MOUSSAIF AGI OS / MOUSSAIXT Deep Engineering Architecture

المعمارية الهندسية العميقة لنظام MOUSSAIF AGI OS / MOUSSAIXT

A 50-page bilingual Arabic–English systems report specifying the cognitive kernel, HyperKnowledge substrate, MCD/QMCD/MCAI routing, HAKIM-MCD pipeline, HyperSkills, HyperAbilities, HyperLoops, agent mesh, tool contracts, persistent memory, verification, world-effect control, evolution laboratory, deployment and cross-environment runtime.

تقرير هندسي ثنائي اللغة من خمسين صفحة يحدد طبقات التفكير والمعرفة والمهارات والقدرات والحلقات والوكلاء والأدوات والذاكرة والتحقق والنشر والتطور والتشغيل العابر للبيئات.

AGI-oriented architecture — not a claim of demonstrated AGI

Architecture basis. Synthesized from the user’s canonical MOUSSAIF corpus and current runtime documents. “MOUSSAIXT” and “HyperAbility” are formalized as architecture-level abstractions where the corpus is distributed or not yet a single enforced runtime primitive. تم التفريق بين العناصر الموثقة والعناصر التي تم توحيدها معماريا في هذا التقرير.

Architecture Index / فهرس المعمارية

فهرس المعمارية

01 Cover / الغلاف

03 System Identity, Boundary & Claim Discipline

04 Macro Architecture: 12-Layer Stack

05 Version Lineage & Canonical Strata

06 Constitutional Precedence Kernel

07 Authorization, Reversibility & Side-Effect Gates

08 Objective Hierarchy & Governing Question Engine

09 Active Difference Extraction

10 MCD: Meta-Complexity / Difference Scoring

11 QMCD: Question-Weighted Meta-Complexity

12 MCAI: Meta-Cognitive Agentic Integration

13 HAKIM-MCD Pipeline

14 Vascular Compute & Sparse HyperRouting

15 Project World Model OS

16 State Objects & Shared Runtime Schema

17 HKOS / Knowledge-as-OS

18 Knowledge Vessel Ontology & Concept Compiler

19 RAG, GraphRAG & Retrieval Fabric

20 Cloud Memory Bridge & Context Control Plane

21 HyperSkill Engineering Model

22 HyperAbility: Composed Capability Layer

23 Bounded HyperLoop

24 HyperDeployment & Knowledge-Integrity AutoDeployment

25 Agent Mesh & Role Councils

26 Cognitive Router

27 Tool Contracts & Framework Adapters

28 Mission Graph & Plan DAG

29 Executor Loop

30 Courts: Gap, Falsification & Demolition

31 Evidence Hierarchy & Claim Contract

32 Verification & World-Effect Loop

33 Self-Reconstruction Memory

34 HyperSkill Factory & Capability Extraction

35 Evolution Laboratory

36 Boot Sequence, Integrity & Self-Test

37 Runtime Discovery & Environment Fingerprinting

38 MOUSSAIXT Cross-Environment Execution Shell

39 Google Drive Persistent Substrate

40 Observability, Trace & Audit Plane

41 Security, Privacy & Integrity Architecture

42 Data & Artifact Lifecycle

43 Failure Recovery, Repair & Rollback

44 Performance, Token Economy & Scale

45 M-TAOS: Task HyperKnowledge OS

46 Research & Theory Construction Kernel

47 Domain Adapters: Research, Brands, Career & Operations

48 Reference Interfaces & Machine-Readable Contracts

49 Maturity Model, Open Gaps & Engineering Roadmap

50 Glossary, Final Architecture Contract & Source Anchors

System Identity, Boundary & Claim Discipline

هوية النظام وحدوده وانضباط الادعاء

ENGINEERING DEFINITION

MOUSSAIF is treated as a family of versioned cognitive/runtime artifacts rather than one monolithic program. The canonical MOUSSAIF1B package defines four operating-system documents plus a registry of 300 HyperSkills. Earlier and parallel strata include the HyperKnowledge OS V3 ontology, HyperLoop Runtime OS V3, M-TAOS task OS, and the MCD/QMCD/MCAI/HAKIM reasoning lineage. MOUSSAIXT is formalized in this report as the cross-environment execution shell that binds these layers to ChatGPT-class tools, Hermes-style local runtimes, coding agents, and cloud memory.

CORE CONTRACTS / COMPONENTS

- Boundary: AGI-oriented architecture ≠ demonstrated AGI
- Canonical state must be versioned and provenance-linked
- Observed state outranks remembered or generated state
- A generated artifact is not equivalent to a deployed artifact
- External effects require tool evidence and post-action inspection

> **Engineering implication.** Design consequence: all claims in the stack should be attributable to a source stratum and an implementation status.

الشرح الهندسي

MOUSSAIF ليس ملفا واحدا بل عائلة من طبقات وإصدارات مترابطة. النواة الحالية MOUSSAIF1B تتكون من أربعة أنظمة تشغيل معرفية وسجل مهارات. توجد طبقات سابقة/موازية مثل HyperKnowledge OS V3 وHyperLoop Runtime OS V3 وM-TAOS ومنهجيات MCD/QMCD/MCAI/HAKIM. في هذا التقرير يُستخدم MOUSSAIXT كاسم لطبقة التنفيذ العابرة للبيئات التي تربط هذه النواة بالأدوات والوكلاء والذاكرة السحابية.

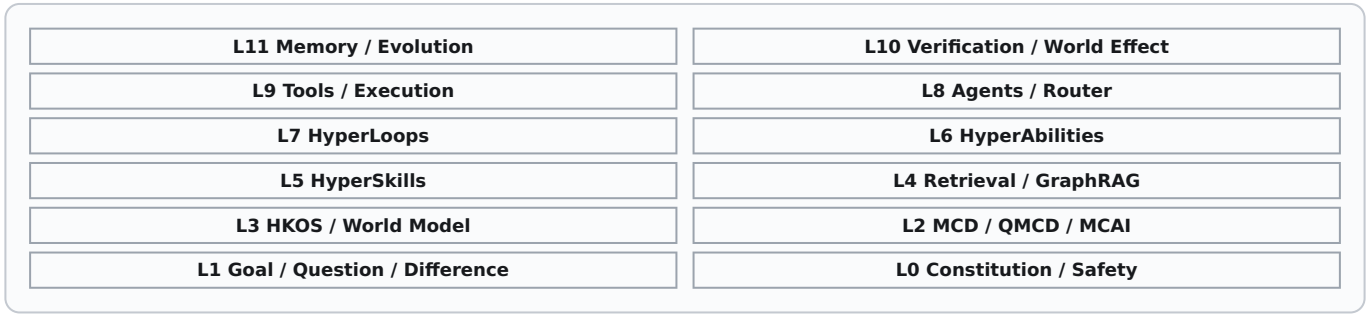
العقود والعناصر الأساسية

- الحد: التوجه نحو AGI لا يعني إثبات AGI
- كل حالة معيارية يجب أن تكون مؤرخة ومرتبطة بالمصدر
- الحالة المرصودة أعلى من الذاكرة أو التوليد
- الملف المُنشأ لا يساوي نشره أو تشغيله
- الأثر الخارجي يحتاج دليلا من الأداة وفحصا بعد التنفيذ

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Macro Architecture: 12-Layer Stack

المعمارية الكبرى: اثنتا عشرة طبقة



ENGINEERING DEFINITION

The system is best understood as a layered stack with explicit contracts between strata. Higher layers may request capabilities from lower layers but cannot bypass constitutional, authorization, evidence, or verification gates. The stack supports lazy loading: only the smallest relevant knowledge, skills, agents, and tools are injected for the current mission. This minimizes context cost while preserving reconstructability through registries and provenance links.

CORE CONTRACTS / COMPONENTS

- L0 Constitution & Safety
- L1 Goal / Governing Question
- L2 Difference & MCD/QMCD/MCAI
- L3 HKOS / World Model
- L4 Retrieval / GraphRAG
- L5 HyperSkills
- L6 HyperAbilities
- L7 HyperLoops / Workflows
- L8 Agents / Router
- L9 Tools / Execution Adapters
- L10 Verification / World Effect
- L11 Memory / Evolution / Observability

> **Engineering implication.** Layer violations are architecture defects. A host adapter or agent cannot bypass L0 constitutional constraints.

الشرح الهندسي

أفضل تمثيل للنظام هو مكدس طبقي بعقود واضحة. الطبقات العليا لا يجوز لها تجاوز الدستور أو التفويض أو الدليل أو التحقق. التحميل كسول Lazy: يتم جلب أقل مجموعة لازمة من المعرفة والمهارات والوكلاء والأدوات، مع بقاء القدرة على إعادة بناء السياق عبر السجلات وروابط المصدر.

العقود والعناصر الأساسية

- L0 الدستور والسلامة
- L1 الهدف والسؤال الحاكم
- L2 الفرق وMCD/QMCD/MCAI
- L3 HKOS ونموذج العالم
- L4 الاسترجاع وGraphRAG
- L5 HyperSkills
- L6 HyperAbilities
- L7 HyperLoops وسير العمل
- L8 الوكلاء والموجّه
- L9 الأدوات ومحولات التنفيذ
- L10 التحقق والأثر الخارجي
- L11 الذاكرة والتطور والرصد

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Version Lineage & Canonical Strata

سلالة الإصدارات والطبقات المعيارية

ENGINEERING DEFINITION

Architecture integrity requires distinguishing lineage from current runtime. The cognitive lineage supplies methods such as Difference Before Volume, Governing Question, MCD, QMCD, MCAI, vascular routing, ROFSC compression, falsification courts, and self-reconstruction memory. The V3 HyperKnowledge and HyperLoop artifacts package these ideas as reusable OS knowledge. MOUSSAIF1B adds a stricter four-OS kernel, authorization tiers, world-model objects, router/executor contracts, and an evolution laboratory.

CORE CONTRACTS / COMPONENTS

- Lineage layer: theories, heuristics, research constructs
- Ontology layer: definitions, boundaries, epistemic tiers
- Runtime layer: skills, loops, agents, tools, plans
- Control layer: authorization, courts, verification, rollback
- Evolution layer: trace datasets, benchmarks, promotion gates

> **Engineering implication.** Version reconciliation should operate on semantic lineage, not filenames alone.

الشرح الهندسي

سلامة المعمارية تتطلب فصل التاريخ الفكري عن النسخة التنفيذية الحالية. سلالة التفكير توفر مبادئ الفرق قبل الحجم والسؤال الحاكم وMCD وQMCD وMCAI والتوجيه الوعائي وضغط ROFSC والمحاكم وذاكرة إعادة البناء. إصدارات V3 حولت ذلك إلى معرفة تشغيلية، بينما MOUSSAIF1B يضيف نواة أكثر صرامة من أربعة OS وعقود تفويض ونموذج عالم وموجه/منفذ ومختبر تطور.

العقود والعناصر الأساسية

- طبقة السلالة: النظريات والهيورستيك والفرضيات البحثية
- طبقة الأنطولوجيا: التعريفات والحدود والمستويات المعرفية
- طبقة التشغيل: المهارات والحلقات والوكلاء والأدوات والخطط
- طبقة التحكم: التفويض والمحاكم والتحقق والتراجع
- طبقة التطور: آثار التنفيذ والاختبارات وبوابات الترقية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Constitutional Precedence Kernel

نواة الأسبقية الدستورية

ENGINEERING DEFINITION

The constitutional layer is the highest-priority control plane. It prevents local optimization from overriding law, platform safety, explicit authorization, privacy boundaries, or truthfulness. The kernel resolves conflicts using a precedence chain: applicable law/platform policy → explicit authorization/privacy → MOUSSAIF constitution → current objective/constraints → project policy → active plan → tool convenience → optimization heuristics. This ordering is architectural, not rhetorical: the router and executor must fail closed when a lower layer conflicts with a higher one.

CORE CONTRACTS / COMPONENTS

- Invariant: constitution before optimization
- Invariant: mechanism before grandeur
- Invariant: smallest complete reversible action
- Invariant: trace or it did not happen
- Invariant: honest PARTIAL/BLOCKED over fabricated success

> **Engineering implication.** The constitution is evaluated before any optimization score or agent preference.

الشرح الهندسي

الطبقة الدستورية هي أعلى Control Plane. تمنع أي تحسين محلي من تجاوز القانون أو أمان المنصة أو التفويض أو الخصوصية أو الصدق. ترتيب الأسبقية يوجه الموجّه والمنفذ فعليا، وعند التعارض يجب أن يفشل النظام بشكل آمن بدلا من التحايل على قيد أعلى.

العقود والعناصر الأساسية

- ثابت: الدستور قبل التحسين
- ثابت: الآلية قبل الخطاب الكبير
- ثابت: أصغر فعل كامل قابل للتراجع
- ثابت: لا أثر بلا سجل
- ثابت: PARTIAL/BLOCKED أفضل من نجاح مُختلق

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Authorization, Reversibility & Side-Effect Gates

التفويض وقابلية التراجع وبوابات الأثر

ENGINEERING DEFINITION

MOUSSAIF1B uses authorization tiers to separate reasoning from external side effects. T0 permits public reasoning without mutation; T1 allows approved reads, simulations, and local artifact creation; T2 covers explicitly authorized reversible local changes with checkpoint, diff, tests, and rollback; T3 requires explicit step-level authorization for public, production, financial, credential-bearing, messaging, privacy-sensitive, or third-party effects; T4 is prohibited. Capability does not imply authorization.

CORE CONTRACTS / COMPONENTS

- Gate inputs: requested action, target, reversibility, sensitivity, blast radius
- Gate outputs: ALLOW / DOWNGRADE_TO_DRY_RUN / REQUIRE_APPROVAL / BLOCK
- Checkpoint must exist before mutable high-risk work
- Rollback remains valid until verification/ stabilization
- Secrets are referenced, never persisted as plaintext knowledge

> **Engineering implication.** Authorization is a runtime input and can downgrade a plan without changing the underlying capability.

الشرح الهندسي

تقسم المعمارية الصلاحيات إلى مستويات T0–T4. القدرة التقنية على الفعل لا تعني الإذن. كل تعديل ذي أثر يجب أن يحمل نقطة حفظ، فرق Diff، اختبارا ومسار تراجع، أما النشر أو الرسائل أو الإنتاج أو البيانات الحساسة فتحتاج تفويضا صريحا على مستوى الخطوة.

العقود والعناصر الأساسية

- مدخلات البوابة: الفعل والهدف وقابلية التراجع والحساسية ونطاق الضرر
- المخرجات: سماح أو Dry Run أو طلب موافقة أو حظر
- نقطة الحفظ تسبق التعديل عالي المخاطر
- يبقى التراجع صالحا حتى اكتمال التحقق
- الأسرار تُخزّن كمراجع لا كنصوص داخل المعرفة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Objective Hierarchy & Governing Question Engine
هرمية الهدف ومحرك السؤال الحاكم

ENGINEERING DEFINITION

A mission is not routed from the user’s surface wording alone. The objective layer reconstructs the desired world state, definition of done, affected systems, constraints, value criterion, and unresolved uncertainty. It then generates the highest-value governing question. Question quality is judged by diagnostic power, information gain, transferability, actionability, proof potential, contextual fit, and ambiguity/noise penalty. The question may recurse after each state change.

CORE CONTRACTS / COMPONENTS

- Input: user objective + current state + constraints
- Output: governing_question + definition_of_done + value_model
- Recursive rule: every answer may yield a better next question
- Stop when additional questioning has lower expected value than action
- Persist only decisive question-state transitions, not hidden chain-of-thought

> **Engineering implication.** The governing question is stored as mission metadata and is allowed to change after evidence events.

الشرح الهندسي

لا يبدأ التنفيذ من صياغة الطلب السطحية فقط، بل من إعادة بناء الحالة الخارجية المطلوبة وتعريف الإنجاز والقيود والقيمة وعدم اليقين. بعدها يُنشأ السؤال الحاكم الأعلى قيمة. جودة السؤال تقاس بقدرته التشخيصية ومكسب المعلومات وقابلية العمل والإثبات وملاءمة السياق وعقوبة الضوضاء.

العقود والعناصر الأساسية

- المدخل: هدف المستخدم والحالة الحالية والقيود
- المخرج: السؤال الحاكم وتعريف الإنجاز ونموذج القيمة
- قاعدة تكرارية: كل جواب قد يولد سؤالا أفضل
- التوقف عندما تصبح قيمة السؤال الإضافي أقل من قيمة الفعل
- تُحفظ تحولات القرار لا سلسلة التفكير السرية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Active Difference Extraction

استخراج الفرق النشط

ENGINEERING DEFINITION

Difference Before Volume is the first computational triage. The system identifies what is new, changed, conflicting, missing, high-leverage, high-dependency, externally actionable, or merely repetitive noise. Instead of applying equal reasoning to all tokens or artifacts, it produces an active-difference set ΔA that drives retrieval, compute depth, skill routing, and verification. This is the principal anti-bloat mechanism in the MOUSSAIF cognitive lineage.

CORE CONTRACTS / COMPONENTS

- ΔA classes: novelty, contradiction, gap, dependency, risk, leverage, world-effect
- Noise classes: duplicates, stale mirrors, low-value metadata, decorative verbosity
- Priority grows with information gain × consequence × centrality × irreversibility
- Difference IDs link source artifact → claim → plan node → result

> **Engineering implication.** Difference extraction is the point where scale is reduced before expensive reasoning begins.

الشرح الهندسي

مبدأ «الفرق قبل الحجم» يحدد ما الجديد والمتغير والمتعارض والمفقود ومرتفع الرافعة أو الاعتماد وما يمكنه تغيير العالم الخارجي وما هو تكرار. تُنتج مجموعة فرق نشط ΔA توجه الاسترجاع وعمق الحساب واختيار المهارات والتحقق، ولذلك فهي آلية أساسية ضد تضخم السياق.

العقود والعناصر الأساسية

- أنواع ΔA: جدة، تعارض، فجوة، اعتماد، خطر، رافعة، أثر خارجي
- الضوضاء: التكرار والنسخ القديمة والميتاداتا منخفضة القيمة والحشو
- الأولوية ترتفع مع مكسب المعلومات والنتيجة والمركزية وكلفة عدم التراجع
- معرف الفرق يربط المصدر بالادعاء وعقدة الخطة والنتيجة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

MCD: Meta-Complexity / Difference Scoring

MCD: مقياس الفرق والتعقيد الفوقي

ENGINEERING DEFINITION

MCD is a routing heuristic for deciding where cognition deserves more resources. The canonical remembered form is MCD = λ1D + λ2I + λ3T + λ4A + λ5P + λ6C − λ7N. D/I/T/A/P/C/N are project-defined dimensions; the important engineering rule is not the names alone but documented semantics, weights, normalization, and calibration. MCD should influence depth, retrieval breadth, model/tool choice, test intensity, and review requirements; it must never be presented as a mystical intelligence score.

CORE CONTRACTS / COMPONENTS

- Engineering contract: dimensions are explicit and bounded
- Weights are versioned with calibration evidence
- Score affects routing, not truth status
- High score cannot bypass evidence or authorization
- Record router regret to improve future calibration

```
MCD = λ1D + λ2I + λ3T + λ4A + λ5P + λ6C − λ7N
```

> **Engineering implication.** MCD must be calibrated per domain and audited for routing bias.

الشرح الهندسي

MCD هي دالة توجيه للموارد المعرفية: MCD = λ1D + λ2I + λ3T + λ4A + λ5P + λ6C − λ7N + أبعادها وأوزانها يجب أن تُعرّف داخل المشروع وتعاير. دورها هو تحديد أين نزيد الاسترجاع والتفكير والأدوات والاختبارات، لا قياس «ذكاء» غامض.

العقود والعناصر الأساسية

- العقد الهندسي: الأبعاد معرفة ومحدودة
- الأوزان مؤرخة ومدعومة بالمعايرة
- النتيجة توجه المسار ولا تثبت الحقيقة
- ارتفاع MCD لا يتجاوز الدليل أو التفويض
- يُسجل Router Regret لتحسين المعايرة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

QMCD: Question-Weighted Meta-Complexity

QMCD: التعقيد الموزون بجودة السؤال

ENGINEERING DEFINITION

QMCD extends MCD by weighting cognition through the quality of the governing question. Canonical forms include MCD_Q = Ω(Q) × MCD and Q_depth = (q_d + q_i + q_t + q_a + q_p + q_c)/(1+q_n). Operationally, Ω(Q) raises compute when the question is discriminative, testable, high-leverage, and low-noise, while weak questions suppress costly exploration. QMCD is also used as a superposition keeper: multiple plausible interpretations remain live until evidence collapses the branch set.

CORE CONTRACTS / COMPONENTS

- Q features: diagnosticity, information gain, temporality, agency, proofability, context
- Branch set remains explicit: {Q1...Qn} with confidence and discriminators
- Collapse condition: evidence or test separates branches
- Failure mode: premature certainty → wrong retrieval and wrong agency

```
MCD_Q = Ω(Q) × MCD
Q_depth = (q_d + q_i + q_t + q_a + q_p + q_c) /
(1 + q_n)
```

> **Engineering implication.** QMCD prevents expensive reasoning from being spent on a poorly formed question.

الشرح الهندسي

QMCD تضرب MCD في جودة السؤال الحاكم. السؤال القوي يزيد كثافة الاسترجاع والتوليد والاختبار، والسؤال الضعيف يخفض كلفة الاستكشاف. كما تعمل QMCD كحافظ لتعدد الفرضيات: لا يتم إغلاق الفروع المحتملة مبكرا قبل وصول الدليل الكافي.

العقود والعناصر الأساسية

- خصائص Q: التشخيص ومكسب المعلومات والزمن والوكالة والإثبات والسياق
- مجموعة الفروع تبقى صريحة مع الثقة ومعايير الفصل
- شرط الانهيار: دليل أو اختبار يفصل الفروع
- نمط الفشل: اليقين المبكر يؤدي إلى استرجاع ووكالة خاطئين

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

MCAI: Meta-Cognitive Agentic Integration

MCAI: التكامل الوكيلي فوق المعرفي

ENGINEERING DEFINITION

MCAI is the candidate-comparison and agency-allocation layer. For uncertain tasks it generates multiple models, plans, or perspectives and scores them on explanatory coverage, evidence, feasibility, cost, reversibility, failure modes, downstream effects, and contradiction with canonical state. It then selects, composes, or preserves disagreement. MCAI is where a cognitive difference becomes an allocation decision: which model, agent, tool, memory source, court, and output form should receive the problem.

CORE CONTRACTS / COMPONENTS

- Candidate object: id, hypothesis/plan, evidence, cost, risk, reversibility, expected value
- Selection is adversarial, not majority-vote
- Diversity is useful only when it increases discriminative coverage
- Unresolved disagreements remain first-class state

> **Engineering implication.** MCAI preserves dissent when evidence cannot justify collapse to one candidate.

الشرح الهندسي

MCAI هي طبقة مقارنة المرشحين وتوزيع الوكالة. عند وجود عدم يقين تُنشئ عدة خطط أو نماذج وتفحص التغطية والدليل وقابلية التنفيذ والكلفة والتراجع وأنماط الفشل والأثر والتعارض مع الحالة المعيارية. ثم تختار أو تركب أو تحتفظ بالخلاف. هنا يتحول الفرق المعرفي إلى قرار: أي وكيل وأداة ومصدر ذاكرة ومحكمة ومخرج يستقبل المهمة.

العقود والعناصر الأساسية

- كائن المرشح: معرف، فرضية/خطة، دليل، كلفة، خطر، تراجع، قيمة متوقعة
- الاختيار خصومي لا تصويت أغلبية سطحي
- التنوع مفيد عندما يزيد التغطية التمييزية
- الخلاف غير المحسوم يبقى جزءا من الحالة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

HAKIM-MCD Pipeline

خط HAKIM-MCD



ENGINEERING DEFINITION

The canonical HAKIM-MCD route converts cognition into bounded agency: Difference Extraction → MCD Scoring → Vascular Compute Routing → Candidate Paths → Falsification Court → Tool/GraphRAG Retrieval → Agency Selection → World-Effect Loop → Self-Reconstruction Memory. The pipeline is explicitly not complete at text generation. Its closure condition is a verified state transition or a truthful PARTIAL/BLOCKED outcome with evidence and residual uncertainty recorded.

CORE CONTRACTS / COMPONENTS

- Each stage has typed input/output
- Stage skipping requires explicit rationale
- Court may route backward for repair
- External action happens only after authorization gate
- Memory writeback records the minimal trajectory needed for continuation

> **Engineering implication.** HAKIM-MCD is a closed-loop control architecture rather than a text-generation recipe.

الشرح الهندسي

المسار المعياري HAKIM-MCD يحول التفكير إلى وكالة محدودة: استخراج الفرق ← MCD ← التوجيه الوعائي ← المسارات المرشحة ← محكمة التفنيد ← الأدوات/GraphRAG ← اختيار الوكالة ← حلقة الأثر الخارجي ← ذاكرة إعادة البناء. لا ينتهي المسار عند توليد نص بل عند تحقق تغير الحالة أو إعلان جزئي/محظور صادق.

العقود والعناصر الأساسية

- كل مرحلة لها مدخل ومخرج نمطي
- تجاوز مرحلة يحتاج تعليلا صريحا
- المحكمة قد تعيد المسار للخلف للإصلاح
- الفعل الخارجي يأتي بعد بوابة التفويض
- الذاكرة تحفظ الحد الأدنى من المسار اللازم للاستمرار

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Vascular Compute & Sparse HyperRouting

الحوسبة الوعائية والتوجيه الفائق المتناثر

ENGINEERING DEFINITION

Vascular routing models computation as selective blood flow rather than flooding every branch. It prioritizes high-centrality artifacts, unresolved contradictions, novel evidence, high-value decisions, fragile dependencies, security/integrity risks, and high-reuse nodes. Sparse HyperRouting represents dependencies as a graph and activates the minimal high-value path while retaining branchability when evidence changes. This is the principal performance architecture for large knowledge stores.

CORE CONTRACTS / COMPONENTS

- Routing features: centrality, novelty, uncertainty, risk, expected information gain, reuse
- Context budget is allocated per branch
- Cold nodes remain addressable by registry pointer
- Re-expansion occurs only after trigger evidence
- Metric: useful-value-per-context-token and verified-success-per-tool-call

> **Engineering implication.** Sparse activation allows very large external knowledge bases without proportional prompt growth.

الشرح الهندسي

التوجيه الوعائي يتعامل مع الحساب كتدفق انتقائي لا كإغراق لكل الفروع. الأولوية للعقد المركزية والتعارضات والدليل الجديد والقرارات عالية القيمة والاعتمادات الهشة ومخاطر السلامة والعناصر مرتفعة إعادة الاستخدام. Sparse HyperRouting يفعل أقصر مسار عالي القيمة مع إبقاء قابلية التفرع عند تغير الدليل.

العقود والعناصر الأساسية

- خصائص التوجيه: المركزية والجدة وعدم اليقين والخطر ومكسب المعلومات وإعادة الاستخدام
- ميزانية السياق توزع لكل فرع
- العقد الباردة تبقى قابلة للعنونة عبر السجل
- إعادة التوسعة تحصل عند ظهور دليل محفز
- مقاييس: قيمة مفيدة لكل توكن ونجاح متحقق لكل استدعاء أداة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Project World Model OS

نظام نموذج عالم المشروع

ENGINEERING DEFINITION

The Project World Model OS is the authoritative operational representation of projects and environments. It tracks approved active projects, relationships, current versus desired state, repositories, branches, remotes, deployments, environments, assets, objectives, metrics, dependencies, constraints, provenance, confidence, freshness, and sensitivity. Proposed, current, and historical state are stored separately. Unknown URLs, repositories, owners, or deployments are discovery_required rather than invented.

CORE CONTRACTS / COMPONENTS

- Entity types: project, repo, deployment, environment, artifact, objective, dependency, owner
- Edge types: depends_on, deployed_as, derived_from, supersedes, blocks, verifies
- Freshness policy determines when observed state expires
- State transitions require source + timestamp + confidence + actor/tool

> **Engineering implication.** Every tool write should produce a world-model delta only after verification.

الشرح الهندسي

نظام نموذج العالم يمثل الحالة التشغيلية الموثوقة للمشروعات والبيئات: المشاريع والعلاقات والحالة الحالية والمطلوبة والمستودعات والفروع والنشر والبيئات والأصول والأهداف والمقاييس والاعتمادات والقيود والمصدر والثقة والحداثة والحساسية. تُفصل الحالة المقترحة عن الحالية والتاريخية، وأي مجهول يصبح discovery_required لا قيمة مختلقة.

العقود والعناصر الأساسية

- أنواع الكيانات: مشروع ومستودع ونشر وبيئة وأثر وهدف واعتماد ومالك
- العلاقات: يعتمد على، منشور كـ، مشتق من، يستبدل، يحجب، يتحقق من
- سياسة الحداثة تحدد متى تنتهي صلاحية الحالة المرصودة
- كل انتقال حالة يحمل المصدر والزمن والثقة والفاعل/الأداة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

State Objects & Shared Runtime Schema

كائنات الحالة ومخطط وقت التشغيل

ENGINEERING DEFINITION

Runtime state is decomposed into typed objects rather than a single chat transcript. Canonical objects include identity_constitution, objective_hierarchy, assumption_ledger, claim_evidence_graph, gap_court_docket, authorization_context, value_model, session_goal, user_constraints, tool_inventory, execution_trace, unresolved_gap_register, and learning_candidates. Every state object carries id, version, updated_at, source, confidence, owner, sensitivity, freshness_policy, and dependencies.

CORE CONTRACTS / COMPONENTS

- Schema rule: immutable event + mutable current projection
- Provenance pointer is mandatory for high-impact fields
- Confidence is scoped, not global
- Sensitive fields use references/handles rather than copied secrets
- State can be reconstructed from event log + canonical artifacts

```
state_object:
  id: string
  version: semver
  source: provenance_ref
  confidence: 0..1
  sensitivity: public|internal|restricted
  freshness_policy: duration|event
  dependencies: [object_id]
```

> **Engineering implication.** Typed state allows multiple model hosts to share the same operational truth.

الشرح الهندسي

الحالة لا تُختزل في محادثة طويلة، بل في كائنات نمطية: الدستور، هرم الهدف، دفتر الافتراضات، رسم الادعاء-الدليل، ملف المحاكم، سياق التفويض، نموذج القيمة، جرد الأدوات، أثر التنفيذ، سجل الفجوات ومرشحي التعلم. لكل كائن معرف وإصدار وتاريخ ومصدر وثقة وحساسية وسياسة حداثة واعتمادات.

العقود والعناصر الأساسية

- قاعدة المخطط: حدث غير قابل للتغيير + إسقاط حالي قابل للتحديث
- رابط المصدر إلزامي للحقول عالية الأثر
- الثقة محلية لا عامة
- الحقول الحساسة تستخدم مراجع بدل نسخ الأسرار
- يمكن إعادة بناء الحالة من سجل الأحداث والأصول المعيارية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

HKOS / Knowledge-as-OS

HKOS / المعرفة كنظام تشغيل

ENGINEERING DEFINITION

HyperKnowledge OS transforms documents into an executable knowledge substrate. The canonical relationship chain is Domain → Subdomain → Concept → KB → Skill → Agent → Workflow → Evaluation → World Effect. A KB is incomplete if it is only long text; it must expose interfaces, procedures, validation, dependencies, failure modes, update mechanisms, retrieval metadata, and provenance. HKOS pages/chunks are treated as semantic cells that agents can retrieve without loading an entire corpus.

CORE CONTRACTS / COMPONENTS

- Knowledge unit: semantic chunk with stable ID and metadata
- Every KB exposes what it can answer and what it cannot
- Operational knowledge must map to at least one procedure or skill
- Version siblings are linked; duplicates are not silently copied
- Retrieval quality is measured by continuation fidelity and task utility

> **Engineering implication.** The unit of HKOS retrieval is a semantic operating cell, not an arbitrary page count.

الشرح الهندسي

HKOS يحول الوثائق إلى بنية معرفة قابلة للتنفيذ. السلسلة: مجال ← فرع ← مفهوم ← Workflow ← Agent ← Skill ← KB ← Evaluation ← World Effect. الـKB غير مكتمل إذا كان نصا طويلا فقط؛ يجب أن يكشف الواجهات والإجراءات والتحقق والاعتمادات وأنماط الفشل وآليات التحديث والميتاداتا والمصدر.

العقود والعناصر الأساسية

- وحدة المعرفة: Chunk دلالي بمعرف ثابت وميتاداتا
- كل KB يحدد ما يعرفه وما لا يعرفه
- المعرفة التشغيلية ترتبط بإجراء أو مهارة
- الإصدارات تربط كأشقاء ولا تُنسخ عشوائيا
- جودة الاسترجاع تقاس بقدرة الاستمرار والمنفعة العملية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Knowledge Vessel Ontology & Concept Compiler

أنطولوجيا أوعية المعرفة ومترجم المفاهيم

ENGINEERING DEFINITION

The core ontology introduces the knowledge vessel as a bounded container whose concepts must pass through definition, boundary, mechanism, measurement, implementation, court attack, and memory repair. The concept compiler prevents vague labels from becoming operational primitives. Each concept receives an identity, scope, parent/child relations, evidence tier, interfaces, tests, failure conditions, deployment forms, and anti-duplication links.

CORE CONTRACTS / COMPONENTS

- Concept states: proposed → defined → bounded → evidenced → operationalized → tested → canonical/deprecated
- Anti-vagueness rule: every noun-like primitive needs an observable contract
- Theory names are lineage until promoted by evidence and implementation

> **Engineering implication.** Concept promotion is gated by observable semantics and falsifiable boundaries.

الشرح الهندسي

تتعامل الأنطولوجيا مع المفهوم كوعاء معرفة محدود يجب أن يمر من التعريف والحد والآلية والقياس والتطبيق ومحكمة الهدم ثم إصلاح الذاكرة. Concept Compiler يمنع الأسماء الجذابة من التحول إلى عناصر تشغيلية قبل تعريفها واختبارها وربطها بالمصدر والواجهات والفشل والنشر.

العقود والعناصر الأساسية

- حالات المفهوم: مقترح ← معرف ← محدود ← مدعوم ← مشغل ← مختبر ← معياري/متقاعد
- قاعدة ضد الغموض: كل عنصر اسمي يحتاج عقدا قابلا للملاحظة
- أسماء النظريات تبقى ضمن السلالة حتى تثبت وتُنفذ

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

RAG, GraphRAG & Retrieval Fabric

نسيج RAG وGraphRAG والاسترجاع

ENGINEERING DEFINITION

Retrieval is hybrid: semantic chunks serve local factual/operational context, while GraphRAG provides relationship-aware traversal across projects, concepts, skills, agents, tools, and versions. Query planning begins from the governing question and active differences, then uses metadata filters, graph neighborhood expansion, and evidence ranking. Retrieval results are not truth by default; freshness, provenance, contradictions, and current observed state are checked before use.

CORE CONTRACTS / COMPONENTS

- Indexes: semantic vector, lexical, metadata, graph edges, recency, authority
- Retrieval packet: chunks + provenance + contradictions + confidence + missing evidence
- Cache only stable canonical summaries; invalidate stale operational state
- Graph edges encode lineage and execution dependencies

> **Engineering implication.** Retrieval output is evidence input, not an automatic truth assignment.

الشرح الهندسي

الاسترجاع هجين: المقاطع الدلالية تعطي السياق المحلي، وGraphRAG يعبر العلاقات بين المشاريع والمفاهيم والمهارات والوكلاء والأدوات والإصدارات. التخطيط يبدأ من السؤال الحاكم والفرق النشط ثم مرشحات الميتاداتا والتوسع في الرسم وترتيب الدليل. نتيجة الاسترجاع ليست حقيقة تلقائيا؛ تُفحص الحداثة والمصدر والتعارض والحالة المرصودة.

العقود والعناصر الأساسية

- الفهارس: دلالي ولفظي وميتاداتا ورسم وحداثة وسلطة
- حزمة الاسترجاع: مقاطع ومصدر وتعارض وثقة ودليل مفقود
- تُخزن الملخصات المستقرة وتُبطل الحالة التشغيلية القديمة
- حواف الرسم تمثل السلالة واعتمادات التنفيذ

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Cloud Memory Bridge & Context Control Plane

جسر الذاكرة السحابية وControl Plane للسياق

ENGINEERING DEFINITION

The cloud-memory bridge externalizes persistent state so model context remains small. A fast-boot control manifest points to the canonical context map, registries, latest-state summaries, skill/tool router, and only the smallest relevant KB slices. Multi-resolution representations follow ROFSC: L0 identity, L1 compact summary, L2 operational summary, L3 detailed specification, L4 source corpus. Compression quality is measured by continuation fidelity rather than token reduction alone.

CORE CONTRACTS / COMPONENTS

- Boot packet target: identity + current state + constraints + decisions + pointers
- Never inject full archives when retrieval pointers suffice
- Writeback updates latest-state registry and provenance graph
- Control plane must survive model/runtime replacement

> **Engineering implication.** The control plane should be compact enough for fast boot and rich enough for faithful continuation.

الشرح الهندسي

جسر الذاكرة السحابية يخرج الحالة الدائمة من نافذة النموذج. ملف Fast Boot يشير إلى خريطة السياق والسجلات والحالة الأخيرة وموجه المهارات/الأدوات وأصغر شرائح KB اللازمة. تمثيل ROFSC متعدد الدقة L0–L4 يتيح التوسع عند الحاجة، والجودة تقاس بقدرة وكيل جديد على متابعة العمل لا بعدد التوكنات المحذوفة.

العقود والعناصر الأساسية

- هدف حزمة الإقلاع: الهوية والحالة والقيود والقرارات والمؤشرات
- لا تُحقن الأرشيفات كاملة عند كفاية المؤشرات
- إعادة الكتابة تحدث سجل الحالة ورسم المصدر
- Control Plane يجب أن يبقى صالحا عند تبديل النموذج أو البيئة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

HyperSkill Engineering Model

نموذج هندسة HyperSkill

ENGINEERING DEFINITION

A HyperSkill is the smallest reusable executable knowledge unit. It defines identity, purpose, triggers, inputs, outputs, preconditions, procedure, tools, validation, failure modes, related skills, provenance, version, status, and when not to run. The current MOUSSAIF1B package expects 300 unique active skills distributed across four OS modules. Full skill bodies are loaded lazily; the router first works from compact registry metadata.

CORE CONTRACTS / COMPONENTS

- Skill ID is stable across versions; version is separate
- Skill output must be verifiable, not only textual
- Skills may compose but must declare conflicts and prerequisites
- Historical success updates only from verified runs
- Deprecation preserves lineage and migration path

```
hyperskill:
  id: HS-...
  trigger: ...
  inputs: {...}
  procedure: [...]
  tools: [...]
  validation: [...]
  failure_modes: [...]
  version: ...
  status: active|deprecated
```

> **Engineering implication.** Skills are atomic enough to route and test independently.

الشرح الهندسي

HyperSkill هي أصغر وحدة معرفة تنفيذية قابلة لإعادة الاستخدام. يجب أن تعرف هويتها وهدفها ومتى تعمل ومدخلاتها ومخرجاتها وشروطها وإجراءها وأدواتها والتحقق والفشل والعلاقات والمصدر والإصدار والحالة ومتى لا تعمل. سجل النواة يحوي مهارات فريدة ويُحمّل نص المهارة الكامل عند الحاجة فقط.

العقود والعناصر الأساسية

- معرف المهارة ثابت والإصدار منفصل
- مخرج المهارة يجب أن يكون قابلا للتحقق
- التركيب يصرح بالتعارضات والمتطلبات
- النجاح التاريخي يتحدث من تشغيلات متحققة فقط
- الإهمال يحفظ السلالة ومسار الهجرة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

HyperAbility: Composed Capability Layer

HyperAbility: طبقة القدرة المركبة

ENGINEERING DEFINITION

HyperAbility is formalized here as a composition abstraction above individual HyperSkills and below persistent agents/loops. It represents a durable capability such as “research-to-deployment,” “repository migration,” or “lead intelligence,” composed from multiple skills, tools, policies, and validation gates. Unlike a HyperSkill, an Ability owns a capability contract and compatibility envelope; unlike an Agent, it has no independent persona or long-lived goal. In the current corpus this is best treated as an architectural standardization layer, not a fully proven runtime primitive.

CORE CONTRACTS / COMPONENTS

- Ability contract: intent class, skill bundle, tool bundle, policy profile, validation suite
- Activation is router-driven and context-bounded
- Ability should expose cost/risk envelope and fallback path
- Ability version changes when composition semantics change

```
hyperability:
  id: HA-...
  intent_class: ...
  skills: [HS-...]
  tools: [...]
  policy_profile: ...
  validation_suite: ...
  fallback: ...
```

> **Engineering implication.** Abilities provide a stable composition interface while skills remain independently evolvable.

الشرح الهندسي

يُعرّف HyperAbility هنا كطبقة تركيب بين HyperSkill والوكيل/الحلقة. هي قدرة مستدامة مثل البحث حتى النشر، تتكون من عدة مهارات وأدوات وسياسات وبوابات تحقق. لا تملك شخصية أو هدفا دائما مثل Agent، بل عقد قدرة وحدود توافق. هذا التوصيف توحيد معماري مستنتج من السلالة وليس ادعاء بأن كل Runtime يطبقه فعليا الآن.

العقود والعناصر الأساسية

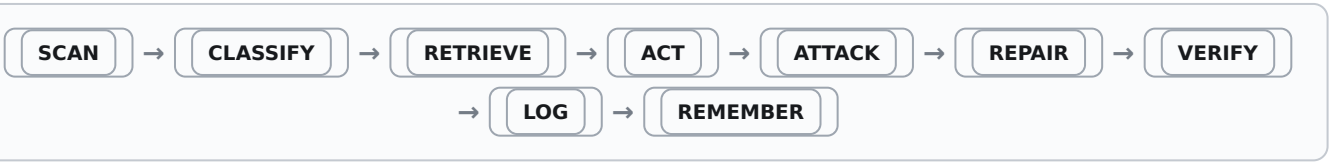
- عقد القدرة: فئة الهدف وحزمة المهارات والأدوات والسياسة والتحقق
- التفعيل بواسطة Router وفي سياق محدود
- تعلن القدرة كلفة/خطر ومسار بديل
- يتغير الإصدار عندما تتغير دلالة التركيب

> ملاحظة: HyperAbility في هذا التقرير طبقة توحيد هندسي مقترحة بين المهارات والوكلاء، وتحتاج Registry رسمي في Runtime المرجعي.

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Bounded HyperLoop

HyperLoop المحدود



ENGINEERING DEFINITION

A HyperLoop is a bounded iterative reasoning/execution controller. Required fields include goal, current state, stop condition, evidence threshold, failure threshold, retry budget, rollback/safe exit, memory writeback, and tool budget. A typical loop is scan → classify → retrieve → generate/act → attack → repair → verify → package → log → remember. Iteration exists to close evidence or state gaps, not to inflate output volume.

CORE CONTRACTS / COMPONENTS

- Loop state: INIT/RUNNING/WAITING_EVIDENCE/REPAIRING/VERIFIED/PARTIAL/BLOCKED/ROLLED_BACK
- Every iteration must change evidence, state, or plan
- No-progress detector terminates repetitive cycles
- External side effects remain separately authorized

```
hyperloop:
  goal: ...
  stop_condition: ...
  evidence_threshold: ...
  retry_budget: ...
  rollback: ...
  memory_writeback: ...
```

> **Engineering implication.** Loops require bounded termination and measurable progress.

الشرح الهندسي

HyperLoop هو متحكم تكراري محدود. يحدد الهدف والحالة وشرط التوقف وعتبة الدليل والفشل وميزانية المحاولات والتراجع وإعادة الكتابة وميزانية الأدوات. مثال: فحص ← تصنيف ← استرجاع ← توليد/فعل ← هجوم ← إصلاح ← تحقق ← تغليف ← سجل ← ذاكرة. التكرار لسد فجوة لا لزيادة الحجم.

العقود والعناصر الأساسية

- حالات الحلقة: INIT/RUNNING/WAITING_EVIDENCE/REPAIRING/VERIFIED/PARTIAL/BLOCKED/ROLLED_BACK
- كل دورة يجب أن تغير الدليل أو الحالة أو الخطة
- كاشف عدم التقدم يوقف التكرار العقيم
- الأثر الخارجي يحتاج تفويضا مستقلا

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

HyperDeployment & Knowledge-Integrity AutoDeployment

HyperDeployment والنشر التلقائي بسلامة المعرفة

ENGINEERING DEFINITION

HyperDeployment is the lifecycle that moves a generated capability into an operational state. The canonical artifact progression is REQUESTED → GENERATED → VERIFIED → STORED → LINKED/REGISTERED → OPERATIONAL. Each transition has evidence requirements. A file written locally is not “stored in Drive” until the storage operation is verified; a designed automation is not “active” until the scheduler confirms it; a deployment is not complete until smoke tests observe the intended external state.

CORE CONTRACTS / COMPONENTS

- Deployment bundle: artifact + manifest + provenance + tests + rollback + owner
- Registry link is mandatory before operational status
- World-effect verification closes the deployment
- Canary is preferred for high-risk new capability versions

> **Engineering implication.** Deployment status is a verified state machine, not a label.

الشرح الهندسي

HyperDeployment هو دورة انتقال القدرة من الطلب إلى التشغيل: REQUESTED ← GENERATED ← VERIFIED ← STORED ← LINKED/REGISTERED ← OPERATIONAL. لكل انتقال دليل إلزامي. إنشاء ملف محلي لا يعني حفظه في Drive، وتصميم Task لا يعني أنها نشطة، والنشر لا يكتمل قبل Smoke Test للحالة الخارجية.

العقود والعناصر الأساسية

- حزمة النشر: الأثر والمانيفست والمصدر والاختبارات والتراجع والمالك
- الربط بالسجل إلزامي قبل OPERATIONAL
- تحقق الأثر الخارجي يغلق النشر
- Canary مفضل للإصدارات الجديدة عالية المخاطر

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Agent Mesh & Role Councils

شبكة الوكلاء ومجالس الأدوار

ENGINEERING DEFINITION

Agents are long-lived or mission-scoped role executors that own goals, local state, and tool/capability access. The corpus uses councils such as Definition Judge, Boundary Cartographer, MCD Mapper, QMCD Keeper, MCAI Agency Architect, HAKIM Integrator, Runtime Orchestrator, HyperSkill Miner, Tool Contract Engineer, Framework Adapter, Memory Librarian, and Execution Auditor. In larger deployments these roles map into a 30-agent mesh, but routing should select only the minimum agents needed for the mission.

CORE CONTRACTS / COMPONENTS

- Agent contract: role, objective scope, capabilities, tools, memory view, permissions, handoff schema
- Coordinator is not allowed to fabricate worker completion
- Handoffs carry evidence and unresolved gaps, not only prose summaries
- Agents may be ephemeral; canonical state remains externalized

> **Engineering implication.** Agent count is a capacity ceiling, not a target for every mission.

الشرح الهندسي

الوكلاء ينفذون أدوارا ذات هدف وحالة محلية ووصول إلى القدرات والأدوات. تظهر أدوار مثل Definition Judge وBoundary Cartographer وMCD Mapper وHyperSkill Miner وTool Contract Engineer وMemory Librarian وExecution Auditor. حتى مع شبكة كبيرة يجب تشغيل أقل عدد لازم للمهمة.

العقود والعناصر الأساسية

- عقد الوكيل: الدور ونطاق الهدف والقدرات والأدوات وعرض الذاكرة والصلاحيات ومخطط التسليم
- المنسق لا يخترع إنجاز الوكلاء
- التسليم يحمل الدليل والفجوات لا الملخص فقط
- الوكيل قد يكون مؤقتا بينما الحالة المعيارية خارجية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Cognitive Router

الموجّه المعرفي

ENGINEERING DEFINITION

The router maps normalized missions to a minimal compatible composition of skills, abilities, agents, tools, and verification steps. MOUSSAIF1B defines a weighted score combining relevance, expected utility, compatibility, evidence availability, historical success, execution cost, security risk, context cost, and redundancy. The router first retrieves broadly by metadata, filters unavailable/unauthorized/stale candidates, then composes the smallest set and adds verification plus recovery where needed.

CORE CONTRACTS / COMPONENTS

- Router output: selected components + scores + exclusions + rationale + fallbacks
- Tool availability is observed at runtime
- Historical success cannot be self-reported; it comes from verified traces
- Router regret compares expected vs realized utility after completion

> **Engineering implication.** Routing should minimize total composition while preserving verification and recovery.

الشرح الهندسي

الموجه يحول المهمة المطبعة إلى أقل تركيب متوافق من المهارات والقدرات والوكلاء والأدوات والتحقق. دالة التوجيه تدمج الصلة والمنفعة والتوافق وتوفر الدليل والنجاح التاريخي مقابل الكلفة والخطر وكلفة السياق والتكرار. يجلب مرشحين ثم يستبعد غير المتاح أو غير المأذون أو القديم، وبعدها يضيف التحقق والتعافي.

العقود والعناصر الأساسية

- مخرج Router: العناصر والدرجات والمستبعدات والتعليل والبدائل
- توفر الأداة يُرصد وقت التشغيل
- النجاح التاريخي لا يكون بتقرير ذاتي بل من الآثار المتحققة
- Router Regret يقارن المنفعة المتوقعة بالمتحققة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Tool Contracts & Framework Adapters

عقود الأدوات ومحولات الأطر

ENGINEERING DEFINITION

Tools are treated as typed external capabilities with explicit side-effect classes. A tool contract specifies name/version, operation, input schema, output schema, authorization tier, idempotency, reversibility, timeout, retry policy, evidence returned, sensitive fields, and verification method. Framework adapters translate the same logical tool/skill contract to Hermes, ChatGPT connectors, coding runtimes, shell/container actions, browser agents, or future environments without changing the core mission semantics.

CORE CONTRACTS / COMPONENTS

- Tool call ≠ proof; post-condition inspection is separate
- Adapter must expose unsupported capability rather than simulate it
- Secret material stays in connector/runtime vault boundaries
- Retry only when operation semantics are safe

> **Engineering implication.** Adapters isolate host-specific syntax from host-independent capability semantics.

الشرح الهندسي

الأداة قدرة خارجية نمطية ذات فئة أثر. عقد الأداة يحدد الاسم والإصدار والعملية والمدخل والمخرج والتفويض وIdempotency والتراجع والمهلة وإعادة المحاولة ونوع الدليل والحقول الحساسة وطريقة التحقق. Framework Adapter يترجم العقد إلى Hermes أو ChatGPT أو بيئة برمجة أو متصفح دون تغيير دلالة المهمة.

العقود والعناصر الأساسية

- استدعاء الأداة لا يساوي الدليل؛ فحص ما بعد الشرط مستقل
- المحول يعلن عدم الدعم ولا يحاكي قدرة غير موجودة
- الأسرار تبقى في حدود خزنة الموصل/البيئة
- إعادة المحاولة فقط عندما تسمح دلالة العملية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Mission Graph & Plan DAG

رسم المهمة وPlan DAG

ENGINEERING DEFINITION

Planning is represented as a directed acyclic graph when possible. Nodes encode observations, retrievals, transformations, tool actions, tests, approvals, and writebacks; edges encode dependencies and evidence flow. Each node has preconditions, expected world effect, authorization tier, cost/risk estimate, success predicate, failure handler, and rollback reference. DAG planning supports concurrency for independent read-only branches while serializing conflicting or mutation-heavy operations.

CORE CONTRACTS / COMPONENTS

- Node statuses: PENDING/READY/RUNNING/ VERIFIED/FAILED/SKIPPED/BLOCKED
- Critical path is value- and risk-aware, not only time-aware
- Evidence produced by one node unlocks dependent nodes
- Plan can be recompiled when observed state invalidates assumptions

> **Engineering implication.** Plan DAGs make dependencies, approvals, concurrency, and rollback explicit.

الشرح الهندسي

تُمثل الخطة كرسم DAG: عقد للرصد والاسترجاع والتحويل واستدعاءات الأدوات والاختبارات والموافقات وإعادة الكتابة، وحواف للاعتماد وتدفق الدليل. لكل عقدة شروط سابقة وأثر متوقع وتفويض وكلفة/خطر ومعيار نجاح ومعالج فشل ومرجع تراجع. تسمح البنية بالتوازي للقراءات المستقلة وتفرض التسلسل للتعديلات المتعارضة.

العقود والعناصر الأساسية

- حالات العقد: PENDING/READY/RUNNING/ VERIFIED/FAILED/SKIPPED/BLOCKED
- المسار الحرج يراعي القيمة والخطر لا الزمن فقط
- دليل عقدة يفتح العقد التابعة
- يمكن إعادة ترجمة الخطة عندما تكسر الحالة المرصودة الافتراضات

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Executor Loop

حلقة المنفذ



ENGINEERING DEFINITION

The executor follows a disciplined sequence: INTERPRET → DISCOVER CURRENT STATE → BUILD CONTEXT PACKET → RUN GAP COURT → ROUTE SKILLS → BUILD PLAN DAG → VALIDATE PRECONDITIONS → CREATE CHECKPOINT → EXECUTE BOUNDED STEPS → INSPECT SIDE EFFECTS → TEST → CRITIQUE → REPAIR OR ROLLBACK → REPORT → UPDATE WORLD MODEL → EXTRACT LEARNING CANDIDATES. The executor starts read-only, uses atomic writes when possible, keeps a retry budget, and never converts partial completion into verified success.

CORE CONTRACTS / COMPONENTS

- Execution trace records tool, input class, output summary, affected objects, checks, authorization
- Repeated failure triggers stop/repair, not infinite retries
- Checkpoint precedes mutation
- Final status is VERIFIED_SUCCESS / PARTIAL / BLOCKED / ROLLED_BACK

> **Engineering implication.** The executor treats verification as an execution stage, not a reporting afterthought.

الشرح الهندسي

المنفذ يمر بتسلسل من تفسير الهدف ثم اكتشاف الحالة وبناء حزمة السياق ومحكمة الفجوات والتوجيه وPlan DAG وفحص الشروط ونقطة الحفظ والتنفيذ المحدود وفحص الآثار والاختبار والنقد والإصلاح/التراجع والتقرير وتحديث العالم واستخراج التعلم. البداية قراءة فقط والكتابات ذرية قدر الإمكان.

العقود والعناصر الأساسية

- الأثر يسجل الأداة ونوع المدخل والملخص والعناصر المتأثرة والفحوص والتفويض
- الفشل المتكرر يوقف أو يصلح ولا يعيد بلا نهاية
- نقطة الحفظ قبل التعديل
- الحالة النهائية محددة وصريحة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Courts: Gap, Falsification & Demolition

المحاكم: الفجوة والتفنيد والهدم

ENGINEERING DEFINITION

Courts are adversarial validation modules, not personas for style. The Gap Court checks definitions, boundaries, evidence, mechanisms, measurements, implementation, validation, operations, security, business value, communication, maintenance, dependencies, rollback, governance, ethics, and human consequences. The wider demolition family includes Loss, Duplication, Misclassification, Temporal, Provenance, Retrieval, Integrity, Security, Continuity, Future-Ingestion, Theory Demolition, and World-Effect courts.

CORE CONTRACTS / COMPONENTS

- Court verdict: PASS / REPAIR_REQUIRED / EVIDENCE_REQUIRED / BLOCK
- Court output must name the missing layer and repair action
- Courts cannot invent evidence; they can only demand or route retrieval/tests
- Critical court failures block promotion/ deployment

> **Engineering implication.** Courts identify missing structure and route repair; they do not manufacture certainty.

الشرح الهندسي

المحاكم وحدات تحقق خصومية لا شخصيات أدبية. محكمة الفجوات تفحص التعريف والحد والدليل والآلية والقياس والتنفيذ والتحقق والتشغيل والأمن والقيمة والتواصل والصيانة والاعتمادات والتراجع والحوكمة والأخلاق. توجد محاكم للخسارة والتكرار والتصنيف والزمن والمصدر والاسترجاع والسلامة والاستمرارية والنظرية والأثر الخارجي.

العقود والعناصر الأساسية

- حكم المحكمة: PASS / REPAIR_REQUIRED / EVIDENCE_REQUIRED / BLOCK
- المخرج يحدد الطبقة المفقودة وإجراء الإصلاح
- المحكمة لا تخترع الدليل بل تطلب استرجاعا أو اختبارا
- الفشل الحرج يحجب الترقية أو النشر

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Evidence Hierarchy & Claim Contract

هرمية الدليل وعقد الادعاء

ENGINEERING DEFINITION

Important claims are scoped by a claim contract: source, date, confidence, project boundary, assumptions, mediators, moderators, thresholds, stop conditions, counter-evidence, and validation status. Evidence categories distinguish established result, direct source fact, explicit user instruction, deployed configuration, measured observation, reproducible experiment, external research, generated synthesis, inference, interpretation, hypothesis, conjecture, and speculative extension. Generated summaries never silently outrank direct evidence.

CORE CONTRACTS / COMPONENTS

- Claim graph edge: CLAIM --supported_by/contradicted_by--> EVIDENCE
- Confidence updates are evidence-event driven
- Temporal validity is explicit for volatile claims
- Speculative constructs use labels such as S1/H1/X rather than implicit certainty

> **Engineering implication.** Claims remain scoped to the environment and time in which their evidence was obtained.

الشرح الهندسي

كل ادعاء مهم يحمل عقدا: المصدر والتاريخ والثقة وحدود المشروع والافتراضات والوسطاء والمعدلات والعتبات وشروط التوقف والدليل المضاد وحالة التحقق. تُفصل الحقائق المباشرة والتعليمات والتكوين المنشور والقياس والتجربة والبحث عن التوليف والاستنتاج والفرضية والتكهن. الملخص المولد لا يتفوق على المصدر المباشر.

العقود والعناصر الأساسية

- حافة الرسم: CLAIM ← supported_by/contradicted_by ← EVIDENCE
- تحديث الثقة يحدث بسبب حدث دليل
- الصلاحية الزمنية صريحة للادعاءات المتغيرة
- التراكيب التجريبية تحمل S1/H1/X بدل يقين ضمني

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Verification & World-Effect Loop

التحقق وحلقة الأثر الخارجي



ENGINEERING DEFINITION

The World-Effect Loop distinguishes proposal → attempted action → verified action → external result → measured effect. This prevents the common agentic error of equating a plan, a tool response, or a created file with the intended real-world outcome. Each mission defines observable postconditions before action. After mutation, the verifier re-reads the authoritative environment, checks invariants, compares intended versus actual state, and records residual gaps.

CORE CONTRACTS / COMPONENTS

- Verification sources should be independent of the action call where possible
- Success predicate is defined before execution
- Side effects are inspected for unintended changes
- World-effect metric can be binary, numeric, or structured evidence

> **Engineering implication.** World-effect closure requires observed postconditions in the authoritative environment.

الشرح الهندسي

حلقة الأثر تميز المقترح ثم محاولة الفعل ثم الفعل المتحقق ثم النتيجة الخارجية ثم الأثر المقاس. بذلك لا يساوي التخطيط أو جواب أداة أو إنشاء ملف النتيجة المطلوبة. قبل التنفيذ تُعرّف Postconditions قابلة للرصد وبعده يُعاد فحص البيئة الموثوقة ومقارنة الحالة الفعلية بالمطلوبة.

العقود والعناصر الأساسية

- يفضل أن يكون مصدر التحقق مستقلا عن استدعاء الفعل
- معيار النجاح يُعرّف قبل التنفيذ
- تفحص الآثار الجانبية غير المقصودة
- مقياس الأثر قد يكون ثنائيا أو رقميا أو دليلا منظما

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Self-Reconstruction Memory

ذاكرة إعادة البناء الذاتي

ENGINEERING DEFINITION

Memory stores the minimal trajectory needed to improve future runs: previous state, active difference, chosen path, important rejected paths, evidence, failure, court verdict, external effect, changed state, residual uncertainty, and next governing question. The goal is reconstructability, not diary-like accumulation. Indexes should be rebuildable from surviving project-level state and manifests, so a new model or runtime can resume without relying on opaque conversational memory.

CORE CONTRACTS / COMPONENTS

- Writeback classes: state delta, decision, evidence, artifact link, failure lesson, skill candidate
- Do not persist hidden chain-of-thought
- Memory compaction preserves causal links and provenance
- Recovery test: can a fresh agent infer next action from canonical state?

> **Engineering implication.** A memory system is successful when a new agent can reconstruct the next action.

الشرح الهندسي

الذاكرة تحفظ الحد الأدنى من المسار الذي يمكّن التحسن: الحالة السابقة والفرق والمسار المختار والبدائل المهمة والدليل والفشل وحكم المحكمة والأثر والحالة الجديدة وعدم اليقين والسؤال التالي. الهدف قابلية إعادة البناء لا تراكم يوميات. يجب أن يستطيع Runtime جديد إعادة الفهارس والمتابعة من الأصول والسجلات.

العقود والعناصر الأساسية

- أنواع الكتابة: فرق حالة وقرار ودليل ورابط أثر ودرس فشل ومرشح مهارة
- لا تُخزن سلسلة التفكير السرية
- الضغط يحفظ السببية والمصدر
- اختبار التعافي: هل وكيل جديد يعرف الخطوة التالية من الحالة المعيارية؟

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

HyperSkill Factory & Capability Extraction

مصنع HyperSkill واستخراج القدرات

ENGINEERING DEFINITION

The HyperSkill Factory converts repeated successful procedures into reusable skills. A candidate is mined from traces only after a verified run, then normalized into trigger/input/output/precondition/procedure/tool/validation/failure schemas. It is attacked by courts, benchmarked against the baseline procedure, and either rejected, revised, or registered. Repeated skill compositions may be promoted to HyperAbilities; stable bounded workflows may become HyperLoops.

CORE CONTRACTS / COMPONENTS

- Extraction source must be a verified trace
- Skill candidate cannot self-promote
- Deduplicate by purpose + procedure + effect, not name alone
- Registry update includes version, provenance, tests, historical result

> **Engineering implication.** Capability learning is trace-derived and regression-gated.

الشرح الهندسي

مصنع المهارات يحول الإجراءات الناجحة المتكررة إلى HyperSkills. المرشح يُستخرج من أثر تشغيل متحقق ثم يُطبّع إلى عقد واضح ويمر بالمحاكم والاختبارات ومقارنة Baseline. قد تتطور التراكيب المتكررة إلى HyperAbility، وسير العمل المستقر المحدود إلى HyperLoop.

العقود والعناصر الأساسية

- مصدر الاستخراج أثر متحقق
- مرشح المهارة لا يرقّي نفسه
- إزالة التكرار حسب الهدف والإجراء والأثر لا الاسم فقط
- تحديث السجل يشمل الإصدار والمصدر والاختبارات والنتيجة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Evolution Laboratory

مختبر التطور

ENGINEERING DEFINITION

The Evolution Lab is the controlled experimentation layer for prompts, skills, router weights, adapters, and bounded model improvements. It generates adversarial and long-horizon tasks, replays real failures without removing causal conditions, mutates capability specifications, builds cleaned trace datasets, and benchmarks candidates against frozen baselines. Promotion requires no critical regression, reproducibility, canary operation, and automatic rollback. The lab cannot self-promote a candidate into production.

CORE CONTRACTS / COMPONENTS

- Critical dimensions: authorization, security, truthfulness, intent preservation, rollback, integrity, value
- Benchmark stores baseline/candidate versions and environment fingerprints
- Canary radius expands only after verified stability
- Learning candidates remain isolated until promotion decision

> **Engineering implication.** Evolution is isolated from production until evidence supports promotion.

الشرح الهندسي

مختبر التطور يجرب بصورة محكومة على الpromptsوالمهارات وأوزان Router والمحولات والتحسينات المحدودة. يبني مهام خصومية، يعيد أخطاء حقيقية مع أسبابها، يعدّل العقود، ينظف آثار التنفيذ ويختبر المرشحين مقابل Baseline ثابت. الترقية تحتاج غياب Regression حرج وإعادة إنتاج وCanary وتراجع تلقائي، ولا يستطيع المختبر ترقية نفسه.

العقود والعناصر الأساسية

- الأبعاد الحرجة: التفويض والأمن والصدق وحفظ النية والتراجع والسلامة والقيمة
- الاختبار يحفظ إصدارات baseline/candidate وبصمة البيئة
- نطاق Canary يتوسع بعد الثبات المتحقق
- مرشحو التعلم يبقون معزولين حتى قرار الترقية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Boot Sequence, Integrity & Self-Test

تسلسل الإقلاع والسلامة والاختبار الذاتي

ENGINEERING DEFINITION

A cold boot loads the deployment kernel, the four MOUSSAIF1B OS modules in order, parses the skill registry, verifies unique IDs/names and required fields, loads state/freshness policies, discovers available tools, resolves authorization context, and runs read-only self-tests. The system enters READY, DEGRADED_SAFE_MODE, or BLOCKED. Degraded mode permits inspection, reasoning, simulation, and repair drafting but not privileged side effects.

CORE CONTRACTS / COMPONENTS

- Self-test: constitutional precedence active
- Self-test: privileged action rejected without T3
- Self-test: stale state detected
- Self-test: router selects minimal skill set
- Self-test: missing tools downgrade instead of faking
- Self-test: simulated failure triggers bounded repair/rollback

> **Engineering implication.** Boot integrity turns missing components into degraded mode rather than silent partial initialization.

الشرح الهندسي

الإقلاع البارد يحمل Kernel ثم OS1–OS4 ثم سجل المهارات ويتحقق من المعرفات والحقول ويحمّل سياسات الحالة والحداثة ويكتشف الأدوات والتفويض ثم يجري Self-Test قراءة فقط. النتيجة READY أو DEGRADED_SAFE_MODE أو BLOCKED. الوضع المتدهور يسمح بالفحص والمحاكاة والإصلاح دون آثار حساسة.

العقود والعناصر الأساسية

- اختبار: أسبقية الدستور فعالة
- اختبار: رفض الفعل الحساس دون T3
- اختبار: اكتشاف الحالة القديمة
- اختبار: Router يختار أقل مجموعة مهارات
- اختبار: الأداة المفقودة تسبب Downgrade لا تلفيق
- اختبار: الفشل المحاكى يؤدي لإصلاح/تراجع محدود

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Runtime Discovery & Environment Fingerprinting

اكتشاف وقت التشغيل وبصمة البيئة

ENGINEERING DEFINITION

Before installation or mutation, MOUSSAIXT discovers the actual runtime: executable/version, active profile, configuration paths, skill directories, tool interfaces, model endpoints, serving engines, operating system, shell, Python/container availability, GPU/VRAM, approved project roots, and connector capabilities. It does not invent configuration keys. Environment fingerprints are attached to execution traces so results can be reproduced or explained when behavior differs across machines.

CORE CONTRACTS / COMPONENTS

- Discovery is read-only by default
- Unsupported registration yields an installation plan, not fabricated success
- Configuration changes are backed up and diffed
- Runtime features are capability flags consumed by the router

> **Engineering implication.** Environment discovery prevents configuration hallucination and improves reproducibility.

الشرح الهندسي

قبل التثبيت أو التعديل يكتشف MOUSSAIXT البيئة الحقيقية: الإصدار والملف النشط ومسارات الإعداد والمهارات وواجهات الأدوات والنماذج ومحركات الخدمة والنظام والShell وPython والحاويات وGPU/VRAM وجذور المشاريع والموصلات. لا يخترع مفاتيح إعداد. تُربط بصمة البيئة بآثار التنفيذ لإعادة الإنتاج.

العقود والعناصر الأساسية

- الاكتشاف قراءة فقط افتراضيا
- عدم دعم التسجيل ينتج خطة تثبيت لا نجاحا مخترعا
- تغييرات الإعداد تحفظ وتُعرض Diff
- خصائص البيئة أعلام قدرة يستهلكها Router

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

MOUSSAIXT Cross-Environment Execution Shell

MOUSSAIXT كغلاف تنفيذ عابر للبيئات

ENGINEERING DEFINITION

MOUSSAIXT is specified as the portability layer that keeps mission semantics stable while execution moves among ChatGPT tool runtimes, Hermes local agents, coding/terminal environments, browser agents, cloud APIs, and future hosts. It owns adapter discovery, capability negotiation, context packaging, state synchronization, and evidence normalization. Core concepts such as HyperSkill IDs, world-model entities, claim contracts, and mission statuses should remain host-independent.

CORE CONTRACTS / COMPONENTS

- Host adapter exposes: read/write/tool/schedule/ browser/model/file capabilities
- Canonical state lives outside transient host context
- Normalization converts host-specific responses into common evidence objects
- Conflict resolver chooses observed newest authoritative state

> **Engineering implication.** MOUSSAIXT is the host portability boundary for the architecture.

الشرح الهندسي

يُعرّف MOUSSAIXT كطبقة قابلية نقل تحافظ على دلالة المهمة عند الانتقال بين ChatGPT وHermes وبيئات البرمجة والمتصفح والAPIs. تدير اكتشاف المحولات وتفاوض القدرات وحزم السياق ومزامنة الحالة وتوحيد الدليل. معرفات المهارات وكيانات العالم وعقود الادعاء وحالات المهمة يجب أن تبقى مستقلة عن المضيف.

العقود والعناصر الأساسية

- المحول يكشف قدرات القراءة/الكتابة/الأداة/الجدولة/المتصفح/النموذج/ الملفات
- الحالة المعيارية خارج سياق المضيف المؤقت
- Normalization يحول الردود إلى كائنات دليل مشتركة
- حل التعارض يختار أحدث حالة موثوقة مرصودة

> ملاحظة: MOUSSAIXT يُعامل كغلاف تشغيل وتوافق بين البيئات، بينما تظل النواة المعرفية مستقلة عن المضيف.

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Google Drive Persistent Substrate

Google Drive كطبقة دائمة

ENGINEERING DEFINITION

The Drive substrate functions as persistent external memory and artifact storage, not as the reasoning engine itself. The canonical control plane points to context maps, current-state manifests, KB cores, skill registries, agent registries, task/runtime manifests, project assets, and provenance. Drive operations should preserve organization, use canonical hubs rather than uncontrolled root-folder proliferation, and distinguish raw sources, derived artifacts, registries, and archives.

CORE CONTRACTS / COMPONENTS

- Storage states are verified by readback
- Canonical pointer registries prevent duplicate “latest” files
- Large corpora are indexed; fast boot loads summaries/pointers only
- Shared/mirrored files retain provenance and ownership metadata

> **Engineering implication.** Drive is canonical storage only where manifests and readback establish canonicality.

الشرح الهندسي

Google Drive يعمل كذاكرة خارجية دائمة ومخزن للأصول لا كمحرك التفكير نفسه. Control Plane يشير إلى خرائط السياق والحالة الحالية وKB وسجلات المهارات والوكلاء والمهام والأصول والمصدر. يجب الحفاظ على بنية معيارية وتجنب تضخم مجلدات الجذر وفصل RAW عن DERIVED وعن السجلات والأرشيف.

العقود والعناصر الأساسية

- حالة التخزين تُثبت بالقراءة اللاحقة
- سجلات المؤشرات تمنع تعدد ملفات latest
- الكوربس الكبير مفهرس والإقلاع يحمل الملخصات والمؤشرات
- الملفات المشتركة/المرايا تحتفظ بالمصدر والملكية

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Observability, Trace & Audit Plane

الرصد والأثر والتدقيق

ENGINEERING DEFINITION

Observability spans cognitive routing and real tool execution. Each mission records objective, observed state, assumptions, evidence grades, selected skills, router scores, plan nodes, tool calls, affected artifacts, tests, failures/repairs, world effects, unresolved risks, rollback, and learning candidates. Sensitive values are redacted. Traces support debugging, audit, skill mining, regression testing, and router-regret analysis.

CORE CONTRACTS / COMPONENTS

- Trace IDs link mission → plan → tool call → evidence → state delta
- Metrics: task success, verification coverage, rollback success, latency, cost, context
- Logs are append-oriented; summaries are derived projections
- Audit can reproduce why a status became VERIFIED_SUCCESS

> **Engineering implication.** Observability data doubles as the training substrate for future skill improvements.

الشرح الهندسي

الرصد يغطي التوجيه المعرفي والتنفيذ الفعلي. كل مهمة تسجل الهدف والحالة والافتراضات والدليل والمهارات والدرجات والخطة والأدوات والأصول المتأثرة والاختبارات والفشل والإصلاح والأثر والمخاطر والتراجع والتعلم. تُحجب القيم الحساسة، وتستخدم الآثار للتدقيق والتصحيح واستخراج المهارات والـRegression.

العقود والعناصر الأساسية

- معرف الأثر يربط المهمة بالخطة والأداة والدليل وفرق الحالة
- المقاييس: نجاح المهمة وتغطية التحقق والتراجع والزمن والكلفة والسياق
- السجل Append-oriented والملخصات إسقاطات مشتقة
- التدقيق يعيد سبب الوصول إلى VERIFIED_SUCCESS

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Security, Privacy & Integrity Architecture

معمارية الأمن والخصوصية والسلامة

ENGINEERING DEFINITION

Security is enforced through least privilege, authorization tiers, sensitivity labels, secret references, reversible mutation, auditability, and isolation of experimental capabilities. Integrity protections include hashes/manifests, unique IDs, canonical/latest registries, anti-duplication checks, stale-state detection, provenance, and safe degraded mode. Privacy-sensitive reads or messaging are treated as privileged effects even when technically available.

CORE CONTRACTS / COMPONENTS

- Threats: secret leakage, stale state, unauthorized side effects, prompt/data poisoning, duplicate canonical state
- Controls: allowlists, provenance checks, checkpoints, diff review, postcondition verification
- T4 actions remain prohibited regardless of optimization value

> **Engineering implication.** Security decisions are embedded in routing and execution, not bolted on afterward.

الشرح الهندسي

الأمن يعتمد أقل صلاحية ومستويات التفويض وتسميات الحساسية ومراجع الأسرار وقابلية التراجع والتدقيق وعزل التجارب. سلامة البيانات تعتمد الهاش والمانيفست والمعرفات الفريدة وسجلات latest ومنع التكرار واكتشاف القدم والمصدر والوضع الآمن. الوصول الحساس أو الرسائل آثار مميزة حتى لو كانت الأداة قادرة عليها.

العقود والعناصر الأساسية

- التهديدات: تسريب الأسرار والحالة القديمة والأثر غير المأذون وتسميم البيانات وتعدد الحالة المعيارية
- الضوابط: Allowlist ومصدر ونقاط حفظ ومراجعة Diff والتحقق بعد الفعل
- أفعال T4 محظورة مهما كانت منفعتها

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Data & Artifact Lifecycle

دورة حياة البيانات والأصول

ENGINEERING DEFINITION

Artifacts move through source, derived, verified, registered, operational, superseded, and archived states. Raw sources remain immutable where possible; derived outputs carry lineage links. The anti-duplication covenant requires lineage scanning before creating new kernels: if a stronger artifact already exists, produce a delta, patch, extension, or deployment derivative. Canonical naming includes project, artifact family, version, scope, method, and date when useful.

CORE CONTRACTS / COMPONENTS

- Artifact metadata: id, title, type, project, version, canonical_status, source_ids, hash, created_at
- Supersession never deletes lineage
- Derived artifacts are regenerated from source-of-truth when possible
- Archive readiness checks classification, provenance, retrieval, integrity, dedup, versioning, backup

> **Engineering implication.** Artifact lineage is a graph of derivation and supersession, not a folder naming convention alone.

الشرح الهندسي

تمر الأصول من SOURCE إلى DERIVED ثم VERIFIED وREGISTERED وOPERATIONAL ثم SUPERSEDED/ARCHIVED. المصادر الخام تبقى ثابتة قدر الإمكان والمشتقات تحمل روابط السلالة. قبل إنشاء Kernel جديد يجب فحص الموجود؛ إن وجد أصل أقوى يُنشأ Delta/Patch/Extension لا نسخة مموهة. الاسم المعياري يكشف المشروع والعائلة والإصدار والنطاق والمنهج والتاريخ.

العقود والعناصر الأساسية

- ميتاداتا: معرف وعنوان ونوع ومشروع وإصدار وحالة معيارية ومصادر وHash وتاريخ
- الاستبدال لا يمحو السلالة
- المشتقات تُعاد من Source of Truth عند الإمكان
- جاهزية الأرشيف تفحص التصنيف والمصدر والاسترجاع والسلامة والتكرار والإصدارات والنسخ

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Failure Recovery, Repair & Rollback

التعافي والإصلاح والتراجع

ENGINEERING DEFINITION

Failure is modeled as state, not an exception to be hidden. Tool errors, failed tests, contradiction, insufficient evidence, stale context, permission denial, and verification mismatch are classified and routed. The system uses bounded retries only when safe, then chooses repair, alternate tool/agent, downgrade to dry run, partial completion, block, or rollback. Rollback restores both external state and canonical world-model projections where appropriate.

CORE CONTRACTS / COMPONENTS

- Failure object: category, node, evidence, retry_count, blast_radius, repair_options, rollback_ref
- No-progress and repeated-identical-error detectors stop loops
- Rollback verification is mandatory
- Failure lessons may become skill candidates only after causal analysis

> **Engineering implication.** Recovery is considered complete only after rollback postconditions are verified.

الشرح الهندسي

الفشل حالة صريحة لا خطأ يُخفى. أخطاء الأدوات والاختبارات والتعارض وضعف الدليل وقدم السياق ورفض الصلاحية وفشل التحقق تُصنف وتُوجه. إعادة المحاولة محدودة، ثم إصلاح أو أداة بديلة أو Dry Run أو Partial أو Block أو Rollback. التراجع يعيد الحالة الخارجية وتوقعات نموذج العالم.

العقود والعناصر الأساسية

- كائن الفشل: الفئة والعقدة والدليل وعدد المحاولات ونطاق الضرر وخيارات الإصلاح ومرجع التراجع
- كاشف عدم التقدم يوقف الدوران
- التحقق من نجاح التراجع إلزامي
- درس الفشل يصبح مهارة بعد تحليل السبب فقط

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Performance, Token Economy & Scale

الأداء واقتصاد التوكن والتوسع

ENGINEERING DEFINITION

MOUSSAIF scales by sparse activation rather than indiscriminate context growth. Compact control manifests, multi-resolution ROFSC summaries, lazy HyperSkill loading, graph-based retrieval, branch budgets, and compute routing reduce token and tool waste. Performance is judged by verified useful work per context token, tool call, latency unit, and monetary cost. Large knowledge corpora remain external and are pulled only as required by the active difference.

CORE CONTRACTS / COMPONENTS

- Budgets: context, tool calls, latency, monetary cost, retries, agents
- Load metadata first; bodies only for selected nodes
- Cache stable ontology, invalidate volatile operational state
- Parallelize independent reads; serialize mutations sharing state

> **Engineering implication.** Scale is achieved by selective activation and externalized memory, not giant prompts.

الشرح الهندسي

يتوسع النظام بالتفعيل المتناثر لا بتضخم السياق. المانيفست المضغوط وROFSC متعدد الدقة والتحميل الكسول للمهارات والاسترجاع الرسومي وميزانيات الفروع والتوجيه تقلل الهدر. الأداء يقاس بالعمل المفيد المتحقق لكل توكن واستدعاء أداة وزمن وكلفة، بينما تبقى الكوربس الكبيرة خارج السياق.

العقود والعناصر الأساسية

- الميزانيات: السياق والأدوات والزمن والكلفة والمحاولات والوكلاء
- حمّل الميتاداتا أولا ثم أجسام العناصر المختارة
- خزن الأنطولوجيا الثابتة وأبطل الحالة المتغيرة
- وازِ القراءات المستقلة وسلسل التعديلات المشتركة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

M-TAOS: Task HyperKnowledge OS

M-TAOS: نظام معرفة المهام



ENGINEERING DEFINITION

M-TAOS is the task-level operating layer that converts recurring objectives into structured task OS units. A task OS combines governing objective, knowledge context, skill/ability set, agent roles, scheduler/event trigger, tool contracts, evidence targets, output registry, and state writeback. It allows cron-like or scheduled activity to remain synchronized with the same canonical knowledge and safety architecture instead of becoming isolated prompts.

CORE CONTRACTS / COMPONENTS

- Task manifest: task_id, objective, cadence/ trigger, context pointers, capabilities, approvals, outputs, KPIs
- Runs are incremental: detect delta since last successful state
- Shared lead/research/content hubs prevent siloed task memory
- Scheduler success is verified separately from task logic

> **Engineering implication.** Task OS units synchronize recurring work with the same canonical state as interactive missions.

الشرح الهندسي

M-TAOS هو مستوى تشغيل المهام المتكررة. تجمع وحدة Task OS الهدف الحاكم والسياق المعرفي والمهارات/القدرات والوكلاء والمشغل الزمني أو الحدث وعقود الأدوات وأهداف الدليل وسجل المخرجات وإعادة كتابة الحالة. هكذا لا تصبح المهام المجدولة Prompts معزولة بل جزءا من نفس المعرفة والسلامة.

العقود والعناصر الأساسية

- مانيفست المهمة: المعرف والهدف والمشغل ومؤشرات السياق والقدرات والموافقات والمخرجات والمقاييس
- التشغيل تفاضلي: فرق منذ آخر حالة ناجحة
- المحاور المشتركة تمنع ذاكرة منعزلة لكل مهمة
- نجاح الجدولة يُتحقق منه منفصلا عن منطق المهمة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Research & Theory Construction Kernel

نواة البحث وبناء النظريات

ENGINEERING DEFINITION

The research layer treats theoretical constructs as falsifiable programs rather than branding. The pipeline is gap detection → named contribution → definitions → formal objects/equations → mechanism → derived principles/tools → metrics → hypotheses/ predictions → adversarial counter-models → demolition court → empirical/implementation program → limitations → lineage update. New theory names must prove a non-redundant gap against the existing State of the Art and the internal MOUSSAIF lineage.

CORE CONTRACTS / COMPONENTS

- Scientific standard: observation → formalization → hypothesis → prediction → test → replication → revision
- Cross-model reasoning preserves unresolved disagreement
- Speculative extensions are explicitly tagged
- Research outputs can compile into operational skills only after mechanism/test definition

> **Engineering implication.** Research constructs remain hypotheses until formalization and tests support stronger status.

الشرح الهندسي

طبقة البحث تعامل النظريات كبرامج قابلة للتفنيد لا كعلامات. المسار: كشف الفجوة ثم المساهمة والتعريف والأجسام/المعادلات والآلية والمبادئ والأدوات والمقاييس والفرضيات والتنبؤات والنماذج المضادة ومحكمة الهدم وبرنامج التجربة/التنفيذ والحدود ثم تحديث السلالة. الاسم الجديد يحتاج فجوة غير مكررة.

العقود والعناصر الأساسية

- المعيار العلمي: ملاحظة ← صياغة ← فرضية ← تنبؤ ← اختبار ← تكرار ← مراجعة
- التفكير متعدد النماذج يحفظ الخلاف غير المحسوم
- الامتدادات التكهنية موسومة صراحة
- الناتج البحثي يصبح Skill بعد تعريف الآلية والاختبار

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Domain Adapters: Research, Brands, Career & Operations

محولات المجالات: البحث والبراند والوظائف والتشغيل

ENGINEERING DEFINITION

Domain adapters specialize the generic OS without forking the kernel. A research adapter changes evidence and citation requirements; a brand/content adapter changes audience, channel, creative, and conversion metrics; a career adapter changes lead/job schemas, fit scoring, outreach authorization, and follow-up state; an operations adapter emphasizes system state, incidents, deployments, and SLAs. All still use the same question/difference/routing/evidence/world-effect/memory contracts.

CORE CONTRACTS / COMPONENTS

- Adapter may add schemas and metrics but not weaken constitutional rules
- Cross-domain knowledge is linked through shared concepts/skills
- Domain-specific tools are capability flags
- Outputs register back into project world model

> **Engineering implication.** Domain adapters specialize metrics and schemas while keeping core invariants unchanged.

الشرح الهندسي

محول المجال يخصص النظام دون نسخ Kernel. البحث يغير معايير الدليل والمراجع، البراند يغير الجمهور والقنوات والإبداع والتحويل، الوظائف تغير مخطط الفرص وFit والتواصل والمتابعة، والتشغيل يركز الحالة والحوادث والنشر وSLA. لكن الجميع يظل تحت عقود السؤال والفرق والتوجيه والدليل والأثر والذاكرة.

العقود والعناصر الأساسية

- المحول يضيف مخططات ومقاييس ولا يضعف الدستور
- المعرفة العابرة للمجالات ترتبط بالمفاهيم/المهارات المشتركة
- أدوات المجال أعلام قدرة
- المخرجات تعود إلى نموذج عالم المشروع

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Reference Interfaces & Machine-Readable Contracts

الواجهات المرجعية والعقود القابلة للقراءة الآلية

ENGINEERING DEFINITION

Machine-readable contracts make the architecture portable. The core mission output includes mission, project_scope, authorization, observed_state, assumptions, evidence_grades, gap_court, selected_skills, router_scores, plan, actions_completed, verification, world_effect, world_model_updates, failures_and_repairs, unresolved_risks, rollback, learning_candidates, and status. Component schemas should be JSON/YAML compatible and versioned independently from prose documentation.

CORE CONTRACTS / COMPONENTS

- MissionStatus enum: VERIFIED_SUCCESS, PARTIAL, BLOCKED, ROLLED_BACK
- EvidenceGrade enum: observed, verified_external, supported, inferred, hypothesized, speculative, unknown
- All IDs are globally unique within their registry namespace
- Schema migration preserves backward-readable history

```
status: VERIFIED_SUCCESS | PARTIAL | BLOCKED |
ROLLED_BACK
evidence: observed | verified_external |
supported | inferred | hypothesized |
speculative | unknown
```

> **Engineering implication.** Machine-readable schemas are the interoperability backbone of MOUSSAIXT.

الشرح الهندسي

العقود الآلية تجعل المعمارية قابلة للنقل. مخرج المهمة يشمل الهدف والنطاق والتفويض والحالة والافتراضات والدليل والمحكمة والمهارات والدرجات والخطة والأفعال والتحقق والأثر وتحديث العالم والفشل والإصلاح والمخاطر والتراجع والتعلم والحالة. المخططات JSON/YAML ومؤرخة مستقلة عن الشرح.

العقود والعناصر الأساسية

- MissionStatus: VERIFIED_SUCCESS / PARTIAL / BLOCKED / ROLLED_BACK
- EvidenceGrade: observed / verified_external / supported / inferred / hypothesized / speculative / unknown
- المعرفات فريدة داخل Namespace السجل
- هجرة المخطط تحفظ التاريخ قابلا للقراءة

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Maturity Model, Open Gaps & Engineering Roadmap

نموذج النضج والفجوات وخارطة الطريق

ENGINEERING DEFINITION

The current architecture is strongest as a documented AGI-oriented cognitive/agentic operating system with rich knowledge artifacts, routing doctrine, reusable skills, task/loop concepts, and persistent external memory. The principal engineering gap is runtime unification: many concepts exist in documents and registries but need a single typed implementation that enforces contracts across hosts. HyperAbility needs a canonical registry/schema; version/canonical conflict resolution needs automation; benchmarks need frozen datasets; and world-effect metrics need domain-specific instrumentation.

CORE CONTRACTS / COMPONENTS

- Stage 1: canonical schemas + registry reconciliation
- Stage 2: reference router/executor implementation
- Stage 3: host adapters + state sync tests
- Stage 4: benchmark suite + trace mining
- Stage 5: capability promotion/evolution gates
- Stage 6: domain world-effect instrumentation

> **Engineering implication.** The roadmap priority is executable contract unification across the distributed knowledge corpus.

الشرح الهندسي

المعمارية قوية حاليا كنظام معرفي وكيلي موثق مع KB غنية ومبادئ توجيه ومهارات وحلقات وذاكرة خارجية. الفجوة الكبرى هي توحيد Runtime: بعض المفاهيم ما زالت موزعة بين الوثائق والسجلات وتحتاج تنفيذا نمطيا واحدا. HyperAbility يحتاج سجلا معياريا، وحل تعارض latest يحتاج أتمتة، والBenchmarks تحتاج بيانات مجمدة، ومقاييس الأثر تحتاج Instrumentation حسب المجال.

العقود والعناصر الأساسية

- المرحلة 1: مخططات معيارية وتسوية السجلات
- المرحلة 2: تنفيذ مرجعي Router/Executor
- المرحلة 3: محولات المضيف واختبارات Sync
- المرحلة 4: Benchmark واستخراج الآثار
- المرحلة 5: بوابات ترقية القدرات
- المرحلة 6: قياس الأثر حسب المجال

> أولوية التطوير هي تحويل العقود الموثقة إلى تنفيذ نمطي واحد مع اختبارات تكامل وقياس للأثر.

Architecture invariant: observed state + explicit authorization + evidence + verification outrank model confidence or output volume.

Glossary, Final Architecture Contract & Source Anchors

المعجم والعقد النهائي ومراسي المصادر

ENGINEERING DEFINITION

Final contract: MOUSSAIF AGI OS / MOUSSAIXT is an AGI-oriented engineering architecture whose unit of progress is verified state change. Cognition begins with the governing question and active difference; compute is routed through MCD/QMCD/MCAI and HAKIM-MCD; knowledge is externalized through HKOS and world-model state; reusable agency is compiled into HyperSkills, HyperAbilities, HyperLoops, agents, and tool contracts; plans execute through bounded DAGs; courts and evidence gates challenge claims; world effects are verified; state is reconstructed into persistent memory; and evolution occurs only through benchmarked, reversible promotion.

CORE CONTRACTS / COMPONENTS

- HKOS = HyperKnowledge Operating System
- MCD/QMCD = difference/complexity and question-weighted routing heuristics
- MCAI = candidate comparison + agency integration
- HAKIM-MCD = cognition-to-world-effect pipeline
- HyperSkill = atomic reusable executable knowledge
- HyperAbility = composed capability abstraction
- HyperLoop = bounded iterative controller
- HyperDeployment = verified capability lifecycle
- M-TAOS = task-level knowledge operating system
- MOUSSAIXT = cross-environment execution shell defined in this report

> **Engineering implication.** The system should be evaluated by verified, recoverable value rather than vocabulary density.

الشرح الهندسي

العقد النهائي: وحدة التقدم في MOUSSAIF هي تغير حالة متحقق. يبدأ التفكير بالسؤال الحاكم والفرق، ويوجه الحساب عبر MCD/QMCD/MCAI وHAKIM-MCD، وتُخرج المعرفة في HKOS ونموذج العالم، وتُركب الوكالة في HyperSkills وHyperAbilities وHyperLoops ووكلاء وأدوات، وتنفذ الخطط بشكل محدود، وتفحص المحاكم والدليل، ويُتحقق الأثر ثم تُكتب الذاكرة، ولا يحدث التطور إلا بترقية مختبرة قابلة للتراجع.

العقود والعناصر الأساسية

- HKOS = نظام تشغيل المعرفة الفائقة
- MCD/QMCD = توجيه الفرق/التعقيد الموزون بالسؤال
- MCAI = مقارنة المرشحين وتوزيع الوكالة
- HAKIM-MCD = خط من التفكير إلى الأثر الخارجي
- HyperSkill = معرفة تنفيذية ذرية قابلة لإعادة الاستخدام
- HyperAbility = قدرة مركبة
- HyperLoop = متحكم تكراري محدود
- HyperDeployment = دورة نشر متحققة
- M-TAOS = نظام معرفة على مستوى المهمة
- MOUSSAIXT = غلاف التنفيذ العابر للبيئات كما عُرّف هنا

Source anchors. Primary source anchors in the user corpus: MOUSSAIF1B_ALL_IN_ONE.md (2026-08-08 copy of v1.0 kernel); MOUSSAIF COGNITIVE METHODS REGISTRY — MCD QMCD MCAI HAKIM ROFSC — 2026-08-07; CURRENT — MOUSSAIF AGI Fattah HyperKnowledge OS V3 — Core Ontology — 20MB.md; CURRENT — MOUSSAIF AGI Fattah HyperLoop Runtime OS V3 — 20MB.md; AGI_HyperKnowledge_Content_OS_800p_Agentic_Manual.md; M-TAOS canonical task OS documents. Architecture statements marked as “formalized here” are synthesis/standardization rather than claims that a runtime primitive is already fully implemented.