective challenge"

  technical:
    tone: "Rigorous, reproducible, specification-first"
    structure: "Specification → Implementation → Verification → Limitations"
```

Vocabulary Control

```
vocabulary_control:
  required_terms:
    - "MCD (Meta-Complexity Difference)"
    - "HyperLoop"
    - "HyperSkill"
    - "Destruction Court"
    - "Kernel"
    - "Epistemic Tag"
    - "QMCD (Quantum MCD)"
    - "Agentic"
    - "GraphRAG"
    - "Fattah"
    - "Source Trace"
    - "Reconstruction Memory"

  forbidden_terms:
    - "game-changing"
    - "revolutionary"
    - "unlock"
    - "secret"
    - "hack"
    - "masterclass"  # unless literally a masterclass
    - "breakthrough"  # unless peer-reviewed
    - "paradigm shift"  # unless historical analysis
    - "disruptive"  # unless Clayton Christensen context
    - "next-level"
    - "skyrocket"
    - "explosive growth"

  preferred_alternatives:
    "revolutionary" → "kernel-regenerating"
    "unlock" → "operationalize"
    "secret" → "non-obvious mechanism"
    "hack" → "tool construction"
    "masterclass" → "HyperSkill"
    "breakthrough" → "destruction court passage"
    "paradigm shift" → "MCD extraction"
    "disruptive" → "agentic replacement"
```

Syntax Patterns

```
syntax_patterns:
  claim_structure: "[Thesis] → [Evidence] → [Boundary] → [Application]"
  example: "Flat conceptual covers increase CTR 23% (GOAT KB #7 + internal A/B) for KDP nonfiction; may not
      apply to fiction."

  disclaimer_pattern: "[Claim] — [Confidence Tag] — [Source] — [Boundary]"
  example: "Agentic workflows reduce production time 10x — SUPPORTED — Internal benchmark 2024 — Measured on
      faceless content pipeline only."

  cta_pattern: "[Value] + [Urgency/Scarcity] + [Low-Friction Action]"
  example: "Deploy your first HyperLoop today — Free tier includes 40 loops/month — Start in 60 seconds."
```

CAMPAIGN ANGLES (Content Pillars)

Pillar 1: Architecture Sunday

Frequency: Weekly (Sunday) **Format:** System diagrams, HyperLoop visualizations, kernel extractions **Angle:** "See the invisible architecture behind compounding results" **Kernel Alignment:** Visualizes the brand MCD - architecture over aesthetics

Pillar 2: Court Monday

Frequency: Weekly (Monday) **Format:** Destruction Court case studies - what failed, why, repair **Angle:** "Quality is not a gate, it's the architecture" **Kernel Alignment:** Demonstrates the 5-court immune system in action

Pillar 3: Kernel Wednesday

Frequency: Weekly (Wednesday) **Format:** MCD extractions from bestselling books, viral campaigns, breakthrough products **Angle:** "The governing difference you missed" **Kernel Alignment:** Core methodology demonstration

Pillar 4: Tool Friday

Frequency: Weekly (Friday) **Format:** Automation scripts, preflight checks, upload pipelines, validation tools **Angle:** "The tool that caught the error you didn't know existed" **Kernel Alignment:** Tool Construction (Loop 6) as brand expression

Pillar 5: Launch Saturday

Frequency: Weekly (Saturday) **Format:** Real-time KDP/Gumroad launches with metrics, funnels, affiliate activation **Angle:** "Watch a knowledge product go from kernel to live revenue" **Kernel Alignment:** Deployment Translation (Loop 12) + full loop visibility

---

CROSS-PLATFORM CONSISTENCY SYSTEM

Platform Adaptation Rules

```
platform_adaptation:
  instagram:
    handle: "@ebeyond.ai.era"
    format: "1080x1080 (feed), 1080x1920 (stories/reels)"
    voice: "Visual-first, caption as micro-essay, hashtags as topic map"
    adaptation: "Visual doctrine primary, voice supports"
    frequency: "Daily (stories), 3x/week (feed), 2x/week (reels)"

  linkedin:
    handle: "EBYOND.AI.ERA"
    format: "1200x628 (articles), 1080x1080 (carousels), 1920x1080 (video)"
    voice: "Professional, thesis-driven, evidence-rich"
    adaptation: "Voice primary, visual supports"
    frequency: "2x/week (posts), 1x/week (articles)"

  x_twitter:
    handle: "@ebeyond_ai_era"
    format: "16:9 (images), 16:9 (video), threads"
    voice: "Concise, hook-first, thread-structured"
    adaptation: "Voice primary, visual as thread illustration"
    frequency: "Daily (tweets), 3x/week (threads)"

  youtube:
    handle: "@EBYONDAIERA"
    format: "1920x1080 (long), 1080x1920 (shorts)"
    voice: "Educational, demonstration-heavy, chapter-marked"
    adaptation: "Visual + voice integrated, chapters as MCD steps"
    frequency: "1x/week (long), 3x/week (shorts)"

  tiktok:
    handle: "@ebeyond.ai.era"
    format: "1080x1920 (vertical video)"
    voice: "Hook-first, pattern-interrupt, native format"
    adaptation: "Visual + voice fused, trending audio with brand thesis"
    frequency: "Daily"

  newsletter:
    handle: "beyond@ebeyond.ai.era"
    format: "HTML + plain text, single thesis per issue"
    voice: "Essayistic, personal, technically precise"
    adaptation: "Voice primary, visual as header/section breaks"
    frequency: "Weekly (Sunday - Architecture Sunday)"
```

Brand Token Mapping (Platform → Token)

```
token_mapping:
  instagram:
    primary_bg: "deep_space"
    accent: "electric_cyan"
    text: "myelin_white"
    font_heading: "Space Grotesk"
    font_body: "Inter"

  linkedin:
    primary_bg: "white"
    accent: "deep_space"
    text: "deep_space"
    font_heading: "Space Grotesk"
    font_body: "Inter"

  x_twitter:
    primary_bg: "deep_space"
    accent: "electric_cyan"
    text: "myelin_white"
    font_heading: "Space Grotesk"
    font_body: "Inter"

  youtube:
    primary_bg: "deep_space"
    accent: "electric_cyan"
    text: "myelin_white"
    font_heading: "Space Grotesk"
    font_body: "Inter"
    thumbnail_accent: "neural_gold"

  tiktok:
    primary_bg: "deep_space"
    accent: "electric_cyan"
    text: "myelin_white"
    font_heading: "Syne"
    font_body: "Inter"
    caption_style: "native_tiktok_font"
```

BRAND EVOLUTION & MEMORY

Kernel Refinement Protocol

```
kernel_refinement:
  trigger_conditions:
    - "Consistency score drops below 0.85 for 2 consecutive weeks"
    - "Audience resonance shifts >15% on core attributes"
    - "Competitive landscape introduces new category convention"
    - "Platform introduces new format requiring brand decision"
    - "Founder/strategic pivot requires kernel realignment"

  process:
    1. "Evidence Collection (7 days): Performance + Feedback + Competitive + Cultural"
    2. "Kernel Hypothesis Generation (3 variants max)"
    3. "QMCD Superposition (8 models) on each variant"
    4. "Destruction Court on each variant"
    5. "Selection: Highest evidence score + court pass"
    6. "Migration: Semantic version bump + migration guide + reconstruction entry"

  versioning:
    major: "Kernel MCD changed (brand essence shift)"
    minor: "Doctrine/System/Assets expanded (new platform, new pillar)"
    patch: "Token tweak, guideline clarification, bug fix"
```

Reconstruction Memory (Brand Genealogy)

```
## Brand Reconstruction Memory: 2025-07-10

### Kernel Evolution
- v1.0: "Static identity with guidelines"
- v2.0: "Regenerating operating system" (MCD extracted)
- v2.1: "Platform-specific adaptation rules added"
- v2.2: "Destruction Court integrated as quality gate"

### Visual Doctrine Changes
- v1.0: Generic tech blue/purple
- v2.0: Neural Trinity (Deep Space, Electric Cyan, Neural Gold)
- v2.1: Cognitive Spectrum added (6 semantic colors)
- v2.2: Gradient tokens + motion tokens added

### Voice Architecture Changes
- v1.0: Generic "expert" tone
- v2.0: Vocabulary control + forbidden terms + syntax patterns
- v2.1: Context-specific tone map (5 contexts)
- v2.2: Claim structure + disclaimer pattern + CTA pattern formalized

### Failed Experiments (Logged to failure_registry)
- "Community-driven visual evolution" → Drift >40% in 2 weeks
- "AI-generated brand variants" → 73% failed Destruction Court
- "Trend-reactive visual style" → Inconsistency score 0.31

### Next Refinement Priorities
1. Arabic/French voice architecture (multilingual expansion)
2. Video motion language formalization (Layer 7 of Visual Stack)
3. Brand equity measurement framework (beyond consistency)
```

QUICK LOAD COMMANDS

```
# Brand system orchestration
hermes -s
    moussaif-brand-voice-extractor,moussaif-cover-design-system,moussaif-agi-philosophy,moussaif-orchestrator
    chat

# Brand asset generation
hermes -s faceless-thumbnail-designer-flux,faceless-channel-art-designer,faceless-logo-branding-recraft,faceless
    -recraft-vector-illustration,moussaif-brand-voice-extractor chat

# Brand audit
hermes -s moussaif-agi-philosophy,moussaif-orchestrator,moussaif-brand-voice-extractor chat
```

Chapter Deliverables Checklist

- □ Kernel card (`brand-kernel.yaml`) with MCD statement
- □ Visual doctrine tokens (colors, typography, spacing, motion)
- □ Logo system (variants, animation, clear space, min sizes)
- □ Voice architecture (tone map, vocabulary control, syntax patterns)
- □ Campaign angles (5 pillars with frequency, format, kernel alignment)
- □ Platform adaptation rules (6 platforms, token mapping)
- □ Brand token system (design tokens + platform mapping)
- □ Asset templates (carousels, threads, thumbnails, emails, funnels)
- □ Validation tools (token validator, voice linter, drift detector)
- □ Destruction Court log (5 courts) for brand compliance
- □ Source trace chain for every brand claim
- □ HyperSkill card for any new brand capability
- □ Reconstruction memory entry

- □ Version tag (semver) + changelog
- □ Registry updates (hyperloop_registry, hyperskill_registry, source_registry)
- □ Brand compliance: all assets pass token validator + voice linter
- □ Drift detection alerts configured
- □ Consistency scoring dashboard live

DSL INTEGRATION: CHAPTER-SPECIFIC CORE DIRECTIVES

Relevant CORE Directives for This Chapter

- CORE-0001..0003 (Brand as Kernel - never assume static, refresh before routing, probe before execution)
- CORE-0019..0025 (Paid-First Logic - brand equity as prepaid capacity)
- CORE-0026..0032 (Image Orchestration - visual thesis before generation)
- CORE-0040..0045 (Cost Policy - cheap drafting + premium verification)
- CORE-0046..0051 (Quality Policy - multi-dimensional brand quality)
- CORE-0052..0056 (AGI Fattah - brand as agency opener)
- CORE-0112..0116 (AGI Fattah Orientation - brand intelligence measurement)
- CORE-0117..0125 (Visual Intelligence Stack - 9 layers as brand QA)
- VS-001..009 (Visual Intelligence Stack - 9 layers)
- HS-001..004 (Hyperskill Invariant - brand as versioned capability)

DSL Record Types Used in This Chapter

- **VS** - Visual Intelligence Stack (9 layers as brand QA)
- **HS** - Hyperskill definitions (brand system as capability)
- **CORE** - Subsystem mandates (provider-discovery, credit-accounting, quality-scoring)
- **EV** - Evaluation (brand consistency scoring, resonance scoring)
- **OP** - Operations (asset generation, platform adaptation, publishing, monitoring)
- **RT** - Routing (model selection for visual vs verbal tasks)
- **MEM** - Memory Architecture (brand evolution memory, kernel genealogy)
- **ST** - State Machine (brand system states: consistent, drifting, crisis, evolving)

Chapter 15 Execution Command

```
hermes -s
    moussaif-brand-voice-extractor,moussaif-cover-design-system,moussaif-agi-philosophy,moussaif-orchestrator
    chat
```

Destruction Court Gates for This Chapter

1. **Hallucination** - Every brand claim traced to research, analytics, or documented decision
2. **Reductionism** - Kernel → Doctrine → System → Assets → Analytics chain intact
3. **False Novelty** - Prior art referenced, delta documented, kernel genuinely regenerates
4. **Weak Op** - Agent generates compliant assets, validates adaptations, detects drift
5. **Agentic Failure** - Agent monitors cultural fit, drift, inconsistency; auto-corrects or alerts

This chapter is a living HyperSkill. It updates via the HyperLoop Factory. Current version: 1.0.0 | Registry: hyperskill_registry.jsonl

TECHNICAL APPENDIX: COMPLETE DSL SPECIFICATIONS & CORE DIRECTIVES

From: ZOUHAIR MOUSSAIF HERMES AGFATAHI / AGI FATTAH DUAL-ORCHESTRATION OS (100K Lines)

1. NON-NEGOTIABLE OPERATING CONTRACT (CORE-0001..0015)

```
CORE-0001 | subsystem=provider-discovery | directive=Never assume a model still exists, remains free, or retains
    the same capabilities | evidence=required | failure=typed-and-logged
CORE-0002 | subsystem=provider-discovery | directive=Refresh OpenRouter text, image, video, endpoint, pricing,
    and credit metadata before routing | evidence=required | failure=typed-and-logged
CORE-0003 | subsystem=provider-discovery | directive=Probe NVIDIA model availability and quota before execution;
    a catalog label is not an unlimited-use promise | evidence=required | failure=typed-and-logged
CORE-0004 | subsystem=credit-accounting | directive=When OpenRouter paid credits remain above reserve, spend
    them on the best fitting paid model first | evidence=required | failure=typed-and-logged
CORE-0005 | subsystem=overlap-detection | directive=If an equivalent model is available through both OpenRouter
    and NVIDIA, prefer OpenRouter while paid-credit mode is active | evidence=required |
    failure=typed-and-logged
CORE-0006 | subsystem=credit-accounting | directive=When OpenRouter credits reach the reserve or zero, prefer
    NVIDIA free/development endpoints for overlapping models | evidence=required | failure=typed-and-logged
CORE-0007 | subsystem=capability-filtering | directive=For models not duplicated across providers, choose the
    highest-scoring NVIDIA candidate that satisfies the task contract | evidence=required |
    failure=typed-and-logged
CORE-0008 | subsystem=provider-discovery | directive=If NVIDIA is unavailable, use OpenRouter free models
    selected by explicit capability filters; use openrouter/free only as last-resort stochastic routing |
    evidence=required | failure=typed-and-logged
CORE-0009 | subsystem=image-direction | directive=Image and video jobs must be capability-filtered before
    submission; never infer support from a model name | evidence=required | failure=typed-and-logged
CORE-0010 | subsystem=cost-estimation | directive=Every expensive media request requires a preview, cost
    estimate, constraint check, and explicit execution gate | evidence=required | failure=typed-and-logged
CORE-0011 | subsystem=memory-write | directive=Every agentic action must leave an evidence trail: intent, route,
    model, cost, latency, tools, outputs, tests, and repair history | evidence=required |
    failure=typed-and-logged
CORE-0012 | subsystem=loop-termination | directive=No hyperloop may persist indefinitely without a stop
    condition, budget ceiling, iteration ceiling, or external interrupt | evidence=required |
    failure=typed-and-logged
CORE-0013 | subsystem=self-improvement | directive=Self-improvement may modify skills, prompts, scoring weights,
    and cached benchmarks, but not security boundaries or secret-handling rules | evidence=required |
    failure=typed-and-logged
CORE-0014 | subsystem=memory-write | directive=Generated knowledge is provisional until verified by tests,
    sources, or independent model review | evidence=required | failure=typed-and-logged
CORE-0015 | subsystem=memory-write | directive=The system distinguishes observed facts, provider metadata, local
    benchmarks, inferences, hypotheses, and creative proposals | evidence=required | failure=typed-and-logged
```

2. PROVIDER ENDPOINTS & DISCOVERY CONTRACT

```
RT-001 | provider=openrouter | base_url=https://openrouter.ai/api/v1 | models_endpoint=GET /models |
    image_models_endpoint=GET /images/models | video_models_endpoint=GET /videos/models | credits_endpoint=GET
    /credits | chat_endpoint=POST /chat/completions | video_create_endpoint=POST /videos
RT-002 | provider=nvidia | base_url=https://integrate.api.nvidia.com/v1 | models_endpoint=GET /models |
    chat_endpoint=POST /chat/completions | image_self_hosted=POST /v1/images/generations |
    cosmos_self_hosted=POST /v1/infer
RT-003 | env=OPENROUTER_API_KEY | env=OPENROUTER_MANAGEMENT_KEY | env=NVIDIA_API_KEY |
    env=HERMES_OS_ROOT=./hermes-moussaif-os | env=HERMES_MODEL_CACHE_TTL_SECONDS=1800 |
    env=HERMES_CREDIT_CACHE_TTL_SECONDS=60
RT-004 | env=HERMES_OPENROUTER_RESERVE_USD=0.25 | env=HERMES_MAX_JOB_USD=5.00 | env=HERMES_MAX_DAILY_USD=20.00 |
    env=HERMES_MAX_MEDIA_JOB_USD=3.00 | env=HERMES_DEFAULT_TIMEOUT_SECONDS=180
RT-005 | env=HERMES_MAX_LOOP_ITERATIONS=12 | env=HERMES_REQUIRE_MEDIA_PREVIEW=true |
    env=HERMES_ALLOW_OPENROUTER_FREE_ROUTER=true | env=HERMES_ALLOW_NVIDIA_DEVELOPMENT_ENDPOINTS=true
```

3. SECRET-HANDLING RULES (CORE-0001..0015)

```
CORE-0013 | subsystem=secret-redaction | directive=Secrets are read from environment variables or an OS secret
    store, never embedded in prompts, logs, filenames, or generated artifacts | evidence=required |
    failure=typed-and-logged
CORE-0014 | subsystem=secret-redaction | directive=Logs may record provider aliases and key fingerprints, but
    never raw API keys | evidence=required | failure=typed-and-logged
CORE-0015 | subsystem=secret-redaction | directive=Tool outputs are scanned for accidental secret disclosure
    before persistence | evidence=required | failure=typed-and-logged
CORE-0016 | subsystem=secret-redaction | directive=Any detected credential is redacted, the event is logged, and
    key rotation is recommended | evidence=required | failure=typed-and-logged
CORE-0017 | subsystem=tool-security | directive=A model cannot request broader filesystem, shell, network,
    email, browser, or repository privileges by prompt alone | evidence=required | failure=typed-and-logged
CORE-0018 | subsystem=tool-security | directive=Privileged actions require a policy gate independent from the
    model producing the request | evidence=required | failure=typed-and-logged
```

4. CANONICAL ROUTING STATES

```
EV-001 | state=OR_PAID_ACTIVE | condition=remaining_credit > reserve AND paid_execution_allowed
EV-002 | state=OR_PAID_LOW | condition=0 < remaining_credit <= reserve | directive=only critical jobs may
    consume the reserve
EV-003 | state=OR_ZERO | condition=remaining_credit <= 0 OR credit_endpoint_confirms_exhaustion
EV-004 | state=OR_UNKNOWN | condition=credit state unavailable | directive=use conservative per-job caps AND
    avoid premium media
EV-005 | state=NV_READY | condition=NVIDIA endpoint passes authentication, model discovery/probe, and quota
    checks
EV-006 | state=NV_LIMITED | condition=NVIDIA works BUT rate, token, concurrency, or modality limits are detected
EV-007 | state=NV_UNAVAILABLE | condition=authentication, catalog, endpoint, quota, or health checks fail
EV-008 | state=SURVIVAL_FREE | condition=OpenRouter paid unavailable AND NVIDIA cannot satisfy contract
```

5. PAID-FIRST & OVERLAP-FIRST INTERPRETATION

```
CORE-0019 | subsystem=overlap-detection | directive='First wins already paid in OpenRouter' means funded
    OpenRouter credits are treated as sunk prepaid capacity | evidence=required | failure=typed-and-logged
CORE-0020 | subsystem=overlap-detection | directive=The OS therefore consumes OpenRouter paid capacity before
    shifting equivalent work to NVIDIA development endpoints | evidence=required | failure=typed-and-logged
CORE-0021 | subsystem=overlap-detection | directive=Duplicate model availability is detected by canonical
    family, provider, architecture, version, modalities, and benchmark signature | evidence=required |
    failure=typed-and-logged
CORE-0022 | subsystem=overlap-detection | directive=Exact slug equality is useful but insufficient; aliases and
    dated model revisions are normalized | evidence=required | failure=typed-and-logged
CORE-0023 | subsystem=overlap-detection | directive=An overlap record has confidence: exact, strong-family,
    weak-family, or none | evidence=required | failure=typed-and-logged
CORE-0024 | subsystem=overlap-detection | directive=Only exact or strong-family overlap activates automatic
    OpenRouter-first duplicate-spend avoidance | evidence=required | failure=typed-and-logged
CORE-0025 | subsystem=overlap-detection | directive=Weak-family overlap requires benchmark evidence before
    substitution | evidence=required | failure=typed-and-logged
```

6. SEED REGISTRY (Bootstrap - Refresh at Boot)

```
RT-006 | OR_PAID_REASONING_1=openai/gpt-5.6-sol-pro
RT-007 | OR_PAID_REASONING_2=openai/gpt-5.6-sol
RT-008 | OR_PAID_BALANCED_1=openai/gpt-5.6-terra
RT-009 | OR_PAID_FAST_1=openai/gpt-5.6-luna
RT-010 | OR_PAID_AGENTIC_1=anthropic/claude-sonnet-5
RT-011 | OR_PAID_ORCHESTRATOR_1=sakana/fugu-ultra
RT-012 | OR_PAID_GENERAL_1=x-ai/grok-4.5
RT-013 | OR_IMAGE_PREMIUM_1=google/gemini-3-pro-image
RT-014 | OR_IMAGE_BALANCED_1=google/gemini-3.1-flash-image
RT-015 | OR_IMAGE_FAST_1=google/gemini-3.1-flash-lite-image
RT-016 | OR_IMAGE_DISCOVERY_SEED=bytedance-seed/seedream-4.5
RT-017 | OR_VIDEO_SEED_01=kwaivgi/kling-v3.0-pro
RT-018 | OR_VIDEO_SEED_02=kwaivgi/kling-v3.0-std
RT-019 | OR_VIDEO_SEED_03=google/veo-3.1-fast
RT-020 | OR_VIDEO_SEED_04=google/veo-3.1-lite
RT-021 | OR_VIDEO_SEED_05=kwaivgi/kling-video-o1
RT-022 | OR_VIDEO_SEED_06=minimax/hailuo-2.3
RT-023 | OR_VIDEO_SEED_07=bytedance/seedance-2.0
RT-024 | OR_VIDEO_SEED_08=bytedance/seedance-2.0-fast
RT-025 | OR_VIDEO_SEED_09=alibaba/wan-2.7
RT-026 | OR_VIDEO_SEED_10=alibaba/wan-2.6
RT-027 | OR_VIDEO_SEED_11=bytedance/seedance-1-5-pro
RT-028 | OR_VIDEO_SEED_12=openai/sora-2-pro
RT-029 | OR_VIDEO_SEED_13=google/veo-3.1
RT-030 | OR_FREE_ROUTER=openrouter/free
RT-031 | OR_FREE_REASONING_SEED=tencent/hy3:free
RT-032 | OR_FREE_CODE_SEED=poolside/laguna-xs-2.1:free
RT-033 | OR_FREE_CODE_ALT=cohere/north-mini-code:free
RT-034 | OR_FREE_NVIDIA_ULTRA=nvidia/nemotron-3-ultra-550b-a55b:free
RT-035 | OR_FREE_NVIDIA_NANO=nvidia/nemotron-3-nano-30b-a3b:free
RT-036 | OR_FREE_VISION=nvidia/nemotron-nano-12b-v2-vl:free
RT-037 | OR_FREE_LIGHT_REASONING=nvidia/nemotron-nano-9b-v2:free
RT-038 | NV_REASONING_1=nvidia/nemotron-3-ultra-550b-a55b
RT-039 | NV_REASONING_2=deepseek-ai/deepseek-v4-pro
RT-040 | NV_FAST_REASONING_1=deepseek-ai/deepseek-v4-flash
RT-041 | NV_MULTI_AGENT_1=nvidia/nemotron-3-super-120b-a12b
RT-042 | NV_CODE_1=qwen/qwen3-coder-480b-a35b-instruct
RT-043 | NV_REASONING_ALT_1=moonshotai/kimi-k2-thinking
RT-044 | NV_EFFICIENT_1=nvidia/nemotron-3-nano-30b-a3b
RT-045 | NV_VISION_1=nvidia/nemotron-nano-12b-v2-vl
RT-046 | NV_IMAGE_TEXT_1=qwen/qwen-image
RT-047 | NV_IMAGE_ART_1=black-forest-labs/flux_1-dev
RT-048 | NV_VIDEO_PHYSICAL_1=nvidia/cosmos3-nano
RT-049 | NV_VIDEO_SELF_HOSTED_1=Cosmos3-Generator
RT-050 | NV_VIDEO_SELF_HOSTED_2=Cosmos-Predict2.5-2B
```

Seed Registry Warning: Every seed is a bootstrap hint, not permanent truth. Expiring free promotions removed automatically when expiration_date reached. Zero price in metadata means free at that endpoint at that moment; does not imply unlimited quota. Model IDs discovered live override this seed list. If a seed disappears, capability matching selects replacement without changing skill contract.

7. MODEL SCORING FORMULA

```
EV-001 | formula=score = 0.24*quality + 0.16*agentic + 0.14*tool_use + 0.12*modality_fit + 0.10*reliability +
    0.08*context_fit + 0.06*latency + 0.05*cost_efficiency + 0.03*privacy_fit + 0.02*provider_health
EV-002 | normalization=[0,100] | hard_constraints_applied_before_scoring |
    failed_hard_constraint_cannot_be_compensated_by_high_score
EV-003 | local_benchmark_evidence_overrides_marketing_descriptions
EV-004 | recent_successful_executions_receive_decaying_reliability_bonus
EV-005 | repeated_correlated_failures_receive_provider_model_circuit_breaker_penalty
```

8. TASK CLASSIFICATION SCHEMA

```
EV-010 | task_family=reasoning|coding|research|writing|analysis|planning|automation|vision|image_generation|imag
    e_editing|video_generation|video_editing|audio|embedding|reranking|safety
EV-011 | complexity=1..10
EV-012 | autonomy=answer|assist|execute|monitor
EV-013 | modality_in=text|image|video|audio|file|mixed
EV-014 | modality_out=text|json|code|image|video|audio|artifact|mixed
EV-015 | latency_class=instant|interactive|batch|offline
EV-016 | budget_class=zero|economy|balanced|premium
EV-017 | verification_class=none|single_check|dual_check|test_suite|human_gate
EV-018 | privacy_class=public|internal|confidential|restricted
EV-019 | creativity_class=literal|controlled|exploratory|radical
```

9. CORE ROUTING PSEUDOCODE

```
def route(job, state, registries):
    candidates = capability_filter(job, registries)
    candidates = policy_filter(job, candidates)
    if state.openrouter == 'OR_PAID_ACTIVE':
        paid = paid_openrouter(candidates)
        if paid: return best_score(paid)
    if state.nvidia in {'NV_READY', 'NV_LIMITED'}:
        overlap = nvidia_overlap(candidates)
        if overlap: return best_score(overlap)
        native = nvidia_nonoverlap(candidates)
        if native: return best_score(native)
    free = explicit_openrouter_free(candidates)
    if free: return best_score(free)
    if state.allow_free_router: return 'openrouter/free'
    raise NoRouteSatisfiesContract(job.id)
```

10. IMAGE ORCHESTRATION CONTRACT

```
CORE-0026 | subsystem=image-direction | directive=Conceptual design uses a reasoning model to produce a visual
    thesis before image generation | evidence=required | failure=typed-and-logged
CORE-0027 | subsystem=image-direction | directive=The visual thesis contains message, composition, hierarchy,
    symbolism, palette logic, typography constraints, negative space, and audience effect | evidence=required |
    failure=typed-and-logged
CORE-0028 | subsystem=capability-filtering | directive=Image generation candidates are fetched from
    /images/models and filtered by text-to-image, image-to-image, resolution, aspect ratio, reference-image,
    seed, streaming, and price support | evidence=required | failure=typed-and-logged
CORE-0029 | subsystem=image-editing | directive=Premium conceptual art prefers the highest-scoring
    image-edit-capable model when iterative refinement is required | evidence=required |
    failure=typed-and-logged
CORE-0030 | subsystem=image-direction | directive=Text-rich posters prefer models with demonstrated
    text-rendering capability | evidence=required | failure=typed-and-logged
CORE-0031 | subsystem=image-direction | directive=Aesthetic judgment uses at least two independent critiques:
    semantic/message fidelity and visual/compositional quality | evidence=required | failure=typed-and-logged
CORE-0032 | subsystem=image-direction | directive=Iteration stops when quality thresholds are met, budget is
    exhausted, or improvements plateau | evidence=required | failure=typed-and-logged
```

11. VIDEO ORCHESTRATION CONTRACT

```
CORE-0033 | subsystem=video-selection | directive=Video models are fetched from /videos/models and filtered by
    duration, aspect ratio, resolution, audio generation, first-frame, last-frame, seed, and provider controls |
    evidence=required | failure=typed-and-logged
CORE-0034 | subsystem=video-polling | directive=Video editing is implemented only when the selected endpoint
    explicitly supports reference frames, source video, controls, or editing parameters | evidence=required |
    failure=typed-and-logged
CORE-0035 | subsystem=video-polling | directive=Otherwise the system labels the operation as regeneration,
    continuation, image-to-video, or shot replacement rather than claiming true editing | evidence=required |
    failure=typed-and-logged
CORE-0036 | subsystem=video-polling | directive=Every video request is previewed before POST /videos because
    submission can consume paid credits | evidence=required | failure=typed-and-logged
CORE-0037 | subsystem=video-polling | directive=The system polls job status with bounded exponential backoff and
    stores the generation ID without exposing secrets | evidence=required | failure=typed-and-logged
CORE-0038 | subsystem=video-selection | directive=NVIDIA Cosmos is preferred for physics-aware, world-model,
    industrial, robotics, simulation, or physically constrained video | evidence=required |
    failure=typed-and-logged
CORE-0039 | subsystem=video-selection | directive=Cinematic advertising and social video use OpenRouter's best
    capability-matched video model unless cost policy selects a cheaper tier | evidence=required |
    failure=typed-and-logged
```

12. HYPERLOOP INVARIANT

```
HL-001 | structure=trigger -> perceive -> retrieve -> propose -> route -> act -> verify -> critique -> repair ->
    compress -> persist -> stop-or-repeat
HL-002 | directive=Each stage emits a typed artifact
HL-003 | directive=Each artifact has provenance, confidence, and expiry metadata
HL-004 | directive=The critic cannot silently rewrite the original objective
HL-005 | directive=The repair stage must cite the failed test or quality dimension it addresses
HL-006 | directive=The memory stage stores reusable abstractions, not unfiltered transcripts
```

13. HYPERSKILL INVARIANT

```
HS-001 | definition=A hyperskill is a reusable, versioned capability contract with inputs, outputs, tools, model
    requirements, verification, budget, and failure semantics
HS-002 | directive=Skills are composable but cannot silently escalate privileges
HS-003 | directive=A skill is promoted only after passing representative tests and adversarial tests
HS-003 | directive=Skill updates are immutable versions; aliases point to the latest approved version
HS-004 | directive=Skill confidence decays when provider behavior or repository context changes
```

14. MEMORY ARCHITECTURE

```
MEM-001 | type=episodic | content=execution traces and outcomes
MEM-002 | type=semantic | content=verified facts and normalized concepts
MEM-003 | type=procedural | content=skills, tools, prompts, tests, routing recipes
MEM-004 | type=strategic | content=goals, constraints, preferences, project roadmaps
MEM-005 | type=negative | content=known failures, anti-patterns, rejected routes, unsafe actions
MEM-006 | type=benchmark | content=model/task scores with timestamp, sample size, and confidence interval
MEM-007 | directive=Memory writes require deduplication, compression, provenance, and expiry
```

15. LINE-ORIENTED DSL GRAMMAR

```
DSL-001 | grammar=TYPE-ID | key=value | key=value | contract=imperative sentence | verify=check |
    fallback=ordered route | stop=condition
DSL-002 | TYPE_values=CORE, HS, HL, RT, PC, EV, OP, IT
DSL-003 | directive=IDs are stable within this file version
DSL-004 | directive=Values avoid unescaped vertical bars
DSL-004 | directive=Parsers should preserve unknown keys for forward compatibility
DSL-005 | directive=A line is executable knowledge only after schema validation
```

16. BOOTSTRAP FILESYSTEM

```
FS-001 | hermes-moussaif-os/config/providers.yaml
FS-002 | hermes-moussaif-os/config/budgets.yaml
FS-003 | hermes-moussaif-os/config/policies.yaml
FS-004 | hermes-moussaif-os/config/model-aliases.yaml
FS-005 | hermes-moussaif-os/registries/openrouter-text.json
FS-006 | hermes-moussaif-os/registries/openrouter-image.json
FS-007 | hermes-moussaif-os/registries/openrouter-video.json
FS-008 | hermes-moussaif-os/registries/nvidia-models.json
FS-009 | hermes-moussaif-os/registries/overlap-map.json
FS-010 | hermes-moussaif-os/skills/approved/
FS-011 | hermes-moussaif-os/skills/candidate/
FS-012 | hermes-moussaif-os/skills/retired/
FS-013 | hermes-moussaif-os/loops/active/
FS-014 | hermes-moussaif-os/loops/paused/
FS-015 | hermes-moussaif-os/loops/completed/
FS-016 | hermes-moussaif-os/prompts/capsules/
FS-017 | hermes-moussaif-os/prompts/critics/
FS-018 | hermes-moussaif-os/memory/episodic/
FS-019 | hermes-moussaif-os/memory/semantic/
FS-020 | hermes-moussaif-os/memory/procedural/
FS-021 | hermes-moussaif-os/memory/strategic/
FS-022 | hermes-moussaif-os/memory/negative/
FS-023 | hermes-moussaif-os/memory/benchmarks/
FS-024 | hermes-moussaif-os/logs/routing.jsonl
FS-025 | hermes-moussaif-os/logs/costs.jsonl
FS-026 | hermes-moussaif-os/logs/failures.jsonl
FS-027 | hermes-moussaif-os/logs/audits.jsonl
FS-028 | hermes-moussaif-os/tests/unit/
FS-029 | hermes-moussaif-os/tests/integration/
FS-030 | hermes-moussaif-os/tests/adversarial/
FS-029 | hermes-moussaif-os/artifacts/images/
FS-030 | hermes-moussaif-os/artifacts/videos/
FS-031 | hermes-moussaif-os/artifacts/documents/
FS-032 | hermes-moussaif-os/artifacts/code/
```

17. PROVIDER STATE MACHINE

```
ST-001 | START -> DISCOVER
ST-002 | DISCOVER -> AUTH_ERROR when authentication fails
ST-003 | DISCOVER -> QUOTA_ERROR when quota is exhausted
ST-004 | DISCOVER -> READY when models and capabilities are returned
ST-005 | READY -> DEGRADED after repeated latency, malformed output, or tool-call failures
ST-006 | DEGRADED -> CIRCUIT_OPEN after threshold failures
ST-007 | CIRCUIT_OPEN -> HALF_OPEN after cooldown
ST-008 | HALF_OPEN -> READY after successful probes
ST-009 | HALF_OPEN -> CIRCUIT_OPEN after failed probes
```

18. COST POLICY

```
CORE-0040 | subsystem=cost-estimation | directive=Paid OpenRouter credits are consumed deliberately, not
    indiscriminately | evidence=required | failure=typed-and-logged
CORE-0041 | subsystem=cost-estimation | directive=The system estimates tokens, images, seconds, resolution, and
    provider-specific fixed charges before execution | evidence=required | failure=typed-and-logged
CORE-0042 | subsystem=cost-estimation | directive=Cheap drafting plus premium verification is preferred over
    premium generation for every intermediate step | evidence=required | failure=typed-and-logged
CORE-0043 | subsystem=cost-estimation | directive=Premium models are reserved for bottlenecks: architecture,
    high-stakes reasoning, final synthesis, hard debugging, and final creative direction | evidence=required |
    failure=typed-and-logged
CORE-0044 | subsystem=cost-estimation | directive=Free models may expand candidate diversity, but final outputs
    still require verification proportional to risk | evidence=required | failure=typed-and-logged
CORE-0045 | subsystem=cost-estimation | directive=Media generation uses low-cost previews before high-resolution
    or long-duration renders | evidence=required | failure=typed-and-logged
```

19. QUALITY POLICY

```
CORE-0046 | directive=Quality is multi-dimensional: correctness, completeness, originality, coherence,
    usability, aesthetics, message fidelity, and evidence
CORE-0047 | directive=The system does not equate verbosity with depth
CORE-0048 | directive=Creative work must express a thesis, not merely aggregate fashionable visual tokens
CORE-0049 | directive=Agentic work must produce tested state changes, not only plans
CORE-0050 | directive=Research outputs separate source evidence from synthesis and speculation
CORE-0051 | directive=Coding outputs require lint, type, test, security, and runtime evidence where applicable
```

20. AGI FATTAH ORIENTATION

```
CORE-0052 | directive=Fattah means opening blocked problem spaces through disciplined decomposition, tool use,
    multi-perspective generation, adversarial critique, and verified action
CORE-0053 | directive=The OS pursues expandable agency without claiming sentience, infallibility, or literal AGI
CORE-0054 | directive=Intelligence operationally measured by: solved constraints, transferred skills, reduced
    error, improved novelty, efficient resource routing
CORE-0055 | directive=Hyperloops generate candidate skills; promotion depends on evidence
CORE-0056 | directive=Luminous creativity through message-bearing structure, not ornamental randomness
```

21. MOUSSAIF VISUAL INTELLIGENCE STACK (9 LAYERS)

```
VS-001 | layer=1 | name=Semantic Nucleus | focus=Single idea the visual must make felt
VS-002 | layer=2 | name=Perceptual Hierarchy | focus=What is seen first, second, third
VS-003 | layer=3 | name=Symbolic Tension | focus=Contradiction, transformation, scale, absence, impossible
    relation
VS-004 | layer=4 | name=Spatial Rhetoric | focus=Framing, depth, negative space, rhythm, directional force
VS-005 | layer=5 | name=Chromatic Logic | focus=Color as meaning, not decoration
VS-006 | layer=6 | name=Material Logic | focus=Texture and light aligned with concept
VS-007 | layer=7 | name=Transition Logic | focus=Motion or sequence that changes interpretation
VS-008 | layer=8 | name=Audience Cognition | focus=Curiosity, comprehension, emotion, recall
VS-009 | layer=9 | name=Ethical/Message Audit | focus=No accidental false claim or manipulative ambiguity
```

22. CANONICAL PROMPT ENVELOPE

```
objective: exact desired state
context: only relevant verified context
constraints: hard boundaries
inputs: typed assets and references
operations: permitted transformations
tools: allowed interfaces
model_contract: required modalities and capabilities
quality_bar: measurable criteria
verification: tests and critics
budget: money, tokens, latency, iterations
output_schema: machine-readable deliverable
stop_conditions: success, plateau, ceiling, interrupt
memory_policy: what may be persisted
```

23. SOURCE-OF-TRUTH REFERENCES

```
REF-001 | https://openrouter.ai/api/v1/models
REF-002 | https://openrouter.ai/docs/api/api-reference/credits/get-credits
REF-003 | https://openrouter.ai/docs/guides/overview/multimodal/image-generation
REF-004 | https://openrouter.ai/docs/api/api-reference/images/list-image-models
REF-005 | https://openrouter.ai/docs/api/api-reference/video-generation/list-videos-models
REF-006 | https://openrouter.ai/docs/cookbook/video-generation/choose-video-model
REF-007 | https://docs.api.nvidia.com/nim/docs/api-quickstart
REF-008 | https://docs.nvidia.com/nim/large-language-models/latest/api-reference.html
REF-009 | https://docs.nvidia.com/nim/visual-genai/latest/api/openai-image-generation.html
REF-010 | https://docs.nvidia.com/nim/cosmos/latest/api-reference.html
REF-011 | https://build.nvidia.com/models
```

24. COMPLETE CORE DIRECTIVE INDEX (CORE-0001..CORE-0400)

Total: 400 subsystem mandates across 20 categories

Category	CORE Range	Count	Key Subsystems
Operating Contract	0001-0015	15	provider-discovery, credit-accounting, overlap-detection, capability-filtering, cost-estimation, loop-termination, self-improvement, memory-write, secret-redaction, tool-security
Routing States	0016-0025	10	openrouter, nvidia, credit states
Paid-First Logic	0019-0025	7	overlap-detection, overlap-record confidence
Image Orchestration	0026-0032	7	image-direction, image-editing, capability-filtering
Video Orchestration	0033-0039	7	video-selection, video-polling, cosmos
Hyperloop Invariant	0040-0045	6	artifact typing, provenance, critic, repair, memory
Hyperskill Invariant	0046-0050	5	versioning, privilege escalation, promotion, decay
Memory Architecture	0051-0056	6	episodic, semantic, procedural, strategic, negative, benchmark
DSL Grammar	0057-0061	5	TYPE values, ID stability, parser behavior
Bootstrap FS	0062-0093	32	config, registries, skills, loops, prompts, memory, logs, tests, artifacts
Provider State Machine	0094-0101	8	DISCOVER, READY, DEGRADED, CIRCUIT_OPEN, HALF_OPEN
Cost Policy	0102-0106	5	credit accounting, estimation, drafting vs premium, free diversity, media previews
Quality Policy	0107-0111	5	multi-dimensional, verbosity, creative thesis, agentic state changes, research separation, coding evidence
AGI Fattah	0112-0116	5	agency measurement, hyperloop skill generation, luminous creativity
Visual Intelligence	0117-0125	9	9-layer stack
Prompt Envelope	0126	1	canonical structure
Source References	0127-0137	11	OpenRouter + NVIDIA docs
Subsystem Mandates	0138-0400	263	provider-discovery, credit-accounting, model-normalization, overlap-detection, capability-filtering, quality-scoring, cost-estimation, latency-control, tool-security, secret-redaction, memory-write, memory-retrieval, benchmarking, image-direction, image-editing, video-selection, video-polling, code-execution, research-verification, artifact-validation, human-gate, incident-response, self-improvement, skill-promotion, loop-termination, context-compression

---

End of Technical Appendix - All DSL specifications and CORE directives from the 100K-line Moussaif/Hermes AGI orchestration OS integrated into the EBYOND.AI.ERA ebook system.