نسخة أرشيفية معاد بناؤها من النص المفهرس - RECONSTRUCTED_FROM_INDEXED_TEXT

<PARSED TEXT FOR PAGE: 1 / 30>
الورقة البحثية الهندسية - النسخة الثانية - 30 صفحة
النظرية التطبيقية لـ MUG/MUT وتقنيات
الجديدة MOUSSAIF AGI
From Theory Unification and Program Governance to Knowledge-Embedded
Agentic Execution
تقدّم هذه الورقة صياغة تطبيقية مركّزة لمنظومة MOUSSAIXT · MUG · MUT داخل برنامج MOUSSAIF AGI،
، HyperKnowledge، Knowledge-as-OS، MCD/QMCD/MCAI/HAKIM-MCD - النظرية السلالة لّوتحو
HyperSkills، HyperAbilities، HyperLoops، المحاكم، WorldEffect والذاكرة التطورية - إلى تقنيات تشغيلية مترابطة
قابلة للتنفيذ داخل RAZZI AL بوصفه harness سيادياً قابلاً للربط مع نماذج ووكلاء وبيئات متعددة.
المؤلف / البرنامج البحثي
زهير مساعف - Moussaif Zouhair
الإصدار
Version 2 - September 2026
التنسيق
A4 · RTL · Amiri · Arabic + technical English
الحالة المعرفية
Applied research architecture - لا تدّعي إثبات AGI
قاعدة الادعاء: هذه الورقة تميّز بين mechanism implemented، specification engineering، theory، وcomparative validated
gain. اسم التقنية أو حجم corpus أو عدد HyperSkills لا يُعد دليلاً على AGI؛ القيمة تُثبت فقط عندما ينتج المكوّن فرقاً قابلاً للقياس تحت
baseline وضبط تجريبي.
MUT/MUG Applied Theory · MOUSSAIF AGI 1 / 30
<PARSED TEXT FOR PAGE: 2 / 30>
MUT · MUG · MOUSSAIFXT · AL RAZZI · APPLIED RESEARCH · 2026 التطبيقية والأطروحة الملخص
الملخص والأطروحة التطبيقية
Abstract and applied thesis
المشكلة
الأنظمة الوكيلية الحديثة تستطيع الاستدلال واستدعاء الأدوات والبرمجة، للكنها غالباً تبقى موزعة بين
prompt، memory، tools، state، evaluators
وruntimes specific-provider. مشروع MOUSSAIF يقترح معالجة مختلفة: ليس بناء وكيل واحد أكبر، بل بناء نظام تشغيل معرفي-حاكمي يحتفظ
بالهدف، السؤال الحاكم، الفرق النشط، المعرفة، السلطة، أثر العالم والذاكرة فوق النموذج وتحتفظ به عبر تبدّل النماذج والبيئات.
في هذا البناء، MUT هو طبقة توحيد النظرية: يربط النظريات والعلاقات والأسئلة والأدوات ومرشحات التجربة. MUG هو طبقة حاكمية البرنامج: يضمن
أن قواعد البحث، internalization، behavior runtime، الأدلة والتطور لا تتناقض فيما بينها. أما MOUSSAIXT فهو shell execution عابر
هذه ليّفع agentic harness كل AL RAZZI ويعمل .local/cloud runtimes، verification، memory، tools، agents، models يربط للبيئات
العقود في runtime/CLI قابل لاحقاً للUI.
الأطروحة
تطرح الورقة أن القيمة الجديدة لا تأتي من مصطلح منفرد، بل من سلسلة سببية مغلقة: نظرية موحّدة في MUT تصبح policy أو mechanism تحت
MUG؛ ثم تُترجم إلى objects Capability/Knowledge؛ ثم يفعّلها MOUSSAIXT داخل بيئة تنفيذ؛ ثم يراقب RAZZI AL الأثر ويعيد

الدليل إلى
الذاكرة والتطور. إذا انكسرت أي حلقة تصبح النظرية وصفاً لا تقنية.
→ MUT → MUG → PEK/HyperCapability → MOUSSAIXT → World Effect
Evidence → Evolution
المساهمة: صياغة تقنيات تطبيقية قابلة للاختبار فوق corpus الحالي: Routing Weighted، Compiler Question-Governing، EVCF-
/Difference
MCAS، Evidence-Carrying Action، World-Effect Transaction، HyperCapability Closure Graph، HarnessGenome،
Evolution
.Cross-Harness Sovereign Routingو

<PARSED TEXT FOR PAGE: 3 / 30>

المنهج وسُلّم الإثبات
Method and evidence discipline
تعتمد هذه الورقة على المراجع المعيارية داخل corpus المشروع: المرجع الموحّد للAGI MOUSSAIF، تقرير MOUSSAIXT الهندسي،
02AGI-MUT-،
خرائط المشروع، ومواد OS-as-Knowledge. المنهج ليس تلخيصاً أرشيفياً ؛ بل translation runtime-to-research: لكل مفهوم نسأل عن المدخلات، الحالة، القرار، المخرجات، الفشل، التحقق والاختبار المقارن.
مراتب الحالة
الرتبة المعنى في هذه الورقة ما يلزم للترقية
0E غير متحقق أو مصدره غير مسترجع مصدر أو artifact موثوق
1E مذكور تاريخياً أو كهدف artifact قابل للفحص
2E موجود في corpus عقود تنفيذ واضحة
3E محدد هندسياً / قابل لإعادة البناء تشغيل فعلي محفوظ
baseline + matched budget + CI ومكرر مضبوط محلي اختبار E4
5E مقارنة مستقلة عابرة للنموذج/المجال Replication خارج البيئة الأصلية
قاعدة الهدم
كل تقنية جديدة في هذه الورقة يجب أن تملك ablation. إذا حذفناها ولم يتغير completion verified أو reliability أو efficiency context أو
recovery، فهي إما غير ناضجة أو زائدة معمارياً . ولا يُسمح بالاستدلال من وجود classes أو diagrams إلى وجود قدرة حقيقية.
على تتقدم Truth: observed state + explicit authorization + provenance-linked evidence + independent verification
ordering
.المخرجات كثافة أو memory، self-report، model confidence

<PARSED TEXT FOR PAGE: 4 / 30>

البنيوي الثالوث: MUT · MUG · MOUSSAIXT
The three-system family
يعطي corpus للمشروع ثلاث بنى تكمل بعضها. الأولى Unification Theory - MUT: توحّد lineage النظريات والروابط والأسئلة والأدوات ومرشّحات التجربة. الثانية Governance Program - MUG: تمنع انفصال البحث عن التنفيذ، وتبقي internalization وbehavior runtime وevidence rules داخل حاكمية واحدة. الثالثة Shell Execution - MOUSSAIXT: تجسد النظرية والحاكمية في بيئات تشغيل متعددة.
MUT
Theory Graph → MUG
Governance Compiler → MOUSSAIXT
Execution Fabric → AL RAZZI

Harness Runtime
لماذا الفصل مهم؟
إذا اندمجت الطبقات الأربع في prompt واحد، يصبح التعديل خطراً: تغيير صياغة قد يغيّر الحاكمية والتنفيذ والاختبار معاً. أما الفصل فيسمح بأن تبقى
نظرية MCD مثلاً داخل MUT، وأن تتحول في MUG إلى policy قابلة للضبط، وأن تستعملها MOUSSAIXT لتخصيص context/tool/ provider،
وأن يسجل RAZZI AL أثر القرار في trace دون أن يغير النظرية الأصلية.
المستوى السؤال المخرج
MUT ما الذي نعتقد أنه آلية ذات قيمة؟ graph Mechanism/Theory
MUG متى يسمح بتفعيلها وبأي قيود؟ policy Governance
MOUSSAIXT كيف تنفذها؟ وأين تنفذها؟ Host/Tool/Model route
RAZZI AL هل نجحت المهمة فعلاً ؟ WorldEffect + Evidence + Trace

<PARSED TEXT FOR PAGE: 5 / 30>

MUT: من خريطة نظرية إلى graph قابل للحوسبة
AGI-02
02AGI-MUT-: من خريطة نظرية إلى graph قابل للحوسبة
Theory unification as an executable graph
تصف مواد 02AGI-MUT- سجلاً واسعاً يجمع cognition، knowledge، agency وevidence، مع 42 نظرية أساسية، 20 constructs legacy،
خمس نظريات طبقية، graph من نحو 200 عقدة، 100 أداة تشغيلية و19 HyperKnowledge bases بوصفها scale registry موثقاً لا benchmark لقدرة النظام. القيمة العملية لهذا السجل ليست العدد، بل إمكانية تحويله إلى dependency graph يمنع الاستعمال العشوائي للمفاهيم.
البنية المقترحة لعقدة MUT
TheoryNode {
id, version, status, scope,
mechanism, assumptions, predictions,
required_evidence, falsifiers,
dependencies, conflicts,
implementation_targets,
benchmark_candidates
}
هكذا تصبح العلاقة بين MCD وQMCD وMCAI وMCD-HAKIM علاقة قابلة للفحص: MCD يحدد فروقاً ، QMCD يعيد وزنها بالسؤال، MCAI
يدمج القيود والذاكرة والأدوات، MCD-HAKIM يمرر الدعوى عبر verification/action/critique/evidence. إذا كان التنفيذ لا يستعمل هذه
العلاقات فلا يحق لنا القول إن MUT صار runtime.
جديدة تقنية: Theory-to-Mechanism Compiler
نقترح طبقة TMC تحول TheoryNode إلى ثلاث مخرجات: PolicyCandidate داخل MUG، HyperCapabilityCandidate داخل runtime، وExperimentCandidate داخل registry benchmark. هذه الطبقة تجعل كل نظرية مطالبة بجسر إلى العمل لا بمجرد تعريف.
Theory Value = Mechanism × Falsifiability × Runtime Relevance × Evidence Quality

<PARSED TEXT FOR PAGE: 6 / 30>

MUG: الحاكمية كطبقة تشغيل لا كوثيقة سياسة
MUG: سياسة كوثيقة لا تشغيل كطبقة الحاكمية

Program governance as runtime control

MUG ليست »مجموعة قواعد« توضع في prompt system. وظيفتها أن تجعل الحاكمية قابلة للإنفاذ: تتلقى intents، knowledge goal، claims، tool

بهذا promotion gates. وreversibility، evidence requirements، provenance، authority تطبق ثم evolution، candidates patches

تصبح الحاكمية جزءاً من runtime وليس تذكيراً لغوياً للنموذج.

خمسة registers داخل MUG

Register Constitution: الحدود غير القابلة للتجاوز، مثل عدم مساواة call tool بلeffect world.

Register Authority: من يملك فعل ماذا وتحت أي scope.

status :Epistemic Registerللclaims، freshness، source rank، contradiction.

shadow/canary/promotion. إلى proposal من candidate lifecycle :Evolution Register

lineage، duplicate/conflict control، canonical resource identity :Integrity Register.

الميزة الجديدة هنا أن MUG لا تحكم فقط actions؛ بل تحكم تحول المعرفة نفسها. KnowledgePatch قد يكون metadata harmless أو قد يغير سياسة

routing؛ لذلك نحتاج class risk قبل promotion.

نوع التغيير الخطر Gate مقترح

schema + provenance منخفض Knowledge metadata

sandbox + tests + court متوسط HyperSkill

held-out benchmark + canary عال Router/Loop

independent review + human approval حرج Authority/Constitution

•

•

•

•

•

MUT/MUG Applied Theory · MOUSSAIF AGI 6 / 30

<PARSED TEXT FOR PAGE: 7 / 30>

MUT/MUG · 2026 · APPLIED RESEARCH · AL RAZZI · MOUSSAIFXT · MUG · MUT تطبيق pipeline: الفعل إلى النظرية من

من النظرية إلى الفعل: pipeline تطبيق MUG/MUT

Applied theory-to-action pipeline

التحول الحاسم هو بناء مسار يجعل النظرية قابلة لأن تغيّر القرار ثم يُقاس أثرها. نقترح pipeline موحداً:

→ Raw Goal → Mission Contract → MUT Mechanisms → MUG Gates → PEK / HyperAbility

MOUSSAIXT Action → WorldEffect

Contract Mission يثبت الهدف والقبول والقيود. MUT يرشح آليات مثل MCD أو HyperKnowledge. MUG يحدد أيها مسموح ومتى، ثم PEK

compiler يبني context working. بعد ذلك MOUSSAIXT يختار tool/model/host ويجمع evidence. إذا لم يتغير state observed كما هو

مطلوب، تعاد المهمة إلى Difference Active بدلاً من closure وهمي.

جديدة تقنية: Governed Theory-to-Action Compiler - GTAC

GTAC هو compiler منطقي بين النظرية والتشغيل. مدخلاته ProjectState + MissionContract + TheoryNodes؛ ومخرجاته RuntimePlan

يستطيع trace أن الفائدة ablation tags. وmechanism activations، context requirements، tool authority، verification depth يحتوي

القول: »تم تفعيل MCD لأن risk/uncertainty تجاوز العتبة«، وليس »النموذج فكر أكثر«.

شرط العلمية: كل activation يحمل flag counterfactual يسمح بتشغيل المهمة لاحقاً دون الآلية نفسها. بهذه الطريقة يصبح ablation جزءاً

من التصميم
وليس تجربة يدوية متأخرة.

<PARSED TEXT FOR PAGE: 8 / 30>

MCD :المعرفية النواة
MCD · QMCD · MCAI · HAKIM-MCD :المعرفية النواة
The applied cognitive kernel
تعامل الورقة هذه السلسلة كأربعة أدوار مترابطة لا كأسماء مترادفة. MCD ينتقي الفرق النشط الذي يستطيع تغيير القرار. QMCD يضع هذا الفرق تحت
السؤال الحاكم ويعيد وزن الخيارات وفق MCAI .need decision يدمج reasoning، memory، tools، uncertainty، constraints وstate.
أما
HAKIM-MCD فيضيف مسار الحكم: claim → evidence → critique → action/experiment → verification.
الآلية مدخلها قرارها المقياس
MCD نخصص أين gain per compute compute differences + risk + cost
QMCD question governing ما الفرق الذي يهم فعلاً relevance decision
MCAI الموارد ندمج كيف multi-source state coherence + recovery
HAKIM-MCD الفعل/بالحكم نسمح متى claims + evidence false-success reduction
من heuristic إلى measurable policy
لكل runtime يسجل أن نقترح route: uncertainty، risk، expected information gain، reversibility، context cost، evidence
weakness، severity gap-remaining. ثم ينتج Score لا ليكون »قانون ذكاء«، بل policy قابلة للمعايرة.
$$S = w_1\Delta + w_2U + w_3R + w_4I + w_5G + w_6E - w_7C$$
Δ = active difference، U = uncertainty، R = risk، I = information gain، G = gap severity، E = expected world-effect
حيث
value، C = compute/context cost.

<PARSED TEXT FOR PAGE: 9 / 30>

:1 تقنية
:1 Difference-Weighted Cognitive Routing تقنية
DWR - compute where difference matters
DWR (Routing Weighted-Difference) هو تحويل مباشر لمبدأ »الفرق قبل الحجم«. بدلاً من إعطاء كل node أو document أو agent
نفس
الحصة، يُوزع compute حسب أثر الفرق المحتمل. مثلاً: ملفان متشابهان بنسبة 99% قد يحمل اختلاف 1% يغيّر security deployment؛ أما
100
صفحة إضافية متطابقة دلالياً فلا تستحق context.
خط التنفيذ
.observed state وdesired state بين differences استخراج
ربط كل difference بقرار أو criterion acceptance.
.uncertainty/risk/information gain تقدير
.resource: retrieval، tool، verifier، subagent، deeper model أو experiment اختيار
إيقاف المسار عندما يصبح gain marginal أقل من threshold cost.
داخل verification depth، Mission DAG داخل agent spawning، EVCF داخل context paging يحكم أن DWRلل يمكن :الجديدة
التقنية
Courts، وحتى selection provider داخل MOUSSAIXT. أي أن MCD لا يبقى heuristic reasoning بل يتحول إلى vascularity
resource

controller.
اختبار الحذف
نقارن DWR مع routing budget-equal وrouting heuristic static على مهام طويلة. نثبت model والtool set والميزانية. النجاح لا يقاس بطول
DWR يتفوق لم إذا. recovery after injected failure، tokens per verified delta، time-to-diagnosis، verified completionبل
بل ،التفكير
في المهام عالية التباين، يُخفض ادعاؤه.
1.
2.
3.
4.
5.

<PARSED TEXT FOR PAGE: 10 / 30>

تقنية :2 محرك السؤال الحاكم
Governing Question Engine
السؤال الحاكم ليس إعادة صياغة prompt، بل محاولة العثور على السؤال الذي إذا حُل قلل أكبر مساحة من الخطأ أو أغلق أكبر عدد من الفروع.
في المهام
المعقدة قد يكون سؤال المستخدم سطحياً: «اصلح التطبيق»، بينما السؤال الحاكم فعلياً «هل الفشل من transition state أم timeout
provider؟».
الإجابة على الثاني تغيّر خطة العمل أكثر من إنتاج code مباشرة.
GQE - Governing Question Engine
نقترح وحدة تولد QuestionCandidates وتربط كل واحد بلexpected power diagnostic، cost evidence، actionability
وfalsifiability.
QMCD يقوم بالترتيب، ثم يجرب النظام السؤال الأعلى أو يجمع evidence رخيصاً قبله.
QuestionCandidate قيمة معلومات الكلفة قرار
هل الخطأ في token auth؟ عالٍ منخفضة اختبار مباشر
هل نعيد كتابة backend؟ غير محددة عالية يؤجل
هل fixture test قديم؟ متوسط/عالٍ منخفضة الفحص diff
يتم حفظ سلسلة الأسئلة في trace. إذا كان السؤال الحاكم لا يسرع اكتشاف السبب أو لا يخفض calls tool فهو مجرد بلاغة. ويمكن ربطه
بلTycho-
؟ بسيطا observation أم جديدا model/world representation نحتاج هل :like abstraction decision
Q* = argmax(Q) [DiagnosticPower × Actionability × Falsifiability / Cost]

<PARSED TEXT FOR PAGE: 11 / 30>

تشغيل كطبقة Knowledge-as-OS: HyperKB
From retrieval store to operating knowledge
في MOUSSAIF لا يُقصد بلHyperKB تجميع وثائق كثيرة، بل تحويل المعرفة إلى وحدات تحمل
constraints، provenance، relations، triggers،
skills وboundaries failure. المرجع الداخلي للOS-as-Knowledge يضع معياراً واضحاً: المعرفة لا تقترب من مستوى AGI لأنها كبيرة، بل
لأنها تتحول
إلى عمليات قابلة للنقل والهدم والتحقق.
HyperKnowledge Unit
{ HKUnit

id, version, claim_type,
definition, mechanism,
evidence_refs, counterevidence,
scope, freshness,
relations,
triggers,
skills, tools,
tests, failure_boundary
}
بهذا يمكن للruntime أن يسأل: »ما المعرفة المرتبطة بهذه signature failure؟« أو »أي HyperSkill يحتاج هذه النظرية؟« بدلاً من البحث النصي فقط.
GraphRAG يصبح جزءاً من fabric retrieval وليس المعمارية كلها.
جديدة تقنية: Knowledge Activation Contract
لكل unit عقد activation يحدد متى يدخل PEK، وما الذي يسمح له بتغييره. بعض descriptive knowledge؛ بعضه constraint؛ بعضه skill
instruction؛ وبعضه rule evidence. الفصل يمنع أن تتحول وثيقة قديمة أو theory speculative إلى policy تنفيذية من غير gate.
الهدف التطبيقي: تحويل corpus إلى readable-machine, versioned, assets ready-ablation. هذه هي الفجوة المعلنة في خرائط المشروع، وليست إضافة
تجميلية.

<PARSED TEXT FOR PAGE: 12 / 30>

MCAS / EVCF: السياق الافتراضي كطبقة ثورية
Extensible Virtual Context and Cognitive Address Space
أهم تقنية جديدة في مسار RAZZI AL هي الفصل بين window context model وcontext addressable system. EVCF يحتفظ بموارد أكبر
بكثير. MCAS يوسع ذلك إلى: repositories، tables، HyperKB، browser/computer states، artifacts، traces وagent outputs.
أيضاً للعنونة قابلة world-model objects وevidence، hypotheses، goals ليجعل
Model Context
· Resident Set ↕ Context Governor ↕ EVCF / MCAS ↔ Repo · KB · DB · Agents
World State
ليست نافذة لا نهائية
الادعاء الصحيح هو context virtualized: النظام يختار pages، يعمل context-to-compute، يحتفظ handles، ويعيد materialization حسب
الحاجة. المورد الكبير قد يبقى في DuckDB أو index AST أو store artifact، والنموذج يرى result query صغيراً ذا provenance.
ctx://repo/symbol/AuthService.verify
ctx://kb/mcd/difference-routing
ctx://evidence/mission/42/test-17
ctx://computer/session/9/window/active
cog://hypothesis/H-003
cog://world/entity/provider-7
مقياس النجاح
datasets على النجاح ومعدل tokens، useful-page ratio، cache hit، context fault rate، virtual bytes، resident tokens، avoided نقيس
أكبر من context. إذا لم يتفوق EVCF على RAG ordinary أو baseline style-RLM في efficiency أو reliability، نعيد هندسته.

<PARSED TEXT FOR PAGE: 13 / 30>

تضخم بلا PEK: Knowledge-Embedded GigaPrompts
Prompt-Embedded Knowledge Compiler
PEK inference لكل يبني compiler هو المقصود .system prompt داخل 500KB وضع تعني لا Knowledge-embedded
GigaPrompts
لازمة مجموعة أقل من: constitution، MissionContract، GoverningQuestion، ActiveDifference، relevant project state،
Capsule
HyperKnowledge، HyperAbility/Skills، HyperLoop state، ToolContracts، evidence، memory وoutput schema.
PEK Trace
كل capsule يجب أن يسجل لماذا دخلت كل وحدة، cost token، freshness، authority وhash. هذا يسمح لاحقاً بتجربة
counterfactual: هل
أدى إدخال X knowledge إلى تحسن أم زاد confusion؟
طبقة وظيفتها
0P Constitution ثوابت السلطة والنزاهة
1P Mission الهدف وشروط القبول
2P Question/Δ السؤال والفرق الحالي
P3 Knowledge/Capability HKUnits + skills/ability/loop
P4 Tools/Evidence contracts + observations
P5 Output الاستجابة ودور schema
جديدة تقنية: Context Utility Learning
يحدث runtime statistics لكل unit: invoked، later-used، verifier-supported، caused-regression، tokens. ينتج ثم
ContextPolicyCandidate يُختبر في shadow. هكذا يصبح engineering prompt جزءاً من evolution-self القابلة للقياس.

<PARSED TEXT FOR PAGE: 14 / 30>

HyperSkill: المعرفة كقدرة ذرية قابلة للاختبار
Executable skill engineering
HyperSkill ليست persona ولا تعليمات مطولة؛ هي contract capability قابلة للاختبار. يجب أن تملك trigger، schema input،
preconditions، knowledge requirements، tools، permissions، procedure، output، postconditions، evidence، failure،
modes
recovery وrollback. بهذا يمكن للمهارة أن تنتقل بين agents والنماذج دون فقد معناها.
الفرق عن skill التقليدية
HyperSkill تقليدية Skill
typed contract + executable hooks Markdown/procedure
تملك لا evidence requirements evidence policy
أداء تاريخ لا runtime metrics + version lineage
قد تزيد السلطة ضمنياً ceiling authority ثابت
جديدة تقنية: Skill Causal Ledger
skill quality يجعل هذا .verifier result وbefore state، cost، evidence، observed delta، expected delta يسجل HyperSkill
استدعاء كل
خالفت أو side effects تكررت إذا المهارة quarantine يمكن .cost وregression_rate، transfer_gain، verified_success_rate: للقياس
قابلة
contract.

قاعدة عامة: تعلم مهارة جديدة لا يمنح سلطة جديدة. improvement Capability =/= increase permission.

<PARSED TEXT FOR PAGE: 15 / 30>

تركيب :HyperAbility
HyperAbility: تركيب القدرات بدلاً من تضخم الوكلاء
Composed capability layer
HyperAbility تقع بين HyperSkill وMission. هي reusable composition مثل »إصلاح feature كاملة«، »بحث علمي موثق«، أو »تحقق database-to-browser«. قوتها أنها تفصل ما يجب أن يتحقق عن من ينفذه: نفس HyperAbility يمكن أن تعمل عبر native agent أو Hermes أو
Codex إذا كانت العقود متوافقة.
HyperCapability Closure Graph
نقترح تقنية HCCG التي تمثل ability كـgraph من skills + knowledge + tools + verification edges. لا تعتبر ability مكتملة إلا إذا كل مسار
justification. أو verifier/rollback له effect
Discover → Plan → Edit → Test → Verify → Close
يمكن للHCCG أن ينتج dynamically من DAG Mission. إذا ظهر pattern نفسه بنجاح عبر مهام متعددة يصبح HyperAbilityCandidate، ثم يمر
Court Generalization قبل التسجيل.
المقياس
إذا ،recovery ،reuse ability، models عبر portability، coordination overhead، completion gain نراقب .skills عدد ليست ability قيمة
كانت planning dynamic مباشرة أفضل من ability محفوظة في معظم المهام، يجب تقليل عدد abilities الثابتة.

<PARSED TEXT FOR PAGE: 16 / 30>

HyperLoop: التكرار المحدود وإعادة فتح الفجوة
Bounded recurrence and closure discipline
HyperLoop/ success، watchdog، budgets، transition rules، entry condition، while-true agent. يملك وليس محدد state machine هو
HyperLoop هو المحوري النمط .recovery و failure/stop، checkpoint Observe → Difference → Retrieve/Plan → Act → Readback → Verify →
Reopen Gap.
↺ OBSERVE → Δ → PLAN → ACT → READBACK → VERIFY
Closure-Constrained Loop :جديدة تقنية
لا يتوقف loop لأن model قال »تم«. ينشئ RemainingGap][من Mission Original مقابل State Observed. ClosureCourt يقرر COMPLETE فقط عندما لا يبقى gap إلزامي غير مبرر ويكون WorldEffect موثقاً .
حالة القرار
Gap قابل للإصلاح REPAIR
الخطة خاطئة REPLAN
provider/tool فشل RECOVER/FALLBACK
authority ناقصة WAIT_FOR_HUMAN
معايير القبول كلها محققة CLOSE
هذا loop يترجم مبدأ MUG بأن الإغلاق قرار حاكمي مبني على evidence لا property داخل message.

<PARSED TEXT FOR PAGE: 17 / 30>

Mesh HyperAgent: الوكلاء كموارد معرفية محدودة

Recursive delegation and agent mesh

ليس persona. هو goal scope + knowledge scope + HyperAbilities/Skills/Loops + model/host policy + tools + HyperAgent

.والقياس للتبديل قابلة وحدة agent يصبح بهذا .memory + evidence contract + budget + authority ceiling

أنماط mesh

.الهندسية للمهام Planner → Executor → Verifier

.للبحث Researcher → Synthesizer → Critic

Candidate A/B/C → Court عند QMCD uncertainty.

للمهام طويلة الأفق. workers bounded → Supervisor

كل يحتاج لا child trace في context. يكفي ResultContract: status، findings، evidence، artifacts، unresolved gaps، parentال

persistent cognition. من EVCF ويقرب context pollution يقلل هذا .recommendation

جديدة تقنية: Agent Population Controller

لا نزيد agents تلقائياً. spawn يتم إذا benefit independence/parallel expected أكبر من cost coordination. وتُسجل لكل نوع agent بيانات

نجاح/كلفة. وبذلك يصبح agent-multi قراراًمحكوماً بلMCD لا شعاراً.

Spawn if E[Information or Parallel Gain] > Coordination + Context + Cost

-
-
-
-

<PARSED TEXT FOR PAGE: 18 / 30>

MOUSSAIXT: التنفيذ العابر للبيئات

Cross-environment execution shell

MOUSSAIXT هو الطبقة التي تجعل Moussaif غير مربوط بموديل أو harness واحد. يحتفظ النظام بعقود مستقرة بينما تتغير hosts. HostAdapter

Native تشمل أن للبيئات يمكن .recover، cancel، checkpoint، observe، execute، capabilities، health، discover يملك النموذجي MCP.و local models، Prime Agent، OpenCode، Claude Code، Codex، Hermes، Windows

Cross-Harness Sovereign Routing - CHSR

CHSR: router ليس يختار model، فقط بل execution substrate وفق capability fit، reliability، context capacity، privacy، نقترح

Moussaif-owned. يبقى global Mission state .historical verified success ،tool support ،latency ،cost

Node ممكن أن ينفذه سبب الاختيار

persistent programmable context Native/Prime repo exploration

worktree + code tooling Codex/OpenCode coding patch

context/agent specialization Claude/Hermes research verification

privacy/locality Native + local model local private data

سيادة الحالة: harness external لا يملك closure أو truth memory. يُعامل كل executor bounded يعيد evidence/artifacts/ observations إلى

AL RAZZI.

<PARSED TEXT FOR PAGE: 19 / 30>
AL RAZZI: harness يربط النظرية بالتشغيل MUT · MUG · MOUSSAIFXT · AL RAZZI · APPLIED RESEARCH · 2026
RAZZI AL: harness يربط النظرية بالتشغيل
The Moussaif-native production harness
RAZZI AL هو نقطة التفعيل العملي للطبقات السابقة. الCLI مجرد client، runtime core يمتلك Mission، state، Genome Context، Windows UI أو TUI إلى التحويل يجعل الفصل هذا .checkpointsو Evidence، ToolGateway، routing، capabilities، HyperKB، Courts
لاحقاً مسألة client فوق API Event/Command وليس إعادة كتابة.
Runtime causal chain
Goal → Mission → Δ/Q → PEK → Ability/Loop → Route → Action → Evidence
الharness يجب أن ينتج events حقيقية: mission.created، context.loaded، skill.activated، model.selected، tool.started، artifact.changed، evidence.captured، court.completed، world_effect.observed، checkpoint.created. UI أو JSONL هذه يقرأ الأحداث للكنه لا يصنعها.
Causal Runtime Trace :جديدة تقنية
كل transition يحمل why: لماذا استدعينا knowledge؟ لماذا agent spawn؟ لماذا استخدمنا tool؟ ما evidence الذي أغلق node؟ هذا trace
يمكن تشغيله لاحقاً في ablation/replay، فيتحول observability إلى مادة علمية وليست logging.
MUT/MUG Applied Theory · MOUSSAIF AGI 19 / 30
<PARSED TEXT FOR PAGE: 20 / 30>
Evidence-Carrying Action MUT · MUG · MOUSSAIFXT · AL RAZZI · APPLIED RESEARCH · 2026 tool call إلى
ToolGateway: من
Evidence-Carrying Action
ToolGateway: من tool call إلى
Universal tool calling under governance
ToolIntent إلى كلها يحولها أن يجب AL RAZZI للكن ؛computer action أو shell request، MCP call، function call يولد قد providerالل
موحد يمر عبر authority، policy، sandbox ثم verifier. لا يسمح للnative-provider tooling بتجاوز Moussaif.
Evidence-Carrying Action - ECA :جديدة تقنية
كل action تحمل قبل معها التنفيذ: intent، expected effect، side-effect class، authority scope، verification methodو rollback/compensation. وبعد تولد ToolObservation مع structured result + evidence refs + observed effect candidate.
ECA {
intent, expected_delta,
effect_class,
authority,
pre_state,
execute,
post_observation,
verifier,
evidence,
rollback
}
تصنيف الأثر
Class مثال متطلب
READ_ONLY ملف قراءة trace
REVERSIBLE_WRITE تعديل checkpoint + diff branch
COMPENSATABLE API update قابل للعكس compensation plan

strong authority + readback external send/delete IRREVERSIBLE

<PARSED TEXT FOR PAGE: 21 / 30>
 فقط البكسل لا الحالة على الفعل :Computer/
Browser Use
Use Browser/Computer: الفعل على الحالة لا البكسل فقط
State-grounded computer use
Use Computer داخل Moussaif يجب ألا يعني سلسلة clicks مرئية فقط. النظام يفضل surface الأكثر تحديداً: CLI application →
→API
filesystem/database → DOM → accessibility/UI Automation → visual. يقلل هذا ambiguity ويجعل verifier قادراً على
interaction
فحص state.
ComputerEnvironmentAdapter
observe، capture، act، verify، checkpoint، reset. تجمع أن يمكن الملاحظة screenshot + DOM + accessibility tree + active
يوحد
state interfaces. توجد لا عندما fallback تصبح window + focused control. visual perception
State-Grounded Action Loop: جديدة تقنية
Observe State → Choose Lowest-Level Reliable
Interface → Act → Re-observe → Verify Delta
كل click يحمل effect expected؛ إذا لم يتحقق يعاد observation بدل اعتبار success command. وتدخل prompts أو التعليمات الظاهرة
على
webpage كلdata untrusted، لا كسياسة قادرة على تغيير mission.
صلة computer/browser :MUG injection prompt يثبت الحاجة إلى حاكمية مستقلة عن محتوى البيئة؛ content world قد يصف العالم
للكنه لا يملك
حق إعادة تعريف الدستور أو الصلاحيات.

<PARSED TEXT FOR PAGE: 22 / 30>
 حالةكتحول النجاح تعريف :WorldEffect
Transaction
Transaction WorldEffect: تعريف النجاح كتحول حالة
From output to verified external effect
أحد أقوى مبادئ MOUSSAIF هو أن output model ≠/ effect world. لذلك نقترح توحيد كل مهمة ذات أثر في WorldEffect
snapshot :Transaction - WET قبل، intended delta، actions، observed after، evidence، verification ثم commit/repair/
rollback.
WorldEffectTransaction {
before_fingerprint,
intended_delta,
actions[],
observed_after,
observed_delta,
evidence[],
verdict,
compensation
}
أمثلة
برمجة: تغير الملف + build نجح + tests + UI/service readback.

قاعدة بيانات: record موجود + schema صحيح + query بعد التنفيذ.
Knowledge: artifact مسجل + provenance + retrieval عليه يعثر.
Browser: form state + تغير confirmation/readback في المصدر النهائي.
هذا يعيد تعريف closure: ليست آخر رسالة LLM، بل نجاح transaction. ويمكن للMCD تحديد depth verification: task only-read ربما يكفيها
human gate. وربما independent readback يحتاج receipt، high-risk external effect
∧ Success = AcceptanceCriteria ∧ VerifiedObservedDelta
NoCriticalUnresolvedGap
•
•
•
•

<PARSED TEXT FOR PAGE: 23 / 30>

Courts: الحاكمية المعرفية كطبقات تقييم
Evidence, falsification and closure courts
المحاكم لا ينبغي أن تكون personas تقوم بنقد لغوي متكرر. هي evaluators لها contracts/inputs وقوة مختلفة. Court Evidence يطابق claims
يراقب Architecture Court، السابقة checks يعيد counterexample، Regression Court عن يبحث Falsification Court ،بالشواهد .الإغلاق يقرر Closure Courtو ،lineage/data يحرس invariants، No-Loss Court
Adaptive Court Economy
لا نُشغل كل المحاكم لكل operation. depth يحددها risk، irreversibility، uncertainty، importance artifact وexternal effect.
هذا مهم كي
لا يصبح governance overhead أكبر من cognition.
Courts Verification Risk
Evidence light schema/readback Low
Evidence + Regression tests + verifier Medium
Falsification + Security + Closure independent verifier + checkpoint High
Court Evidence Bundle :جديدة تقنية
هكذا. world-effect، remaining gaps، evidence، tests، artifacts، claims، mission criteria: ثابت bundle بنُي، raw chat تمرير منً بدلا
تمنع reasoning producer من verifier biasing، ونحصل على replay قابل للمقارنة.

<PARSED TEXT FOR PAGE: 24 / 30>

الذاكرة: من trace إلى خبرة قابلة لإعادة البناء
Reconstructive and experiential memory
MOUSSAIF يفرق بين context المؤقت والذاكرة. يجب فصل WorkingMemory، MissionMemory، ProjectMemory، ProceduralMemory، ErrorMemory، HyperKnowledge وRuntimeState. المخبرة طويلة الأجل لا تُكتب لأن model قال »تعلّمت«، بل لأن
trajectory مر عبر extraction وتحقق.
ExperienceCompiler
Trajectory → Outcome + Evidence → Lesson Candidate → Generalization Court → Knowledge/Skill Patch
يستخرج failure signatures، successful recovery، routing mistakes، missing knowledge، unnecessary context، الcompiler

repeated human corrections. يولد ثم candidates لا production mutations.
Self-Reconstruction Memory
المعنى هنا أن النظام يستطيع إعادة بناء »لماذا وصل إلى الحالة الحالية«: decisions، versions، parents، evidence، checkpoint. هذه الذاكرة أهم
من تلخيص conversation لأنها تسمح بإعادة تشغيل mission بعد restart وبدراسة أثر تغيير واحدة.
knowledge Negative: يتم حفظ الطرق الفاشلة الموثقة مع scope، حتى لا يعيد النظام تجربة ends dead المعروفة بلا داع، للكن لا تُعمم failure
واحدة على كل المجالات.

<PARSED TEXT FOR PAGE: 25 / 30>

المحكوم والتطور Self-Evolution: HarnessGenome
Governed self-evolution beyond prompt tweaking
التطور الذاتي في المشروع يجب أن يعمل كلlaboratory research منفصل عن production. نقترح HarnessGenome يمثل configuration نشطاً :
ينتج تحسين أي .recovery، memory، verification، skill/ability/loop set، model/harness policy، routing، context policy
.evaluationو hypothesisو parent له CandidateGenome
الخط التطوري
→ Production Genome → Candidate → Shadow/Replay → Held-out Bench → Court → Canary
Promote/Rollback
تعديل تستطيع لا للكنها ،code أو model choice، loop ordering، skill composition، routing، prompt/context تطال أن variation يمكن
oracle benchmark أو root authority التي تقيمها. هذا الفصل يمنع tampering harness.
Evolution Lineage Graph :جديدة تقنية
Pareto front نحتفظ .status، cost، failures، evidence، score vector، hypothesis، mutation_type، parent_ids يسجل كل candidate
بدلاً من إجبار كل شيء في score واحد، قد يكون candidate أسرع وآخر أكثر موثوقية.
Promote only if ΔHeldOut > 0 AND SafetyRegression = 0 AND
EvidenceSufficient

<PARSED TEXT FOR PAGE: 26 / 30>

MUG الأمني: النزاهة والسرية وعدم الازدواج
Integrity, security and zero-duplication governance
نتعامل MUG مع الأمن والنزاهة كجزء من المعمارية لا كملحق. المشروع يملك corpus كبيراً ومتعدد النسخ؛ لذلك identity لا تعتمد filename وحده،
بل logical id + provenance + hash content + semantic lineage. duplicate ليس دليلاً إضافياً ، والمرآة لا تُعد مصدراً مستقلاً .
Resource Identity
ResourceIdentity = logical_id + provenance + hash + version + lineage
،SAME، VERSION_OF، DERIVED_FROM، MIRROR_OF، BACKUP_OF، SUPERSEDES.
العلاقات: CONFLICTS_WITH
أثناء integration، deduplication destructive غير مطلوب؛ الهدف تقليل ambiguity semantic.
Secure runtime
.PEK/logs داخل raw keys من ً بدلا secret references
.المكتشفة hooks/plugins تشغيل قبل project trust

approval policy. عن منفصلة sandbox policy
effect class + authority gate لكل tool/computer action.
event/log redaction وللمفاتيح tokens/cookies.
قاعدة MUG: تحسين القدرة لا يغيّر السلطة. Agent أو HyperSkill جديد يحتفظ ceiling authority الممنوح له؛ ولا يسمح evolution-self أن يرفع
صلاحياته تلقائياً .
•
•
•
•
•

<PARSED TEXT FOR PAGE: 27 / 30>

مهمة :التطبيق
التطبيق: مهمة هندسية كاملة من MUT إلى WorldEffect
End-to-end production mission example
نفترض المهمة: »اصلح authentication integration في repository حقيقي، شغّل الاختبارات، ولا تكسر التغييرات المحلية«. يبدأ RAZZI AL
acceptance criteria: failing integration passes، unrelated يثبت MissionContract ثم ،Context Genomeو ProjectFingerprintبل
tests unchanged، user dirty state محفوظ.
المسار
MUT: تفعيل MCD/QMCD + knowledge عن auth/Git transactions.
MUG: تصنيف الكتابة REVERSIBLE_WRITE، منع overwrite للملفات dirty.
EVCF: mount repo map، symbols، test logs، relevant commits.
GQE: السؤال الحاكم: هل الفشل validation token أم fixture stale؟
HyperAbility: يركب discover/edit/test/verify. RepositoryFeatureRepair
MOUSSAIXT: اختيار Native/Codex/Hermes حسب measured fit.
ToolGateway: branch/worktree ثم edits وtests.
WET: before hash → intended delta → after + test evidence.
ClosureCourt: يفحص remaining gaps وregression.
ExperienceCompiler: إذا كان stale fixture pattern يولد SkillCandidate متكررا.
هذه المهمة تُظهر لماذا لا تكفي diagrams architecture. كل طبقة يجب أن تظهر في trace أو metric أو gate. إذا لم نستطع إثبات أن MCD غيّر
route أو أن MUG منع overwrite فلا نحسبهما آليتين مطبقتين في تلك run.
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

<PARSED TEXT FOR PAGE: 28 / 30>

برنامج الاختبار: Benchmarks وAblations
A production-grade evidence program
المرحلة التالية ليست زيادة corpus، بل تجميد runtime reference وقياس ما تضيفه الآليات. نحتاج baseline ثابتاً يشارك model، tools، budget،
set task. ثم نحذف أو نضيف طبقة واحدة كل مرة.
Variant إضافة سؤال الاختبار
أساس خط Simple agent loop A0
repo navigation؟ يتحسن هل Context Genome A1
resident context؟ يخفض هل EVCF/MCAS A2
3A PEK/HyperKB هل تتحسن المعرفة التشغيلية؟
compute allocation؟ يتحسن هل MCD/QMCD A4
reuse/transfer؟ يرتفع هل HyperCapability stack A5
false success؟ يخفض هل Courts/WET A6
held-out missions؟ تتحسن هل Self-evolution A7
أفضل؟ harnesses بين routing هل MOUSSAIXT federation A8
المقاييس
Verified Mission Completion، False Success Rate، Remaining Gap at Closure، Recovery Success، Restart Continuity،
Context Efficiency، Tool/Computer Action Success، WorldEffect Verification، Transfer Gain، Cost per Verified،
Delta
Harness Routing Accuracy.
E5: لا يكفي run واحد ناجح. نحتاج repeated seeds، raw traces، matched budget، confidence intervals، held-out tasks، ثم
قاعدة
replication عبر model/domain/environment مستقل.

<PARSED TEXT FOR PAGE: 29 / 30>

فرضيات تطبيقية جديدة قابلة للدحض
New testable hypotheses
تُصاغ التقنيات المقترحة كفرضيات لا كحقائق. كل فرضية التالية يجب أن تتعرض لتجربة يمكن أن تفشلها:
الرمز الفرضية الفشل الذي يخفضها
1H DWR يرفع token per progress verified في المهام عالية التباين لا فرق عن routing static
diagnostic gain يزيد steps دون يخفض time-to-root-cause GQE H2
3H MCAS/EVCF يتفوق على RAG في tasks أكبر من context نفس success بكلفة أعلى
باستمرار أفضل dynamic planning cross-model transfer يحسن HyperAbility reuse H4
reliability gain بلا overhead false success يخفض WET + ClosureCourt H5
lineage tasks خارج يختفي gain tampering دون يحسن held-out tasks HarnessGenome evolution H6
الفائدة يمحو coordination cost single-harness routing على يتفوق CHSR H7
مقياس تجميعي تجريبي
AGI-Oriented Harness Utility = V × T × R × C × E / (Cost + Risk)
معادلة «ليست هذه. V = verified completion، T = transfer، R = recovery، C = context efficiency، E = evidence strength
حيث
AGI «بل index تجريبي لتجنب التحسين أحادي البعد.

<PARSED TEXT FOR PAGE: 30 / 30>

الخلاصة وخارطة الانتقال إلى Candidate Production

Conclusion, roadmap and project anchors

تظهر هذه الورقة أن MUT، MUG وMOUSSAIXT يمكن أن تُفهم كنظام واحد بثلاث وظائف: توحيد ما نريد أن يكون صحيحاً وقابلاً للاختبار، حكم
متى وكيف يسمح لتلك الآليات أن تتحول إلى سلوك، ثم تنفيذها عبر بيئات متعددة مع readback. RAZZI AL هو الharness الذي يغلق الحلقة ويجعل
النتائج قابلة للرصد وإعادة التشغيل.
أقوى التقنيات التي ينبغي تنفيذها واختبارها الآن ليست أسماء إضافية، بل: GTAC لجسر النظرية بالفعل، GQE/DWR لتخصيص المعرفة والحوسبة،
harnesses، CHSR لتوجيه ،الأثر لإثبات ECA/WET ،القدرات لتركيب working set، HCCGللل PEK ،الافتراضي للسياق EVCF/ MCAS
وHarnessGenome للتطور المحكوم.

خارطة 90 يوماً

freeze runtime + schemas + tracing + EVCF baseline :0-30.
WET/Courts + cross-harness adapters + failure injection :31-60.
ablations + held-out missions + self-evolution shadow/canary :61-90.

النتيجة العلمية المطلوبة: الانتقال من rich-specification / rich-artifact إلى نظام له light-validation traces causal وgains controlled. عندها
فقط يمكن رفع بعض الطبقات من 4E3/E إلى 5E.

مراجع المشروع الأساسية

[M1 MOUSSAIF AGI OS / MOUSSAIXT - Deep Engineering]
Architecture Report, 50p, 2026.
[M2 - الكتاب العربي الجامع للنظرية والمعمارية MOUSSAIF Unified AGI]
والتشغيل, 2026 ,p.225
[M3 MUT-AGI-02 / Conscious AGI Master Reference - Unified]
Theory Graph and registry.
[M4 Conscious AGI Project Summary - MUT · MUG · MOUSSAIXT]
system family and evidence gaps.
[M5 / AGI-Oriented Knowledge-Base as Cognitive OS - HyperKB]
HyperSkills / HyperLoops framework.
[M6, ParallaxWorld-MCD FULL AGI OS - difference-first reasoning]
governing question, MCD operationalization.
[M7 MUT MUG ALL Artifact Memory Map - canonicality, duplicate]
and provenance registry, snapshot 2026-09-04.

الخاتمة: MOUSSAIF لا يحتاج في هذه المرحلة إلى »مفاهيم أكثر« بقدر ما يحتاج إلى أن تصبح مفاهيمه الحالية runtime primitives + experiments. الثورة القابلة للدفاع علمياً هي أن نثبت أن layers evidence/context/governance/knowledge تغيّر الأداء والموثوقية تحت حذفها
وإعادتها، لا أن نضاعف أسماءها.

•
•
•