هندسة الهارنس الوكيلي للذكاء العام الموجّه

من Claude Code وPrime Agent وCodex وHermes إلى AL RAZZI / MOUSSAIFXT

معمارية توحيد الهارنسات الناجحة، عشر نظريات جديدة للهارنس، وخمس نظريات لـ Agentic AGI داخل MUT · MUG · MOUSSAIFXT

Harness Engineering · Virtual Context · HyperKnowledge · Tool Use · Computer Use · Evidence · World Effect · Self-Evolution

إعداد وتركيب بحثي: زهير مساعف / Zouhair Moussaif

نسخة بحثية هندسية · 05 سبتمبر 2026 · A4 / Amiri / Arabic RTL

تنبيه منهجي: هذه الورقة تقترح برنامجاً معمارياً وهندسياً قابلاً للاختبار، لا تعتبر وجود طبقات أو أسماء أو حجم corpus دليلاً على تحقق AGI. النجاح يُقاس بأثر العالم، الاختبارات، إعادة التشغيل، المقارنة بخطوط أساس، ونقل التعلم عبر مهام جديدة.

الملخص والأطروحة المركزية

Abstract · Research Question · Contributions

تبحث هذه الورقة في سؤال هندسي محدد: كيف يمكن بناء AGI-oriented harness لا يكتفي باستدعاء نموذج لغوي وأدوات، بل ينسق المعرفة، السياق الممتد، التنفيذ، الوكلاء، الحماية، التحقق، واستثمار الخبرة داخل نظام قابل لإعادة البناء؟ تنطلق الورقة من أن الهارنسات الناجحة أثبتت أن جودة النظام لا تتحدد بالنموذج وحده. Prime Agent يبرز قيمة البيئة البرمجية المستمرة وContinual Harness والوكلاء المتفرعين؛ Hermes Agent يبرز الذاكرة والمهارات والتشغيل متعدد البيئات؛ Codex يبرز الفصل بين sandbox والموافقات، والواجهة البرمجية والمهارات؛ بينما توجيهات Anthropic للوكلاء طويلة الأفق تؤكد حفظ الحالة، التحقق، توزيع العمل، وضبط كثافة الاستدعاءات. [U1-U7]

المساهمة هنا ليست تجميع هذه الأنظمة في Frankenframework، بل استخراج الآليات التي أثبتت نفعها ثم إعادة صياغتها كعقود داخل AL RAZZI Hyper Harness المربوط بـ MOUSSAIFXT. المشروع الداخلي يعرّف MUT كطبقة توحيد النظريات، وMUG كحاكمية البرنامج، وMOUSSAIXT كغلاف تنفيذ عابر للبيئات؛ كما يضع سلسلة MCD/QMCD/MCAI/HAKIM-MCD و→ HyperKB → HyperSkill → HyperAbility → HyperLoop → HyperAgent ومحاكم الدليل والأثر. [M1-M5]

الإسهامات الرئيسة

- معمارية إنتاجية موحدة تتعامل مع Claude/Prime/Codex/Hermes كـ execution substrates قابلة للاستبدال، لا كملاك للحالة السيادية.
- تعريف Virtual Context / MCAS بحيث يتجاوز النظام قيد نافذة النموذج عبر paging وcompute-to-context دون الادعاء بسياق لا نهائي.
- عشر نظريات Harness جديدة، لكل منها آلية، أداة، بروتوكول، فرضية قابلة للهدم ومقياس.
- خمس نظريات Agentic-AGI تربط MUT وMUG وMOUSSAIXT بالحاكمية المعرفية، أثر العالم، واستمرارية الهوية الوكيلية.
- برنامج benchmark وablation وإعادة تشغيل يميز architecture-rich عن evidence-rich.

حد الادعاء: الورقة تصوغ research program + engineering architecture؛ أي نتيجة عن AGI لا ترتفع عن فرضية إلا بعد قياس مستقل ومتكرر.

المنهج وسلّم الإثبات

Claim-ranked engineering · Evidence tiers · Falsification

تستخدم الورقة منهجاً مزدوجاً. الأول **هندسي مقارن**: نراجع الآليات الواضحة في الهارنسات الحالية ونفصل بين ما هو runtime mechanism وما هو مجرد واجهة أو اصطلاح. الثاني **تركيبي مساعفي**: نربط تلك الآليات بسلالة المشروع التي تجعل الفرق النشط والسؤال الحاكم والحاكمية والدليل وأثر العالم وحدات تشغيل. لذلك لا يكفي أن نقول إن النظام يملك memory أو skills؛ يجب تحديد متى تُقرأ، من يملكها، كيف تُختبر، وكيف تساهم في إغلاق مهمة.

الرتبة	المعنى في هذه الورقة	مثال
E0	غير متحقق / فرضية أو تسمية	نظرية جديدة بلا تجربة
E1	موصوف أو مذكور في مصدر	ميزة موثقة في upstream
E2	Artifact أو عقد هندسي موجود	Schema / registry / module
E3	تشغيل واختبار محلي موثق	Golden test / trace
E4	مقارنة بخط أساس تحت ميزانية متطابقة	Ablation / matched-budget
E5	تكرار مستقل عبر مهام وبيئات	Reproduction / cross-domain

يعكس هذا السلم مبدأ corpus الداخلي: الملف المولّد لا يساوي النشر، ونداء الأداة لا يساوي أثر العالم. [M1,M3] كما تتبنى الورقة اختبار الحذف: إذا أزلنا طبقة مثل MCD routing أو Closure Court ولم يتغير النجاح أو الكلفة أو التعافي، فإن القيمة المعمارية لهذه الطبقة في ذلك المجال تصبح غير مثبتة.

قاعدة البحث

كل مفهوم جديد يجب أن يغيّر قراراً أو يضيف مقياساً أو يولّد اختباراً. المصطلح الذي لا يبدّل سلوك النظام ولا يمكن هدمه يُخفض إلى استعارة أو وصف، لا إلى نظرية.

خريطة الهارنسات الحديثة

Successful patterns rather than brand imitation

النمط المشترك في الهارنسات القوية هو نقل الذكاء من single prompt إلى نظام تشغيل حول النموذج. تختلف المشاريع في موضع القوة: بعضُها يجعل البيئة البرمجية المستمرة مركزاً، بعضُها يجعل الذاكرة والمهارة محوراً، وبعضها يركز على أمان تنفيذ الأوامر ومراحل الموافقة. لذلك نميز خمس طبقات مقارنة: **Context، Capability، Execution، Continuity، Verification.**

الهارنس	أقوى آلية قابلة للتبني	المحدودية التي نعالجها
Prime Agent	Persistent IPython/RLM، recursive subagents، Continual Harness، daemon continuity	نحتاج فصل التطور عن السيادة وإضافة World-Effect/Courts
Claude-style agentic engineering	long-horizon state tracking، parallel tool use، subagent discipline، filesystem rediscovery	نحتاج runtime محايداً للنموذج مع سجل claims/authority
Codex	sandbox + approval separation، typed SDK، streaming، AGENTS/skills، workspace scope	نضيف cross-harness routing وknowledge-as-OS
Hermes	persistent memory، self-created skills، progressive disclosure، checkpoints، cron، multiple backends	نحتاج ترقية المهارة عبر benchmark/court بدل self-promotion
NVIDIA AVO	long-horizon variation، persistent lineage، supervisor، stagnation redirect	نحوّلها إلى Evolution Plane محكوم لا self-modification مباشر
mini-SWE / Aider	minimal baseline، repo map، Git transactionality	مقياس لمنع الإفراط في المعمارية

بالتالي، الهدف ليس أن يعمل AL RAZZI مثل كل واحد منهم في آن واحد، بل أن يملك capability negotiation يسمح باستيراد القوة المناسبة لكل عقدة في Mission DAG مع بقاء الحالة السيادية والمعرفة والحاكمية داخل Moussaif.

Prime Agent: البرمجة المستمرة بدل تضخم السياق

Persistent RLM kernel · Continual Harness · Recursive agency

يقدم Prime Agent نموذجاً مهمّاً للهارنس البرمجي: لكل جلسة persistent IPython kernel، بينما يملك المضيف TypeScript تنفيذ الأبناء والمحاسبة ودورة الحياة. تظل المتغيرات والبيانات والبرامج في البيئة الخارجية، ويمكن للنموذج التعامل معها عبر كود بدلاً من إعادة إدخالها كاملة في كل turn. كما يسجل Continual Harness ملاحظات prompt والذاكرة ووصف المهارات ومواصفات subagents كحالة قابلة للتنقيح والرجوع. [U1,U2]

نستخلص أربع قواعد. أولاً، **context is programmable**: قيمة RLM ليست Python في ذاته، بل إمكان إرسال computation إلى البيانات. ثانياً، الاستدعاء المتكرر لوكيل ابن يجب أن يكون عملية runtime قابلة للمحاسبة لا فقرة نصية. ثالثاً، الاستمرارية تتطلب daemon/session state لا الاعتماد على بقاء الطرفية. رابعاً، تحسين السلوك الدائم يجب أن ينفصل عن system prompt غير القابل للعبث.

التحويل إلى Moussaif

HyperKernel → Prime RLM، Continual Cognition Store → Continual Harness، HyperAgent → child agents، Delegation → persistent context variables، EVCF/MCAS handles. ويضاف فوق ذلك: Evidence Ledger، Authority Gate، World-Effect verification، Closure Court.

المعيار الذي نرفض عنده التبني المباشر هو أن يصبح Prime هو مالك المهمة أو الذاكرة السيادية. في AL RAZZI ينبغي أن يكون PrimeAgentAdapter substrate اختيارياً: يستقبل عقداً محدوداً ويرجع structured result وevidence وartifacts، بينما يحتفظ Moussaif بالحالة الكلية.

Claude-style agentic engineering

Long-horizon state · Subagents · Context discipline · Verification

تشدد إرشادات Anthropic للوكلاء طويلة الأفق على التقدم المتدرج، حفظ الحالة في filesystem، إعادة اكتشاف الحالة بعد نافذة سياق جديدة، استعمال subagents عندما تكون هناك فائدة من التوازي أو عزل السياق، والتفريق بين الأفعال المحلية القابلة للعكس والأفعال المشتركة أو المدمرة. كما تدعم النماذج الحديثة التوازي في tool calls وتستفيد من أدوات تحقق مثل browser/computer use في المهام الطويلة. [U4]

هذه النقاط مهمة لأن الهارنس الناجح لا يحاول منع model reset، بل يجعل reset حالة طبيعية. عندما يستطيع الوكيل بدء نافذة جديدة، قراءة progress وGit وtests واستعادة تعريف المهمة، فإن الاستمرارية تصبح خاصية runtime لا خاصية ذاكرة سرية داخل النموذج. وهذا يطابق هدف Self-Reconstruction Memory في corpus.

أربعة مبادئ نتبناها

- Filesystem rediscovery: الحالة الأساسية قابلة لإعادة البناء من artifacts وevents لا من ملخص واحد.
- Subagent economy: لا spawn إلا للتوازي أو العزل أو الاستقلالية النقدية.
- Parallel tool calls when independent: التوازي مورد اقتصادي يُقاس بالزمن والكلفة وليس سلوكاً افتراضياً دائماً.
- Reversibility-aware autonomy: الأفعال المحلية القابلة للعكس أوسع من النشر أو الإرسال أو تغييرات البنية المشتركة.

نطور هذا المنطق عبر MUG: كل subagent يحمل authority ceiling، وكل عملية توازي تمر عبر AgentPopulationController كي لا تتحول كثرة الوكلاء إلى ضجيج.

Codex: هندسة التنفيذ والعزل

Sandbox/approval separation · Typed SDK · Skills · Repository harness

يعطي Codex درساً مركزياً: sandbox يحدد ما يمكن للعملية تقنياً الوصول إليه، بينما approval policy يحدد متى يحتاج الفعل إلى موافقة. الفصل بينهما يمنع خلط القدرة بالبروتوكول. كما تعرض واجهة Python الحالية نمطاً typed لبدء thread مع read_only/workspace_write/full_access، مع streaming وinterrupt وsteering. [U5,U6]

ومن جهة هندسة repository، توصي مواد OpenAI باستخدام AGENTS.md كسياق دائم، وفصل goals/plans/build/context، وتحويل الإجراءات المتكررة إلى skills بدلاً من تكرار prompt يدوي. [U7] هذه الفكرة تلتقي مباشرة مع HyperSkill Factory، لكن Moussaif يضيف شرطاً أقوى: المهارة الجديدة لا تُعتمد لأنها مفيدة مرة، بل تمر عبر trace موثق، اختبار، مقارنة baseline ومحكمة تعميم.

Codex → Moussaif

Sandbox يصبح جزءاً من Secure Execution Plane، approval يدخل Authority Engine، AGENTS.md يدخل Project Context Genome، وskills تتحول إلى HyperSkills ذات I/O وpreconditions وevidence وrecovery. أما SDK/event streaming فيصبح مرجعاً لواجهة AL RAZZI التي يجب أن تكون صالحة لـ CLI وـ UI بدون parsing للنص.

تنبيه إنتاجي: الأمان الفعلي لا يجوز أن يعتمد على حسن نية model أو prompt. يجب أن تكون القيود تحت طبقة agentic cognition، وأن تُختبر محاولات تجاوزها عبر MCP أو remote executor أو adapters.

Hermes: الذاكرة والمهارة والتشغيل متعدد البيئات

Persistent memory · Progressive skills · Checkpoints · Cron · Backends

Hermes Agent هو المرجع الأكثر قرباً من بيئتنا المحلية: ذاكرة مستمرة منسقة، skills تُحمّل حسب الحاجة، إنشاء مهارات من الخبرة، بحث في الجلسات السابقة، checkpoints قبل تعديل الملفات، subagents معزولون، MCP، cron، وterminal backends تشمل local وDocker وSSH وبيئات serverless. [U3] كما يكتشف ملفات سياق المشروع مثل AGENTS.md وCLAUDE.md ويستعمل progressive disclosure لتقليل token load.

القيمة المعمارية لهيرمس هي أنه يبرهن عملياً أن الوكيل يمكن أن يعيش خارج جلسة chat واحدة وأن يجمع بين CLI والبوابات والمهارات والذاكرة. لكن في Moussaif نريد درجة أعلى من الفصل: الذاكرة الحوارية ليست HyperKnowledge تلقائياً، والمهارة التي كتبها الوكيل لا تصبح production capability تلقائياً.

نضيف

- Epistemic status
- Skill promotion court
- World-effect verification
- Cross-harness canonical mission state
- EVCF virtual addressing
- Runtime performance registry

نحتفظ

- Context discovery
- Persistent curated memory
- Skills as procedural assets
- Checkpoints/rollback
- Multiple terminal backends
- Cron/background execution

وبالتالي، HermesAdapter داخل MOUSSAIFXT لا يختزل Hermes إلى shell. يمكنه التفاوض على capabilities مثل memory، terminal، MCP، browser، skills، وsubagents، لكن AL RAZZI يظل مالك MissionContract وEvidence وWorldEffect.

المصفوفة التركيبية: ماذا نرث وماذا نطور؟

Engineering inheritance matrix

المحور	Prime	Claude	Codex	Hermes	Moussaif upgrade
السياق	RLM/IPython	multiwindow state	repo instructions	context files	EVCF + MCAS + PEK
الذاكرة	harness state	filesystem continuity	project artifacts	curated memory	typed memory tiers + epistemic ledger
المهارات	executable Python	subagent/tool discipline	skills	self-created skills	HyperSkill + Generalization Court
الوكلاء	recursive/A2A	native orchestration	threads/tasks	isolated subagents	HyperAgent + population economics
الأمان	host constraints	reversibility guidance	sandbox + approvals	approval /command container	MUG Authority + Secure Execution Plane
التطور	refine/	self-correction	skills/postmortems	skills improve	shadow → eval → court → canary
الإغلاق	goals/gates	verify state	tests/diff	checkpoints	WorldEffect + ClosureCourt

هذه المصفوفة تنتج مبدأ **الوراثة الوظيفية**: لا نرث اسم feature، بل نرث علاقة سببية. فإذا كان persistent kernel يحسن معالجة السياق الكبير لأنه يحتفظ بالمتغيرات خارج prompt، فالمطلوب هو المحافظة على تلك العلاقة حتى لو كان implementation مختلفاً. وإذا كان checkpoint قبل edit يقلل خسارة العمل، فالمطلوب عقد rollback موحد عبر hosts.

قاعدة عدم التضخم

أي طبقة جديدة يجب أن تتفوق في ablation على baseline أبسط. وجود mini-SWE كوكيل صغير وفعال يذكّرنا أن الكفاءة قد تأتي من loop جيد وأدوات واضحة، وأن المعمارية الثقيلة يجب أن تبرر overheadها. [U8,U9]

AL RAZZI الحالي: من Specification إلى Runtime

Verified internal baseline and remaining gap

وفق تقرير النشر الداخلي الحالي، توجد طبقات فعلية في AL RAZZI OS: Command/Event contracts، runtime مفصول عن CLI، session persistence في project discovery، SQLite وregistries، PEK compiler، HyperKB indexing، repo map للـLoop/Ability/HyperSkill/ Agent، Zenith-style Active Difference، MCD router، delegation، Windows subprocess sandbox، Git transaction ToolGateway، Evidence Ledger، أربع independent verifier وCourts، Self-Reconstruction candidates، إضافة إلى golden suites. [M0]

لكن الورقة لا تساوي تقرير التنفيذ بالدليل النهائي. لذلك نضع V0 hypothesis: **AL RAZZI هو baseline instrumentable وليس بعدُ برهاناً على تفوق harness.** المطلوب الآن تحويل المكونات إلى مسار إنتاجي قابل للمقارنة: provider حقيقي، tool effects حقيقية، browser/computer tasks، context restart mid-mission، cross-harness delegation أكبر من نافذة النموذج، ثم ablation تحت ميزانية متساوية.

الفجوة الإنتاجية

- إثبات أن كل صندوق معماري يقع على causal path حقيقي داخل mission.
- قياس false-success بدلاً من الاكتفاء بعدد الاختبارات الناجحة.
- قياس context efficiency وrecovery وcross-domain transfer.
- إثبات أن self-evolution يرفع أداء held-out tasks ولا يخرق governance.
- إثبات أن MOUSSAIXT يستطيع تبديل substrate دون فقد الحالة أو المعنى.

هذه الفجوة هي نقطة انطلاق النظريات الجديدة في الصفحات التالية: النظريات ليست بديلاً عن الهندسة، بل تعريفاً لما يجب أن تبرهنه الهندسة.

MOUSSAIXT · MUG · MUT: البنية الثلاثية

Theory Unification · Program Governance · Execution Shell

تحدد وثائق المشروع ثلاث بنيات متكاملة: MUT يوحد النظريات والعلاقات والأسئلة والأدوات ومرشحي التجارب، MUG يحافظ على اتساق قواعد البحث والتشغيل والحاكمية، وMOUSSAIXT يربط models/tools/memory/agents/verification بالبيئات المحلية والسحابية. [M2] هذا الفصل مهم لأنه يمنع ثلاث أخطاء: أن تتحول النظرية إلى runtime غير مراقب، أو تتحول الحاكمية إلى شعارات خارج الكود، أو يصبح host الخارجي هو صاحب القرار.

MOUSSAIXT

السؤال: أين وكيف تُنفذ العقدة؟ أي model/harness/tool/host يحققها بأفضل evidence/cost؟

MUG

السؤال: من يملك سلطة تغيير الحالة، ما حدود الاستقلال، وما دليل الإغلاق؟

MUT

السؤال: ما الذي نعتقد أنه صحيح، وما العلاقات بين النظريات، وما التجربة التي تفصل بينها؟

تقترح هذه الورقة أن AL RAZZI هو **Harness Bridge** بين الثلاثة. عند كل mission، يستدعي MUT المعرفة والنظرية الملائمة، يفرض MUG السلطة والمحاكم، ويستعمل MOUSSAIXT substrate التنفيذ. لا يجوز لأي واحد منها أن يبتلع الاثنين الآخرين.

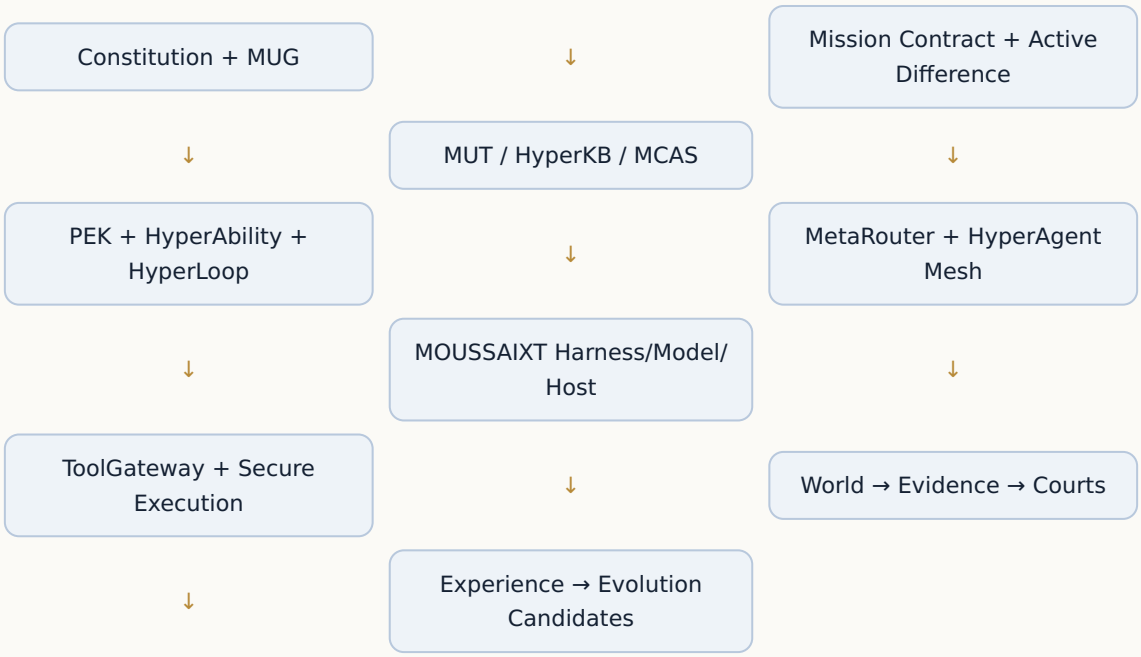
العقد الكلي

Goal → Governing Question → Active Difference → MUT knowledge → MUG authority → PEK/HyperAbility → Mission DAG → MOUSSAIXT substrate → Tool/Computer World → Evidence → Courts → Memory/Evolution

المعمارية الكلية المقترحة

From 12-layer stack to production-grade harness

يحافظ التصميم على السلم الداخلي الموثق: Constitution/Safety، Goal/Question/Difference، MCD/QMCD/MCAI، HKOS/World Model، Retrieval، HyperSkills، HyperAbilities، HyperLoops، Agents/Router، Tools/Execution، Verification/World Effect Memory/Evolution. [M1] ويضيف V5/V6 ثلاث planes عرضية: Virtual Context Plane، Secure Execution Plane، وEvolution Plane.



المبدأ الجديد هو أن **الطبقات ليست pipeline جامداً**. التحقق قد يعيد فتح world observation، Active Difference قد يعدّل World Model، وContextFault قد يستدعي resource إضافياً دون إعادة mission. لذلك runtime الأنسب هو event-driven state machine مع DAG، لا سلسلة prompts ثابتة.

Universal Provider & Harness Fabric

Models and harnesses as replaceable cognitive substrates

لكي يكون النظام فعلاً agentic OS يجب ألا يكون Claude-only أو GPT-only. تُفصل ثلاثة قرارات: **Model selection**، **Harness selection**، **Host selection**. فقد يعمل Hermes فوق نموذج محلي، أو AL RAZZI native فوق API بعيد، أو Prime Agent كبيئة RLM لعقدة بحثية، أو Codex لعقدة software engineering.

Capability Manifest

كل Model/Harness/Host يعلن persistence، context، structured output، tools، vision، coding، reasoning: capabilities، estimated cost/latency، historical reliability، checkpoint، streaming، sandbox، browser/computer، subagents.

يقوم MetaRouter بحل عقدة المهمة إلى requirement vector ثم يختار substrate بالاعتماد على fit وreliability وprivacy وcost وlatency وtool support وcontext need. لا يُعطى أي brand أولوية ثابتة. وبعد التنفيذ تُسجل النتيجة في RuntimePerformanceRegistry لتصبح routing policy evidence-driven.

هذا يخلق فرقاً جوهرياً عن multi-provider gateways العادية: gateway يختار endpoint، بينما Harness Fabric يختار شكل *cognition* والتنفيذ أيضاً. ويمكن لعقدتين في DAG نفسها استعمال harnessين مختلفين، بشرط أن يبقى protocol للنتائج والأدلة موحداً.

السياق الافتراضي الممتد: EVCF / MCAS

Virtual context as address space, not inflated prompt

أحد أكبر حدود الهارنسات هو مساواة قدرة النظام بنافذة النموذج. نقترح Extensible Virtual Context Fabric (EVCF) الذي تطور إلى Moussaif Cognitive Address Space (MCAS). الفكرة ليست سياقاً لا نهائياً، بل أن model context يصبح **resident working set**، بينما تبقى الملفات والـHyperKB والجداول والرموز والـtraces والصور والحالة الوكيلية موارد معنونة خارجية.

كل مورد يأخذ URI مثل ctx://evidence/... , cog://hypothesis/... , ctx://kb/... , ctx://repo/.... ويمكن للـruntime تنفيذ search/ page/query/compute/pin/evict/prefetch/diff. إذا احتاج reasoning مورداً غير resident يحدث ContextFault، فيحمّل الحاكم فقط الجزء اللازم ويعيد PEK.

معادلة تشغيلية

C_eff = quality(materialization, retrieval, compute, continuity) ،C_eff ≠ C_model. نعرّف القدرة السياقية الفعلية تقريبياً: تحت قيد نافذة C_model. زيادة bytes الخارجية وحدها لا تزيد الذكاء؛ الذي يزيده هو الوصول الصحيح منخفض الكلفة إلى المورد الصحيح.

يُختبر EVCF على repository أو corpus أكبر من نافذة النموذج: النجاح يتطلب حل المهمة دون إدخال كامل corpus، مع تسجيل pages accessed، tokens avoided، resident tokens، cache hit، ومعدل context faults.

Computer Use وTool Calling كطبقة أثر

From function call to governed world transition

الهارنس agentic لا يكتمل بـ function calling فقط. كل provider قد يعبّر عن tool call بطريقة مختلفة، لذلك يحول AL RAZZI أي طلب إلى ToolIntent موحد: expected effect، authority، agent، mission، arguments، tool_id. ثم يمر عبر ToolGateway قبل التنفيذ. النتيجة تصبح ToolObservation مع structured result وartifacts وevidence وerror وworld-effect candidate.

في computer use نطبق هرمية الموثوقية: API أولاً، ثم CLI، ثم direct data/filesystem، ثم DOM، ثم accessibility/UI automation، ثم visual clicking. المرئي هو fallback عندما توجد واجهة بنيوية أدق. كل ComputerAction يحمل intent وtarget وexpected effect وrisk وverification method.

After Action

· Re-observe · compare expected/observed delta · capture evidence · update World Model

Before Action

· Observe state · classify effect · check authority · checkpoint if needed

القاعدة الحاسمة: **click issued ≠ task succeeded.** نجاح computer use لا يثبت إلا عندما يطابق observed state الأثر المتوقع أو تقرر المحكمة أن الفرق المتبقي مقبول.

الأمن والحاكمية: القدرة لا تعني الإذن

Authority, sandbox, trust and reversible effects

نتبنى الفصل الذي يتضح في Codex ونعيده داخل MUG: Sandbox Policy هو قيد تقني، أما Authority Policy فهو قرار معنوي/تشغيلي بشأن السماح. يمكن أن يملك process وصولاً تقنياً لملف، لكن المهمة غير مخولة بحذفه. كما لا يجوز اعتبار نجاح sandbox approval دليلاً على أن المستخدم قصد نشر أو إرسال شيء.

نقترح أربع trust dimensions مستقلة: ProjectTrust، PluginTrust، ToolTrust، RemoteHostTrust. محتوى repository أو webpage يعامل كبيانات غير موثوقة ولا يستطيع تغيير الدستور أو authority. الأسرار تدخل عبر CredentialBroker كمراجع scoped ولا تُنسخ في PEK أو logs.

Effect classes

READ_ONLY → REVERSIBLE_WRITE → COMPENSATABLE_WRITE → IRREVERSIBLE/EXTERNAL_EFFECT. ترتفع متطلبات checkpoint/evidence/approval مع صعوبة التراجع ونطاق الضرر.

يقترح البحث اختباراً adversarial واضحاً: توفير MCP tool يمكنه remote execution ثم طلب عملية منعها sandbox المحلي. يجب أن يمنع MUG انتقال السلطة عبر حدود adapter دون إعادة تصنيف الأثر والثقة. هذه ليست feature إضافية بل اختبار لسلامة النظام السيادي.

Evidence · Courts · World Effect

Completion is an observed state transition

تضع السلالة الداخلية مبدأ صارماً: output النموذج لا يساوي أثر العالم. [M1,M3] لذلك تمثل كل مهمة بثلاث حالات: before، intended delta، observed after. تُربط الأدلة بـ claims محددة: process result، browser state، database query، API readback، test result، diff، artifact hash أو human approval.

تعمل Courts كطبقات منفصلة لا كـ"critique prompt" واحد. Evidence Court يسأل هل الادعاء مسنود؛ Falsification Court يبحث عن اعتراض حاسم؛ Regression Court يحمي وظائف سابقة؛ Closure Court يقارن Mission Contract بالواقع؛ وNo-Loss Court يحمي الأصول غير المقصودة.

معيار الإغلاق

COMPLETE ⇔ mandatory criteria satisfied ∧ no critical remaining gap ∧ evidence sufficient ∧ world effect verified ∧ no unacceptable regression. وإلا تكون الحالة PARTIAL أو BLOCKED أو REOPEN.

التجديد هنا هو ربط هذه المحاكم بـMUG لا بنموذج بعينه. يمكن أن يكون المنتج Claude أو Codex أو Hermes، لكن التحقق النهائي يظل مستقلاً عن brand وعن confidence المولد.

التطور طويل الأفق: من AVO إلى Moussaif Evolution Plane

Variation, lineage, stagnation and governed promotion

عرضت NVIDIA في أغسطس 2026 Agentic Variation Operators (AVO) كمعمارية طويلة الأفق تجمع persistent memory مع supervision وcandidate lineage، وذكرت تشغيل بحث تحسين kernels لسبعة أيام واستكشاف مئات الاتجاهات، ثم نقل النظام إلى ARC-AGI-3 وتحقيق نتيجة كاملة على public set. [U10] نستخدم هذه النتيجة كدليل على أهمية harness design، لا كبرهان عام على AGI.

نحوّل المبدأ إلى MoussaifEvolutionPlane: كل variation ينتج candidate له parents وhypothesis وmutation type وscore وevidence. عندما ينخفض ProgressSlope أو يتكرر الخطأ، يفعّل StagnationDetector بدائل: model جديد، harness مختلف، context slice آخر، hypothesis جديدة، أو تبسيط.

حد التطور

candidate لا يغير production مباشرة. المسار: PROPOSED → SHADOW → BENCHMARK → COURT → CANARY → PROMOTED/REJECTED. تغييرات الأمن والدستور لا تُرقى ذاتياً.

هذه الطبقة تجعل التعلم من الخبرة قابلاً للاختبار: التحسن الحقيقي هو أن النسخة الجديدة تتفوق على frozen baseline في held-out tasks دون زيادة غير مضبوطة في الكلفة أو regression.

النظرية H1: الغشاء الحاسوبي السيادي

Sovereign Harness Membrane Theory

H01 · Sovereign Harness Membrane Theory

تقترح النظرية أن الهارنس ليس غلافاً خارجياً للنموذج، بل غشاءً ينظم التبادل بين cognition والعالم. النموذج يولد اقتراحات، لكن الغشاء يحدد ما يدخل من سياق، وما يخرج كفعل، وما يُقبل كذاكرة. لذلك تتحول جودة الوكيل من خاصية داخل النموذج إلى خاصية مشتركة بين model وharness وworld coupling.

الآلية / Mechanism

Agency = f(Model Capability, Context Governance, Action Boundary, Evidence Feedback). إذا ضعف أي عامل قد يكون النموذج قوياً لكن الوكالة هشة. الغشاء يملك adapters وauthority وcontext وevidence ولا يسمح بانتقال غير موثق بين الداخل والخارج.

البروتوكول التشغيلي

Observe → classify ingress/egress → authorize → execute → readback → ledger. كل crossing يصبح event يمكن تتبعه.

الأدوات المقترحة

- HarnessMembraneInspector
- AuthorityBoundaryTracer
- ContextIngressPolicy
- EffectEgressPolicy

قابلية الهدم والقياس

Falsification. إذا لم يرفع الغشاء reliability أو يقلل false-success مقارنةً بحل مباشر بنفس model/tools، فالدعوى تفشل في ذلك المجال.
Metrics. context efficiency، recovery rate، unauthorized-effect rate، false-success rate.

صلة النظرية بالبنية: MUG يملك السيادة؛ MOUSSAIXT يوفّر العبور؛ AL RAZZI يطبق الغشاء كـruntime contract.

النظرية H2: سيادة السياق الافتراضي

Virtual Context Sovereignty Theory

H02 · Virtual Context Sovereignty Theory

الذكاء الوكيلي طويلِ الأفق لا ينبغي أن يكون رهينة لما هو resident في نافذة النموذج. السيادة السياقية تعني أن النظام، لا النموذج، يملك قرار ما يجب الاحتفاظ به، وما يُعنون خارجياً، وما يُسترجع أو يُحسب عند الحاجة.

الآلية / Mechanism

Cognitive Working Set = selected materialization(MCAS, mission state, Δ, model budget). السياق الخارجي يبقى addressable، والـPEK هو materialization مؤقت لا نسخة من المعرفة الكلية.

البروتوكول التشغيلي

عند الحاجة إلى معلومة غير resident: emit ContextNeed → resolve URI → choose representation → materialize minimal page → continue.

الأدوات المقترحة

- VirtualContextGovernor
- ContextFaultResolver
- ResourcePager
- ComputeToContextEngine
- ContextTrace

قابلية الهدم والقياس

Falsification. تفشل النظرية إذا لم يتمكن النظام من حل مهام على corpus أكبر من نافذة model دون full stuffing، أو إذا كانت أخطاء paging أعلى من baseline مباشر.

Metrics. mission success، cache hit، page-fault latency، tokens avoided، useful-context ratio.

صلة النظرية بالبنية: امتداد مباشر لـEVCF/MCAS وHyperKB وROFSC ضمن MUT.

النظرية H3: تحويل المعرفة إلى قدرة تنفيذية

Executable Knowledge Transduction Theory

H03 · Executable Knowledge Transduction Theory

لا تصبح المعرفة ذات قيمة agentic لأنها قابلة للاسترجاع فقط. المعرفة تنتقل من claim إلى قدرة عندما تُربط بشروط تفعيل، أدوات، إجراءات، أدلة وفشل. النظرية تصف التحويل من HyperKnowledge إلى HyperSkill ثم HyperAbility/HyperLoop.

الآلية / Mechanism

K → Contract → Skill → Composition → Verified Effect → Experience. كل مرحلة تضيف executable semantics دون حذف provenance.

البروتوكول التشغيلي

استخرج trace ناجحاً → حدد invariant الإجرائي → أنشئ candidate skill → اختبره على حالات جديدة → سجل الحدود → روّجه أو ارفضه.

الأدوات المقترحة

- KnowledgeUnitCompiler
- HyperSkillFactory
- CapabilityLinker
- GeneralizationCourt

قابلية الهدم والقياس

Falsification. إذا لم يرفع تحويل المعرفة reuse أو transfer مقارنة باسترجاع النص الأصلي، فالنظرية تضعف.
Metrics. skill transfer gain، repeated-task cost، regression، scope precision.

صلة النظرية بالبنية: تجسيد مباشر لـKnowledge-as-OS في [M4] corpus وربطه بالهارنس.

النظرية H4: الحوسبة الموزونة بالفرق

Difference-Weighted Compute Theory

H04 · Difference-Weighted Compute Theory

المشكلة ليست ندرة compute فقط بل سوء توزيعه. ينبغي تخصيص التفكير والاسترجاع والوكلاء والتحقق بحسب الفرق الذي يمكنه تغيير القرار، لا بحسب حجم النص أو طول المهمة.

الآلية / Mechanism

MCD/QMCD .Budget_i ∝ w(Δ_i) = impact × uncertainty × dependency × risk × information_gain / cost
يمثلان آلية تقدير الفرق والسؤال الحاكم.

البروتوكول التشغيلي

استخرج ΔA → احسب الوزن → خصص /mode FAST
STANDARD/DEEP/COURT → أعد الوزن بعد كل evidence
.event

الأدوات المقترحة

- ActiveDifferenceExtractor
- MCDScorer
- ComputeAllocator
- QuestionRouter

قابلية الهدم والقياس

Falsification. اختبار الحذف: إذا أعطى توزيع متساوٍ للcompute نفس النجاح والكلفة، فلا قيمة إضافية للموجه الوعائي.
Metrics. success under fixed budget، unnecessary tool/model calls، latency، cost per verified delta.

صلة النظرية بالبنية: امتداد هندسي لMCD/QMCD وComputational Vascularity.

النظرية H5: اتحاد الهارنسات واستقلال الركيزة

Harness Federation & Substrate Independence Theory

H05 · Harness Federation & Substrate Independence Theory

إذا كانت المهمة والسيادة والذاكرة مرتبطة بهارنس واحد، فإن النظام لا يكون عاماً بل تابعاً للركيزة. تقترح النظرية أن كل harness خارجي يجب أن يكون capability provider محدوداً تحت Mission Contract موحد.

الآلية / Mechanism

Canonical Mission State ⊥ Execution Substrate. يمكن تبديل Claude/Codex/Hermes/Prime أو local harness بين العقد دون تبديل هوية المهمة.

البروتوكول التشغيلي

حل → node requirements → query available harnesses → route → collect typed result/evidence → merge only verified artifacts into canonical state.

الأدوات المقترحة

- HarnessAdapter
- HarnessRouter
- CapabilityNegotiator
- CrossHarnessResultNormalizer

قابلية الهدم والقياس

Falsification. تفشل النظرية إذا أدى تبديل substrate إلى فقد حالة، اختلاف semantics غير مكتشف، أو عدم قدرة النظام على إكمال نفس contract.

Metrics. normalized evidence completeness، substrate failover success، routing gain، cross-harness continuity.

صلة النظرية بالبنية: هي قلب MOUSSAIXT باعتباره execution shell عابراً للبيئات.

النظرية H6: إغلاق الفعل المقترن بالدليل

Evidence-Coupled Action Closure Theory

H06 · Evidence-Coupled Action Closure Theory

الهارنس القوي لا يغلق المهمة عند اكتمال generation أو tool return. الإغلاق نفسه عملية معرفية تربط claim بأثر خارجي وباختبار لمطابقة delta المطلوبة.

الآلية / Mechanism

Closure = match(IntendedDelta, ObservedDelta, Evidence, Constraints). evidence لا يكون ملحقاً بعد التنفيذ بل جزءاً من contract قبل التنفيذ.

البروتوكول التشغيلي

قبل الفعل حدد expected evidence → نفذ → اقرأ الحالة → التقط evidence → شغل close/reopen court.

الأدوات المقترحة

- EvidenceLedger
- WorldEffectVerifier
- ClosureCourt
- ArtifactHasher

قابلية الهدم والقياس

Falsification. تُهدم النظرية إذا لم تقلل false-success على مهام الأدوات مقارنةً بإغلاق مبني على self-report.

Metrics. verified completion، court reopen rate، readback coverage، false-success.

صلة النظرية بالبنية: تجسيد لـWorld-Effect Loop ومحاكم الفجوة/التكذيب في MOUSSAIF.

النظرية H7: التنويع المحفوز بالركود

Stagnation-Triggered Variation Theory

H07 · Stagnation-Triggered Variation Theory

الاستقلال الطويل لا يحتاج loop أطول فقط، يحتاج اكتشاف أن الاستراتيجية الحالية لم تعد تنتج فرقاً مفيداً. عند الركود يجب تغيير الفرضية أو الأداة أو model/harness أو تمثيل السياق.

الآلية / Mechanism

If d(Progress)/dt < ε for window W → trigger VariationOperator. التغيير نفسه موجه بسجل الفشل وليس random mutation.

البروتوكول التشغيلي

راقب ProgressVector → اكتشف plateau → ولّد alternative candidate مع hypothesis → اختبر cheapest discriminator → retain/reject.

الأدوات المقترحة

- StagnationDetector
- AgenticVariationOperator
- CandidateLineageGraph
- MetaHarnessSupervisor

قابلية الهدم والقياس

Falsification. إذا زاد variation الكلفة دون رفع escape-from-stagnation أو objective score، يجب تعطيله.

Metrics. compute wasted before redirect، progress slope، candidate diversity، stagnation recovery.

صلة النظرية بالبنية: يربط AVO بالMCD: الركود نفسه يصبح Active Difference عالي الأولوية.

النظرية H8: الاستمرارية بإعادة البناء

Reconstructive Continuity Theory

H08 · Reconstructive Continuity Theory

الوكيل طويل الأفق لا يحتاج الحفاظ على نفس hidden context؛ يحتاج القدرة على إعادة بناء الحالة الحاسمة بعد restart أو context reset. الاستمرارية خاصية للآثار والعقود، لا لذاكرة نموذجية مستمرة.

الآلية / Mechanism

Identity_t+1 = reconstruct(MissionContract, event log, artifacts, checkpoints, unresolved gaps). الملخص وحده غير كافٍ إذا أخفى الفشل أو السلطة أو side effects.

البروتوكول التشغيلي

عند restart: load snapshot → replay tail → verify external state fingerprints → rebuild PEK → resume READY node.

الأدوات المقترحة

- StateReconstructor
- CheckpointManager
- EventJournal
- ContinuityScanner

قابلية الهدم والقياس

Falsification. يفشل الادعاء إذا استمر agent نصياً لكنه يعيد نفس الأخطاء، يكرر effects، أو ينسى constraints بعد restart.

Metrics. recovery time، state divergence، duplicate-effect rate، restart continuity.

صلة النظرية بالبنية: امتداد لـSelf-Reconstruction Memory وHermes/Prime daemon continuity.

النظرية H9: أيض القدرة

Capability Metabolism Theory

H09 · Capability Metabolism Theory

المهارات ليست assets ثابتة؛ يجب أن تدخل دورة حياة تشبه الأيض: اكتساب، اختبار، استعمال، تقوية، تراجع، حجر أو إحلال. بذلك يتجنب النظام تراكم آلاف skills متشابهة أو مكسورة.

الآلية / Mechanism

Experience → Candidate → Validation → Active Capability → Monitoring → Repair/Quarantine/Supersede.

الجودة تاريخية ومجالية.

البروتوكول التشغيلي

بعد كل invocation حدث success/failure/cost/scope → إذا هبطت reliability تحت threshold فعّل quarantine أو repair candidate.

الأدوات المقترحة

- CapabilityHealthRegistry
- SkillDeduplicator
- QuarantineManager
- GeneralizationCourt

قابلية الهدم والقياس

Falsification. تُهدم النظرية إذا لم تقلل skill bloat أو routing errors أو regression مقارنة بسجل ثابت غير مراقب.

Metrics. transfer success، quarantine recovery، duplicate rate، active-skill precision.

صلة النظرية بالبنية: تطوير لـHyperSkill Factory وzero-duplication governance.

النظرية H10: جينوم الهارنس التكيفي

Adaptive Harness Genome Theory

H10 · Adaptive Harness Genome Theory

يمكن تمثيل configuration الكلي للهارنس كجينوم من سياسات السياق والتوجيه والوكلاء والأدوات والتحقق والذاكرة. التعديل يصبح نسخة candidate قابلة للمقارنة، لا تغييرات مبعثرة في ملفات config.

الآلية / Mechanism

Genome = {context, router, models, harnesses, skills, loops, verification, recovery, security}. كل genome يحمل parent وbenchmark signature.

البروتوكول التشغيلي

freeze production genome → generate candidate → shadow replay → held-out benchmark → court → canary → promote/rollback.

الأدوات المقترحة

- HarnessGenomeRegistry
- ShadowRunner
- TrajectoryReplayEngine
- CanaryPromoter
- AutoRollback

قابلية الهدم والقياس

Falsification. تفشل النظرية إذا لم يمكن إرجاع الفرق في الأداء إلى genome version أو إذا أدى التطور إلى regressions غير مكتشفة.
Metrics. reproducibility، canary rollback، regression rate، gain over frozen genome.

صلة النظرية بالبنية: الجسر بين Self-Reconstruction وEvolution Laboratory وAVO-style lineage.

تركيب النظريات العشر

A unified Harness Science hypothesis

تتكامل النظريات العشر في مسار واحد: H1 يعرّف الهارنس كغشاء سيادي، H2 يحرره من نافذة model، H3 يحول المعرفة إلى capability، H4 يوزع compute على الفرق، H5 يحرره من substrate، H6 يربط الإغلاق بالأثر، H7 يعطيه قدرة تغيير الاستراتيجية، H8 يحفظ الهوية عبر restart، H9 يمنع تراكم قدرات ميتة، H10 يجعل configuration نفسه قابلاً للتطور المقاس.



الفرضية الجامعة هي أن **العمومية الوكيلية لا تنتج فقط من general model، بل من قدرة harness على إدارة انتقالات المعرفة والحالة والفعل عبر الزمن والبيئات.** هذه فرضية هندسية قابلة للهدم: يمكن مقارنة full stack مع minimal agent loop على نفس model وtools وbudget. إذا لم يظهر gain في transfer، long-horizon completion أو recovery، فالتعقيد غير مبرر.

أداة جامعة: Harness Causal Profiler

يبني graph يربط كل طبقة بـ events والقرارات والنتائج، ثم يتيح ablation: ماذا يحدث إذا أوقفنا ContextGovernor أو MCD أو Court أو skill factory؟ الهدف تحويل architecture claims إلى causal evidence.

نظرية A1: العمومية عبر أثر العالم المحكوم

Governed World-Effect Generality

H01 · Governed World-Effect Generality

تقترح النظرية أن معياراً أقوى للAgentic AGI هو القدرة على تحقيق أهداف متنوعة في بيئات جديدة مع الحفاظ على حاكمية الدليل والسلطة والتراجع، لا مجرد حل benchmark نصي. العمومية هنا علاقة بين adaptation وverified world effect.

الآلية / Mechanism

Generality ≈ Transfer × VerifiedEffect × Governance × Recovery. أي مكوّن صفري يخفض الادعاء حتى لو كانت الإجابات اللغوية قوية.

البروتوكول التشغيلي

ثبت model/budget → شغل مهام /software/research → قس transfer وverified effects وrecovery → browser/data → قارن baselines.

الأدوات المقترحة

- GeneralityBenchmarkRunner
- WorldEffectSuite
- CrossDomainTransferMatrix

قابلية الهدم والقياس

Falsification. إذا تفوق النظام فقط في مجال واحد أو بسبب budget أكبر، لا تدعم النتائج عمومية الوكالة.

Metrics. policy adherence، recovery under novelty، verified completion، cross-domain transfer.

صلة النظرية بالبنية: يجسد هدف MOUSSAIF كـAGI-oriented architecture دون ادعاء AGI مثبت.

نظرية A2: الإغلاق المزدوج MUT–MUG

MUT–MUG Dual Closure Theory

H02 · MUT–MUG Dual Closure Theory

المهمة العامة تحتاج إغلاقين مستقلين: إغلاق معرفي يقرر ما إذا كانت النظرية/الخطة متماسكة ومسنودة، وإغلاق حاكمي يقرر ما إذا كان الفعل مسموحاً ومتوافقاً مع الهدف والسلطة. MUT وMUG يمثلان هذين القطبين.

الآلية / Mechanism

Accept(Action) = EpistemicClosure_MUT ∧ GovernanceClosure_MUG. يمكن أن تكون الخطة صحيحة تقنياً لكنها غير مخولة، أو مخولة لكنها معرفياً ضعيفة.

البروتوكول التشغيلي

قبل high-impact node: حل claims/evidence في MUT → حل authority/constraints في MUG → لا تنفيذ إلا إذا اتفق الإغلاقان.

الأدوات المقترحة

- MUTClaimResolver
- MUGAuthorityResolver
- DualClosureGate

قابلية الهدم والقياس

Falsification. تُهدم النظرية إذا لم يقل هذا الفصل أخطاء scope أو false certainty أو unauthorized effects.

Metrics. repair after disagreement، authority violation، epistemic rejection precision.

صلة النظرية بالبنية: تجسيد مباشر لتعريف النظام الثلاثي في [M2].

نظرية A3: الوكالة المستقلة عن الركيزة

MOUSSAIXT Substrate-Independent Agency

H03 · MOUSSAIXT Substrate-Independent Agency

إذا كانت هوية الوكيل مرتبطة بنموذج أو provider أو حاسوب واحد، فلا توجد وكالة نظامية مستقرة. تقترح النظرية أن الوكالة تُعرّف بالحالة والأهداف والمعرفة والسلطة، بينما models/harnesses/hosts أعضاء قابلة للاستبدال.

الآلية / Mechanism

Agent Identity = Mission + WorldModel + Memory + Authority + Capability Graph، لا current model = session.

البروتوكول التشغيلي

ابدأ mission على checkpoint → substrate A → فشل A → استأنف على B → تحقق من semantics وeffects.

الأدوات المقترحة

- MOUSSAIXTAdapterMesh
- StatePortableMission
- HarnessFailoverTester

قابلية الهدم والقياس

Falsification. يفشل الادعاء إذا كانت النتائج تعتمد اعتماداً غير معلن على context مخفي أو API خاص لا يمكن ترجمته.

Metrics. provider portability، semantic drift، state preservation، failover success.

صلة النظرية بالبنية: هذه النظرية تعطي MOUSSAIXT دوراً أعمق من adapter collection: حفظ agency عبر substrata.

نظرية A4: الهوية الوكيلية بإعادة البناء الذاتي

Self-Reconstructive Agentic Identity Theory

H04 · Self-Reconstructive Agentic Identity Theory

الاستمرارية الوكيلية في نظام طويل الأفق لا تتطلب وعياً مستمراً، بل بنية يمكنها أن تعيد بناء وضعها المقصود من آثار موثوقة بعد الانقطاع. بذلك تصبح الهوية تشغيلية وقابلة للاختبار.

الآلية / Mechanism

I_agent(t) = reconstructable invariant state :artifact lineage، authority، unresolved gaps، commitments، goal، learned procedures.

البروتوكول التشغيلي

اقطع process/context عشوائياً أثناء mission → restart على model مختلف → اطلب استئنافاً دون إعادة prompt الكامل → قس الاتساق.

الأدوات المقترحة

- IdentityReconstructor
- CommitmentLedger
- ContinuityBenchmark

قابلية الهدم والقياس

Falsification. يفشل إذا ضاعت commitments أو تكررت irreversible actions أو تغيّر الهدف دون evidence.

Metrics. cross-model identity drift، restart convergence، duplicate effect، commitment retention.

صلة النظرية بالبنية: يربط Self-Reconstruction Memory بالMCAS وبـMUG.

نظرية A5: الحاكمية الإبستيمية في العالم المفتوح

Open-World Epistemic Governance Theory

H05 · Open-World Epistemic Governance Theory

في البيئات الجديدة لا يكفي أن يختار الوكيل فعلاً؛ يجب أن يدير ما لا يعرفه. النظرية تجعل claim/hypothesis/evidence/counterevidence كائنات runtime، وتسمح بـ"لا أدري" كحالة صحيحة تؤدي إلى experiment أو retrieval بدلاً من hallucinated closure.

الآلية / Mechanism

Unknown → Hypothesis Set → Discriminating Experiment → Evidence Update → Action. الحاكمية تحدد متى يكفي الدليل ومتى يلزم التوقف أو التصعيد.

البروتوكول التشغيلي

في unknown environment: أنشئ hypotheses متعددة → نفذ أرخص تجربة مميزة → حدّث confidence → استمر حتى threshold أو block.

الأدوات المقترحة

- EpistemicLedger
- HypothesisEngine
- ExperimentPlanner
- FalsificationCourt

قابلية الهدم والقياس

Falsification. تُهدم إذا لم تقلل premature commitment أو hallucination في مهام novelty مقارنة baseline.

Metrics. success under novelty، falsification rate، hypothesis efficiency، calibration.

صلة النظرية بالبنية: تصل MCD/QMCD وGovernance-Space/FSKE بالسلوك agentic القابل للاختبار.

أطلس الأدوات المقترحة

20 tools that operationalize the theories

#	الأداة	الوظيفة
01	MissionContract Compiler	تحويل الهدف إلى acceptance/constraints/authority/evidence.
02	ActiveDifference Extractor	استخراج novelty/gap/risk/dependency/leverage.
03	GoverningQuestion Engine	اختيار السؤال الأعلى قيمة معلوماتية وعملية.
04	VirtualContextGovernor	paging/pinning/eviction/prefetch للسياق.
05	MCAS Resolver	حل ctx:// وcog:// إلى موارد وتمثيلات.
06	PEK Compiler	تجميع resident working set لكل turn/node.
07	HyperKernel	compute-to-context، query، scripts، delegation.
08	HyperSkill Factory	استخراج procedure قابل لإعادة الاستعمال من traces.
09	Capability Metabolism Registry	صحة المهارات والحجر والتفوق/الإحلال.
10	MetaRouter	اختيار model/harness/host/context/verification depth.
11	HarnessAdapter Mesh	Prime/Hermes/Codex/Claude/OpenCode integration.
12	ToolGateway	normalize + authority + execution + observations.
13	ComputerUse Runtime	DOM/UI/visual actions مع post-state verification.
14	EvidenceLedger	ربط claims بأدلة قابلة لإعادة القراءة.
15	WorldEffectVerifier	مقارنة intended vs observed delta.
16	ClosureCourt	COMPLETE/PARTIAL/BLOCKED/REOPEN.
17	StagnationDetector	كشف plateau/loop/churn.
18	VariationOperator	اقتراح candidates وتغيير الاستراتيجية.
19	TrajectoryReplayEngine	counterfactual replay دون تكرار side effects.
20	HarnessGenome Registry	نسخ/benchmarks/canary/rollback للهارنس.

كل أداة يجب أن تكون plugin/contract قابلاً للاستبدال؛ الأسماء هنا مقترحات معمارية، لا شرطاً على أسماء الملفات.

البروتوكولات التشغيلية

12 protocols from discovery to evolution

P1 · Project Archaeology

→ Read-only discovery → source precedence → Context Genome trust decision.

P2 · Mission Contract

→ Goal → desired state → acceptance → constraints → authority evidence policy.

P3 · Virtual Context

Mount → index → select representation → page fault → PEK trace.

P4 · Capability Composition

Knowledge → HyperSkills → Dynamic HyperAbility → bounded HyperLoop.

P5 · Cross-Harness Delegation

Node requirements → capability negotiation → substrate → typed result/evidence.

P6 · Tool/Computer Effect

Intent → authority → checkpoint → execute → observe → verify.

P7 · Remaining-Gap Reopening

→ []Original requirement ↔ observed state → RemainingGap replan/close.

P8 · Independent Verification

Producer ≠ verifier for medium/high-risk nodes.

P9 · Recovery & Rollback

/Classify failure → retry/alternate/replan/checkpoint/rollback human.

P10 · Experience Compilation

Trajectory → lesson candidate → scope → test → generalization court.

P11 · Harness Evolution

→ Candidate genome → shadow → ablation/benchmark → canary promote/reject.

P12 · Reproducible Evaluation

→ Freeze model/tools/data/prompts/budget → seeds → raw traces report CIs.

العقود والSchemas المشتركة

Machine-readable interfaces for a UI-independent runtime

لكي يبقى AL RAZZI قابلاً للتحول من CLI إلى UI يجب أن تكون الحقيقة في contracts لا في terminal strings. نقترح مجموعة schemas ثابتة versioned:

```
MissionContract { goal, desired_state, acceptance[], constraints[],
  authority, evidence_policy, budgets, project_scope }
MoussaifEvent { event_id, type, mission_id, trace_id, actor, payload, ts }
VirtualContextObject { uri, source, hash, type, size, provenance,
  freshness, materializations[], policy }
ToolIntent { tool_id, args, effect_class, expected_effect, authority }
ToolObservation { status, result_ref, artifacts[], evidence[], error }
WorldEffect { before, intended_delta, observed_after, observed_delta, verdict }
PatchCandidate { type, parent, hypothesis, diff, tests, score, evidence }
```

يجب أن تكون CLI وTUI وfuture Windows UI عملاء فوق نفس Command/Event API. وتُحفظ JSONL كعقد public versioned للstream، بينما internal events يمكن أن تستخدم transport آخر.

Invariant

لا قيمة لأي UI status لا يمكن تتبعه إلى state/event حقيقي. progress bar المولد بالوقت أو "status "done غير المدعوم بـClosureCourt يعتبر عيباً معمارياً.

خوارزمية التنفيذ المرجعية

Reference mission algorithm and stopping logic

```
run_mission(goal):
  M = compile_mission(goal)
  S = observe_world(M.scope)
  while not terminal(M):
      Δ = active_difference(M, S)
      q = governing_question(M, Δ)
      ctx = EVCF.materialize(q, Δ, M.budget)
      cap = compose_capability(ctx, Δ)
      plan = build_or_update_DAG(M, cap, Δ)
      node = choose_ready_node(plan)
      substrate = MetaRouter.route(node)
      result = execute_via_MOUSSAIXT(substrate, node)
      evidence = capture(result)
      S2 = observe_world(node.targets)
      verdict = verify_effect(S, S2, node, evidence)
      update_ledgers(verdict, evidence)
      gaps = reopen_remaining_gaps(M, S2)
      if stagnating(gaps, history): vary_strategy()
      if failure(verdict): recover_or_rollback()
      S = S2
  return ClosureCourt(M, S, evidence_ledger)
```

هذه الخوارزمية تضمن أن السؤال والسياق والقدرة والsubstrate لا تُحدد مرة واحدة في البداية، بل تتغير مع evidence. ويعطي ذلك لMCD معنى runtime: الفرق النشط يعيد توزيع الحوسبة ويعدل السؤال الحاكم والDAG.

شرط التوقف ليس token budget وحده. إذا نفد budget قبل الإغلاق تكون النتيجة PARTIAL/BLOCKED مع unresolved gaps، لا نجاحاً وهمياً.

برنامج الاختبار الإنتاجي

Benchmark matrix for harness-level evidence

العائلة	المهمة	المقياس الحاكم
Software	bug fix / feature / refactor / recovery	tests + diff quality + regression + cost
Repository	large-code navigation above context limit	context efficiency + symbol accuracy
Research	multi-source synthesis + hypothesis testing	citation accuracy + claim calibration
Browser	stateful web workflow	verified state transitions + recovery
Computer	Windows app interaction	action success + post-state verification
Long horizon	multi-hour bounded mission	progress slope + restart continuity
Novelty	unknown tool/environment	hypothesis efficiency + adaptation
Federation	same DAG across multiple harnesses	routing gain + state preservation
Evolution	repeated failure → candidate policy	held-out gain + regression

تتم المقارنة على الأقل بين: A0 minimal loop، A1 + repo map، A2 + HyperKB، A3 + MCD/Active Difference، A4 + Courts، A5 + EVCF، A6 full AL RAZZI. ويمكن إضافة Prime/Hermes/Codex كـbaselines خارجية عندما تسمح البيئة والترخيص.

المقاييس العليا

Verified completion rate · false-success rate · cost per verified mission · recovery success · restart continuity · useful-context ratio · cross-domain transfer · harness-routing gain · self-evolution held-out gain.

محكمة الهدم والAblation

How the architecture can prove itself wrong

أكبر خطر على مشروع غني بالمفاهيم هو أن يصبح كل failure مبرراً لإضافة طبقة جديدة. لذلك تُعامل كل نظرية وآلية كفرضية يمكن حذفها. الاختبار الأساسي ليس "هل تعمل؟" بل: **هل تضيف قيمة فوق baseline أبسط تحت نفس الميزانية؟**

ما الذي نقيس؟

- تغير النجاح
- تغير الكلفة/الزمن
- false-success
- recovery
- context use
- transfer
- regression

أمثلة حذف

- أوقف MCD واستعمل routing ثابتاً.
- أوقف ClosureCourt واعتمد self-report.
- استبدل EVCF بـnaive context stuffing.
- امنع skill learning واستعمل raw prompts.
- امنع cross-harness routing واستعمل substrate واحداً.

إضافة إلى ablation، نستخدم Demolition Court: إذا كان المفهوم يفسر كل شيء بعد وقوعه ولا يحدد تنبؤاً قبل التجربة، يُخفض. وإذا نجح فقط لأن implementation يمنح ميزانية أكبر، يُعاد تصميم المقارنة. وإذا لم تُحفظ raw traces وenvironment fingerprint، لا تُرفع النتيجة إلى مستوى reproducible evidence.

هذا يتفق مع corpus الذي يجعل falsifiability، mechanism، وworld effect شروطاً للارتقاء النظري. [M4-M6]

النشر الإنتاجي: Windows CLI أولاً

AL RAZZI as headless core ready for future UI

المسار المقترح للإنتاج يظل CLI-first. يطلق المستخدم al-razzi run "mission"، لكن CLI لا يملك business logic؛ هو عميل لـCommand Bus. Runtime process يدير mission/state/knowledge/router، وExecution workers تعمل تحت sandbox/host adapters. هذا يسمح لاحقاً بـTUL أو Windows UI أو ACP/AG-UI دون إعادة كتابة core.

Smart setup

al-razzi setup يكتشف MCP servers، Python/Node، Git، provider keys references، Ollama/local endpoints، Hermes، sandboxes، ثم يطلب profile defaults: FAST / DEEP / CODING / VISION / COMPUTER. الإعداد ينتهي باختبار capability ولا يطلب تحرير عدة ملفات يدوياً.

في Windows يجب اختبار locked files، process-tree cancellation، CTRL+C، PowerShell، spaces/Unicode/Arabic paths، OneDrive، Git worktrees، وSQLite session database. ينبغي أن يستخدم migrations وtransactions وWAL عند الملاءمة، مع backup وcheckpoints.

للنشر الأول نوصي بثلاث modes: SAFE_LOCAL للأعمال داخل workspace؛ SANDBOXED_LONG_RUN للمهام الطويلة؛ وBOUNDED_EXTERNAL لأي أثر خارجي مع approval policy أعلى. هذه profiles لا تغير قدرات الوكيل؛ تغير authority envelope.

اختبارات الفشل والأمن

Chaos, prompt injection, tool drift and recovery

الإنتاج الحقيقي يبدأ عندما نفترض أن أجزاء النظام ستفشل. يجب حقن failures في context cache، worker، database، tool، provider، sandbox، browser state، وcheckpoint. لكل failure تصنيف يحدد retry أو alternate model/tool/harness أو rollback أو human escalation. لا retry لعملية خارجية غير idempotent دون state readback.

الفشل	الاستجابة	التحقق
Provider timeout	circuit breaker + alternate provider	no duplicate tool effects
Tool timeout	observe process + cancel/retry if safe	state fingerprint
Prompt injection in repo/web	treat as untrusted data	constitution unchanged
Context stale	invalidate cache + rematerialize	hash/freshness match
Worker crash	checkpoint resume	DAG node consistency
Wrong external effect	compensation/rollback/escalate	post-recovery readback
Self-evolution unsafe candidate	quarantine/reject	no authority elevation

المقياس الحاسم هنا هو Mean Time To Verified Recovery، لا مجرد catch exception. ويجب أن تشمل الاختبارات محاولة عبور authority عبر plugin أو MCP أو remote harness.

الخلاصة وبرنامج البحث

From an architecture-rich harness to evidence-rich Agentic AGI research

تخلص الورقة إلى أن أفضل طريق عملي لمشروع Moussaif ليس بناء model ضخم مستقل، بل بناء **حزام تشغيل معرفي** يجعل النماذج الحديثة أجزاءً قابلة للتبديل داخل نظام يملك الذاكرة والسياق والأدوات والدليل والحاكمية. Prime يبرهن قوة programmable persistent runtime؛ Hermes يبرهن قيمة memory/skills/backends؛ Codex يبرهن أهمية execution boundaries والـSDK؛ Anthropic يوضح ديناميكيات long-horizon/subagents؛ وAVO يدفع نحو variation طويلة الأفق والـlineage. ما يضيفه Moussaif هو محاولة توحيد هذه القوى حول Active Difference وKnowledge-as-OS وWorld Effect وMUT/MUG/MOUSSAIXT.

العشر نظريات المقترحة لا تُعلن كحقائق. قيمتها أنها تولد protocol وtool وmetric وablation. والخمس نظريات Agentic-AGI تعيد تعريف العمومية بصورة تشغيلية: نقل حقيقي عبر المجالات، استقلال عن substrate، إدارة المجهول، استمرارية قابلة لإعادة البناء، وإغلاق مزدوج معرفي/حاكمي.

Roadmap 90 يوماً

- الأسبوع 1-2: freeze AL RAZZI genome وتوثيق real causal paths.
- الأسبوع 3-4: real provider + tool + browser + restart tests.
- الأسبوع 5-6: EVCF stress + cross-harness adapters Hermes/Codex/Prime عند التوفر.
- الأسبوع 7-8: ablations وmatched-budget baselines.
- الأسبوع 9-10: experience compiler + shadow self-evolution.
- الأسبوع 11-12: report with raw traces and reproducible package، long-horizon endurance، canary.

النتيجة البحثية المطلوبة ليست قول "وصلنا إلى AGI"، بل إنتاج منحنى دليل: كل نسخة من harness يجب أن تكون قابلة لإثبات أنها تتعامل مع سياق أوسع، تفشل بصدق، تتعافى أكثر، تنقل مهارات أكثر، وتحقق أثراً خارجياً أدق من النسخة التي قبلها.

المراجع والمصادر - I

Internal corpus + primary upstream sources

مصادر مشروع MOUSSAIF الداخلية

[M0] AL RAZZI OS (MoussaifXT Hyper Harness CLI) - Deployment Complete. Internal deployment artifact, September 2026 يصف runtime، golden tests، modules، HyperKB indexing، registries، Zenith loop، ToolGateway، Courts.

[M1] *MOUSSAIF AGI OS / MOUSSAIXT Deep Engineering Architecture*, 2026-15 slide architecture summary. تعريف MUT = Theory Unification، MCD/QMCD 50 /pages, August 2026. 12-layer stack، authorization، MOUSSAIXT، Self-Reconstruction، World Effect، HyperSkills/Abilities/Loops، HKOS، MCAI.

[M2] *MOUSSAIF Conscious AGI Research Program - Project + Gaps*, 2026-15 slide architecture summary. تعريف MUT = Theory Unification، MOUSSAIXT = Execution Shell، MUG = Program Governance.

[M3] *MOUSSAIF Unified AGI* - الكتاب العربي الجامع للنظرية والمعمارية والتشغيل, 225 pages, 16 Aug 2026. Claim-ranked evidence ladder and unified lineage.

[M4] *AGI-Oriented Knowledge-Base as Cognitive OS*, 33 demolition court، HyperLoops، HyperSkills، Knowledge-as-OS، pages, 27 Jun 2026.

[M5] *ParallaxWorld-MCD Full AGI OS*, MCD 350 governing question، Volume Before Difference، pages, 12 Aug 2026. world model/courts.

[M6] *Open Governance-Space Theory / Reverse Capture*, 2026. Meta-governance and competing authorities as research constructs.

Upstream engineering

[U1] PrimeIntellect-ai/prime-agent, GitHub, accessed Sep 2026. Persistent IPython/RLM, recursive agents, daemon continuity, continual harness.

[U2] Prime Agent, *RLM Runtime Architecture*, GitHub docs, 2026. KernelManager, Python runtime, child lifecycle and harness state.

[U3] NousResearch/hermes-agent, GitHub and official docs, 2026. Persistent memory, skills, context files, checkpoints, cron, subagents, multiple terminal backends.

المراجع والمصادر - II

Claude · Codex · NVIDIA · baselines and protocols

[U4] Anthropic, *Prompting Best Practices - Agentic Systems*, Claude Platform Docs, accessed Sep 2026. Long-horizon state tracking، parallel tools، subagent orchestration، reversibility and context management.

[U5] OpenAI, *openai/codex*, GitHub, 2026. Terminal coding agent, CLI/app/IDE surfaces, Apache-2.0 repository.

[U6] OpenAI Codex Python SDK API Reference, 2026. Thread lifecycle، Sandbox.read_only/workspace_write/full_access، streaming /interrupt/steer.

[U7] OpenAI Cookbook, *Iterating development workflows with Codex*, 2026. AGENTS.md، goal/plan/build/context separation، reusable harness-author skills.

[U8] SWE-agent/mini-swe-agent v2, documentation, 2026. Minimal ~100-line agent baseline and SWE-style evaluation support.

[U9] Aider-AI/aider, repository map and Git integration docs, 2026. Symbol-level repo maps, automatic Git history and undo-oriented workflows.

[U10] NVIDIA Technical Blog, *NVIDIA AVO Reaches 100% on ARC-AGI-3*,... 21 /Aug 2026. Agentic Variation Operators، persistent memory supervision، long-horizon kernel optimization and public ARC-AGI-3 report.

[U11] Model Context Protocol, official specification/repository, 2026. Standardized tool/resource interoperability; treated here as an external boundary, not internal Moussaif state architecture.

[U12] Agent Client Protocol (ACP) and AG-UI protocol repositories, 2026. References for future UI/editor client interoperability.

بيان ختامي

هذه المراجع لا تجعل النظريات الجديدة حقائق مثبتة. هي توفر آليات مقارنة وقيود تصميم. صلاحية النظريات H1-H10 وA1-A5 تعتمد على البرامج التجريبية المقترحة، خصوصاً cross-domain transfer، restart/failure injection، matched-budget ablation، وتكرار مستقل.

صيغة الاستشهاد المقترحة: Moussaif, Z. (2026). *Harness Engineering for AGI-Oriented Agentic Systems: From Claude, Prime, Codex and Hermes to MUT-MUG-MOUSSAIXT*. MOUSSAIF Research Series, working paper.