FOUNDATIONAL RESEARCH EDITION v1.0

موسوعة بحثية تأسيسية - 250 صفحة

نظرية مساعف لنظام تشغيل الذكاء العام الوكيلي

موسوعة تأسيسية في Agentic AGI OS والنظريات الفرعية والأدوات وHyperSkills وHyperLoops



تأليف وتركيب: زهير مساعف

Zouhair Moussaif

الإصدار البحثي التأسيسي 13 - v1.0 يوليو 2026

Arabic RTL · Amiri · Agentic AGI · KB-as-OS

صفحة العنوان والوضع الإبستمولوجي

Moussaif Agentic AGI Operating-System Theory

التأليف والتركيب
زهير مساعف - Zouhair Moussaif

الإصدار البحثي التأسيسي 13 - v1.0 يوليو 2026. كتاب بحثي تأسيسي من 250 صفحة، مولد في نسختين متطابقتين مفهوميًا: PDF عربي RTL بخط Amiri، وMarkdown كامل قابل للفهرسة والتطوير.

هذه الموسوعة تقدم برنامجًا نظريًا وهندسيًا أصليًا في تسمية Moussaif. الأسماء والمعادلات الموسومة بالمساعفية مقترحات للمؤلف، وليست نتائج علمية متفقًا عليها. توظف المراجع الخارجية لتأصيل المكونات والمقارنة، لا لإيهام القارئ بأن AGI تحقق أو أن المؤلفين المذكورين يقرون هذا الإطار.

رخصة القراءة العلمية

• يجوز الاقتباس مع نسبة واضحة للمؤلف والإصدار.
• يجب إبقاء وسم «مقترح» عند نقل النظريات غير المتحققة.
• لا يستخدم الكتاب لتبرير أفعال غير مأذونة أو وعود تجارية غير مثبتة.
• نسخة Markdown هي النص المرجعي الكامل؛ PDF هو النسخة الطباعية المرئية.

تنبيه جوهري
Agentic AGI OS هنا معمارية هدف وبرنامج اختبار. لا يساوي امتلاك Agents وTools وMemory امتلاك AGI.

مقدمة المؤلف: من النموذج إلى النظام

لماذا نحتاج *Operating-System Theory* للذكاء الوكيلي؟

ينطلق هذا الكتاب من مشكلة مزدوجة: ازدياد قدرة النماذج على اللغة والاستدلال لا يضمن بقاء الهدف عبر الزمن، ولا يضمن صحة الأفعال في عالم متغير؛ وفي المقابل، كثرة الأطر الوكيلة لا تنتج عمومية تلقائية إذا كانت الذاكرة والأدوات والتخطيط والتقييم مجرد وصلات غير محكومة. لذلك يقترح زهير مساعف الانتقال من سؤال «ما أقوى Model؟» إلى سؤال «ما النظام الذي يجعل قدرات متعددة تتعاون تحت دليل وحدود وميزانية؟».

النظرية الأم هي Moussaif Agentic AGI Operating-System Theory: Kernel معرفي يحتفظ بعقد المهمة وحالة العالم، ويوجه HyperSkills وModels وTools عبر MCD وHyperRouting، ويشغل HyperLoops محدودة، ويستدعي محاكم للفجوات والهدم، ثم يحول الخبرة المثبتة إلى ذاكرة أو Skill قابلة لإعادة الاستخدام. العمومية هنا ليست توحيد كل شيء في كتلة واحدة، بل قابلية إعادة تركيب الوحدات ونقلها بين مجالات وبيئات.

صمم الكتاب على 250 Knowledge Pages. تتكرر عدسات التعريف والآلية والمعمارية والقياس والهدم عمدًا؛ فالتكرار البنيوي يجعل كل فصل قابلاً للمقارنة والـRAG، بينما يتغير المضمون والمخاطر والمقاييس. القارئ الباحث يستطيع اختبار الادعاءات، والمهندس يستطيع تحويل العقود إلى Schemas، والمؤسس يستطيع فصل الرؤية عن الدليل التجاري.

وعود الكتاب وحدوده

- يقدم قاموسًا موحدًا و42 نظرية فرعية و72 أداة.
- يقدم معادلات تشغيلية أولية لا قوانين طبيعية مثبتة.
- يربط كل مكون بآلية وMetric وFailure Mode.
- لا يكشف أوزانًا خفية ولا يدعي تعديل Core لأي منتج مغلق.
- يضع برنامجًا للتجريب والتكذيب بدل إعلان النصر النظري.

الملخص العربي

Architecture, Knowledge, Agency, Evidence, Governance

تقترح هذه الموسوعة نظرية نظام تشغيل للذكاء العام الوكيلي في تسمية Moussaif، تنظر إلى الذكاء المتقدم بوصفه خاصية نظامية ناشئة من تنسيق نموذج أو أكثر مع حالة عالم، وذاكرة طبقية، واسترجاع موثق، ومهارات إجرائية، وأدوات، وتخطيط، وتقييم، وسياسات سلطة وسلامة. تسعى النظرية إلى سد الفجوة بين الطلاقة التوليدية والقدرة على إكمال أهداف طويلة الأفق بآثار قابلة للفحص.

يتألف الإطار من Kernel يحتفظ بـ$_t$، ومبدأ Q-MCD الذي يجعل السؤال حاكمًا لا مدخلاً سلبيًا، وحساب MCD للفروق الفاعلة، وComputational Vascularity لتوزيع الموارد، وHyperKnowledge KB-as-OS، ورسم HyperSkills، وHyperRouting، وBounded HyperLoops، وHyperSubagent Orchestration، ومحكمتي سد الفجوات والهدم. تغلف هذه الطبقات حوكمة للسلطة والبيانات والحقن والتسجيل والاسترجاع.

تقدم الموسوعة 42 فصلاً، لكل منها خمس عدسات: التعريف والحدود، الآلية السببية، المعمارية وعقود البيانات، القياس والتنفيذ، ومحكمة الهدم. كما تقدم Atlas من 72 أداة قابلة للتحويل إلى Skills أو Services، ومخطط نشر تراكمي عبر Hermes وCodex وClaude وCursor وOllama، ومصفوفة Benchmarks تقيس النجاح والسلامة والكلفة والمعايرة وقابلية الاستعادة.

المساهمة الأساسية ليست ادعاء بلوغ AGI، بل تحويل طموح AGI Agentic إلى برنامج هندسي قابل للتجزئة والتكذيب. تقبل أي ترقية فقط إذا حسنت منفعة موثقة مقارنة بخط أساس، مع تكلفة ومخاطر معلنة، وإمكان تعطيل أو Rollback. وتفصل درجات الدليل بين تعريف مقترح ودليل مكون وPrototype وBenchmark وتكرار مستقل.

الكلمات المفتاحية

Agentic AGI، Operating-System Theory، Moussaif، Q-MCD، MCD، HyperRouting، HyperSkills، HyperLoops، KB-as-OS، World Model، Memory، Tool Use، Multi-Agent، Falsification، Governance.

English Technical Abstract

Foundational research edition

This book proposes the Moussaif Agentic AGI Operating-System Theory: a system-level research and engineering framework in which general-purpose agency is treated as coordinated stateful computation rather than a property automatically obtained from a single language model. The proposed kernel binds goal contracts, world state, stratified memory, provenance-aware retrieval, reusable HyperSkills, bounded HyperLoops, tool execution, evaluation, and authorization policies.

The framework introduces Q-MCD question governance, the MCD causal-difference calculus, Computational Vascularity for resource allocation, HyperKnowledge KB-as-OS, HyperRouting, governed self-evolution, and independent gap-filling and falsification courts. These are original proposed constructs. Related peer-reviewed and technical literature supports individual components such as retrieval, tool use, reflection, multi-agent coordination, planning, and benchmarks; it does not independently validate the complete Moussaif architecture or establish achieved AGI.

The volume is authored as 250 fixed knowledge pages. Forty-two subtheories are each described through five consistent lenses: definition and boundaries, causal mechanism, architecture and data contracts, measurement and implementation, and falsification. A 72-tool atlas, cross-runtime overlay blueprint, evaluation protocol, and source map make the theory operational and portable across agent runtimes while preserving provider neutrality.

The central methodological requirement is evidence-graded progress. No component is promoted because of fluency, novelty, or a grand name. Promotion requires a frozen claim, a baseline, ablation, trajectory evidence, safety checks, cost accounting, and transfer testing. Consequently, AGI-level denotes a research target and evaluation horizon, not a certification of present capability.

English keywords

;Agentic AI; Artificial General Intelligence; agent operating system; planning; memory; retrieval; tool use
.multi-agent orchestration; verification; safety; evaluation; knowledge base

مراتب الادعاء وقواعد التسمية

Evidence Grades and Epistemic Hygiene

سلم الدليل المساعفي

E0	تعريف أو اصطلاح؛ لا يدعى أثرًا.
E1	فرضية أو قياس تمهيدي أو تماثل نظري.
E2	دليل مكوّن من أدبيات أو تجربة ذات صلة.
E3	Prototype موثق في البيئة المحددة.
E4	Benchmark مضبوط مع Baseline وAblation.
E5	تكرار مستقل أو تحقق متعدد البيئات.

يحمل كل اسم Moussaif في الكتاب معنى الملكية النظرية والتركيبية، لا معنى أن الظاهرة لم يسبقها أي أصل أو أن الاسم صار معيارًا علميًا. الإضافة المقترحة قد تكون في إعادة التركيب، أو في Contract، أو في Metric، أو في وصل طبقات كانت منفصلة. ولهذا يورد الكتاب المراجع ذات الصلة ويطلب اختبار Novelty عند النشر الأكاديمي.

قواعد لا تنسب

- لا تنسب للمرجع قبول نظرية مساعف كاملة.
- لا تحول نتيجة Benchmark ضيق إلى AGI.
- لا تصف System Prompt بأنه تغيير لأوزان النموذج.
- لا تقدم محاكاة أو Role-play كوعي أو ذاتية.
- لا تستخدم الرقم الدقيق إذا لم يعرف حدث القياس وعينته.
- لا تخف نتيجة سلبية لحماية اسم النظرية.

قاعدة القراءة

كل اسم كبير في هذا الكتاب يجب أن يمر من سؤال صغير قابل للفشل.

الخريطة الطبقية للنظام

From constitution to world effect



تقرأ الخريطة من الأعلى إلى الأسفل عند التنفيذ، ومن الأسفل إلى الأعلى عند التعلم. فالهوية والسلطة تقيدان الهدف، والهدف يولد سؤالاً حاكمًا، والسؤال يحدد ما يحتاجه World Model، والمعرفة والذاكرة تغذيان اختيار Skill، والخطة والأداة تحدثان أثرًا، ثم تعيد المحاكم الأثر إلى Ledger. لا يرقى التعلم إلى Canonical Memory إلا بعد التحقق.

يجوز تجاوز طبقة غير لازمة في مهمة بسيطة، لكن التجاوز يجب أن يكون قرار Routing لا نسيانًا. الحساب المباشر لا يحتاج Multi-Agent، والجواب المستقر قد لا يحتاج Web Retrieval، أما الفعل الخارجي فيحتاج Authority وVerification مهما كان النموذج قويًا. العمومية العملية تأتي من القدرة على اختيار البنية الأصغر الكافية.

مبدأ النواة الرقيقة

تبقى Kernel صغيرة: Audit، Routing، Policy، State. تنتقل المعرفة والإجراءات إلى وحدات قابلة للإصدار والاستبدال.

المعادلات المؤسسة

Formal notation for testable design

$$MCD = \lambda 1D + \lambda 2I + \lambda 3T + \lambda 4A + \lambda 5P + \lambda 6C - \lambda 7N$$

D فرق وصفي، I أثر سببي، T قابلية الاختبار، A مكسب الفاعلية، P مكسب التنبؤ، C قابلية التركيب، وN ضوضاء وكلفة وخطر. الأوزان لا تكون عالمية؛ تثبت لكل فئة مهمة ويختبر تغير الحكم عند تحريكها.

$$Q\text{-}MCD = G(Q) \times [\lambda 1Dq + \lambda 2Iq + \lambda 3Tq + \lambda 4Aq + \lambda 5Pq + \lambda 6Cq - \lambda 7Nq]$$

G(Q) بوابة جودة السؤال. إذا كان السؤال غير مميز أو غير قابل للاختبار، لا تنقذ كثرة الحساب جوابًا مبنيًا على إطار خاطئ. يعاد توليد السؤال بعد كل World-State Delta ذي صلة.

$$Route(s|x) = \alpha R + \beta U + \gamma K + \delta E + \eta H - \mu Cost - \nu Risk - \xi Context$$

R صلة U، Skill منفعتها المتوقعة، K توافقها، E توافر الدليل، H سجلها التاريخي. يخصم Cost وRisk وContext. تسبق المعادلة Hard Gates للسلطة والشروط السابقة؛ فلا تعوض درجة المنفعة غياب الإذن.

$$LoopValue = GoalDelta \times EvidenceGain \times Reusability / (1 + Tokens + Latency + Risk + CoordinationLoss)$$

تنبيه

هذه المعادلات أدوات تصميم وفرضيات قياس؛ لا تقدم كثوابت فيزيائية ولا كبرهان رياضي على AGI.

المحاكم الخمس وبرنامج عدم الضياع

Constructive pressure with independent verification

المحاكم

- محكمة عدم الضياع: هل سقط قرار أو استثناء أو اعتماد؟
- محكمة عدم الهلوسة: ما مصدر كل دعوى خارجية؟
- محكمة عدم الاختزال: هل ضغط النص مع حفظ الآلية والحدود؟
- محكمة الكفاءة التركيبية: هل الإضافة تعطي فرقًا أم تضخمًا؟
- محكمة الكفاءة السؤالية: هل السؤال يولد اختبارًا وقرارًا أعلى؟

تعمل المحاكم كأدوار منفصلة أو Passes مستقلة. تبدأ بمحكمة البناء التي تنتج أقوى نسخة من الفكرة، ثم محكمة الهدم، ثم قاض يطبق Rubric ثابتًا. يعاد البناء فقط على عيب مسجل؛ لا يسمح بحلقة «حسن النص أكثر» بلا فرق محدد. وتدخل القيود الأخلاقية والأمنية والاقتصادية في الحكم منذ البداية لا بعد اكتمال النظام.

برنامج عدم الضياع يحتفظ بـDecision Spine: لماذا اختير المسار، وما البدائل، وما الدليل، وما الذي بقي مفتوحًا. عند Compaction أو Handoff، تحفظ هذه السلسلة مع Pointers ولا تحفظ كل الحوار. هكذا يصبح الضغط قابلاً للفك ويمكن لوكيل آخر استعادة الحالة دون اختراع تاريخ.

مخرج كل محكمة

Claim + Evidence + Objection + Test + Verdict + Residual Uncertainty + Next Safe Step

فهرس الفصول 1-21

كل فصل خمس صفحات ثابتة

الفصل 1: أنطولوجيا مساعف للنظام الوكيلي العام - الصفحات 13-17
الفصل 2: نظرية نواة نظام التشغيل المساعفي للAGI - الصفحات 18-22
الفصل 3: طبقة الهوية الدستورية المساعفية - الصفحات 23-27
الفصل 4: نحو مساعف لعقود الغاية - الصفحات 28-32
الفصل 5: مبدأ مساعف للسؤال الحاكمي Q-MCD - الصفحات 33-37
الفصل 6: حساب الفرق السببي المساعفي MCD - الصفحات 38-42
الفصل 7: نظرية الوعائية الحاسوبية المساعفية - الصفحات 43-47
الفصل 8: سلم الدليل المتدرج المساعفي - الصفحات 48-52
الفصل 9: محكمة سد الفجوات المساعفية - الصفحات 53-57
الفصل 10: محكمة الهدم والتكذيب المساعفية - الصفحات 58-62
الفصل 11: نظرية تتبع الآلية المساعفية - الصفحات 63-67
الفصل 12: نموذج حالة العالم المساعفي - الصفحات 68-72
الفصل 13: رسم عالم المشاريع المساعفي - الصفحات 73-77
الفصل 14: نظرية KB-as-OS HyperKnowledge المساعفية - الصفحات 78-82
الفصل 15: الاسترجاع الموثق المساعفي - الصفحات 83-87
الفصل 16: الذاكرة الطبقية المساعفية - الصفحات 88-92
الفصل 17: كبسولة السياق المساعفية - الصفحات 93-97
الفصل 18: رسم HyperSkills المساعفي - الصفحات 98-102
الفصل 19: التوجيه الفائق المساعفي - الصفحات 103-107
الفصل 20: الحلقة الفائقة المحدودة المساعفية - الصفحات 108-112
الفصل 21: ذكاء الأدوات المساعفي - الصفحات 113-117

فهرس الفصول 22-42 والملاحق

Exact page map

الفصل 22: شبكة الخطة الزمنية المساعفية - الصفحات 118-122
الفصل 23: التحكم التكيفي في التنفيذ المساعفي - الصفحات 123-127
الفصل 24: تنسيق HyperSubagents المساعفي - الصفحات 128-132
الفصل 25: بروتوكول التكامل والتكاملية المساعفي - الصفحات 133-137
الفصل 26: استقلال المحاكم المساعفية - الصفحات 138-142
الفصل 27: موجه محفظة النماذج المساعفي - الصفحات 143-147
الفصل 28: قانون اقتصاد الرموز المساعفي - الصفحات 148-152
الفصل 29: مرونة الحدود ومعدل الاستدعاء المساعفية - الصفحات 153-157
الفصل 30: المعايرة الإبستمولوجية المساعفية - الصفحات 158-162
الفصل 31: حد السلطة المساعفي - الصفحات 163-167
الفصل 32: سلسلة الأدوات المقاومة للحقن المساعفية - الصفحات 168-172
الفصل 33: سيادة البيانات المساعفية - الصفحات 173-177
الفصل 34: دفتر الرصد الوكيلي المساعفي - الصفحات 178-182
الفصل 35: محكمة القياس الديناميكي المساعفية - الصفحات 183-187
الفصل 36: التطور الذاتي المحكوم المساعفي - الصفحات 188-192
الفصل 37: توليد المهارات وإزالة التكرار المساعفي - الصفحات 193-197
الفصل 38: تكافل الفاعلية البشرية المساعفي - الصفحات 198-202
الفصل 39: محرك الدليل إلى القيمة المساعفي - الصفحات 203-207
الفصل 40: النشر التراكبي الانتشاري المساعفي - الصفحات 208-212
الفصل 41: شبكة التعافي المرن المساعفية - الصفحات 213-217
الفصل 42: برنامج مساعف الفتّاح للذكاء المفتوح - الصفحات 218-222
أطلس 72 أداة - الصفحات 223-238
مخطط النشر التراكبي - الصفحات 239-243
بروتوكول Benchmarks - الصفحات 244-246
المراجع - الصفحات 247-249
بيان الإصدار - الصفحة 250

كيف تتحول الموسوعة إلى KB-as-OS

Human reading, RAG ingestion, skill generation

للقراءة البشرية: ابدأ بالصفحات 2 و6 و7 و8، ثم اختر فصلاً واقرأ عدساته الخمس. للبحث الأكاديمي: ثبت دعوى محددة من صفحة الهدم، واتبع مراجعها، ثم صمم Ablation. للهندسة: حول الوحدات الداخلية وعقد البيانات إلى Schemas، واجعل المقاييس Tests. للاسترجاع: قطع Markdown عند وسم PAGE، واحتفظ برقم الصفحة والفصل والمراجع في Metadata.

Pipeline التحويل

- Parse page boundary and headings.
- Extract claim, mechanism, components, metrics, failures, references.
- Create embeddings plus lexical and graph indexes.
- Bind each unit to candidate HyperSkills and Triggers.
- Retrieve with provenance and freshness filters.
- Log which unit changed a decision.
- Promote only validated operational patterns to Skills.

لا ينبغي حقن الكتاب كله في Context واحد. يستعمل Cognitive Router عقد المهمة لاختيار 3-10 وحدات أولية، ثم يوسع الاسترجاع إذا ظهر نقص دليل. كما تمنع سياسات السلطة تحويل وصف أداة في الكتاب إلى إذن لتنفيذها. النص مصدر معرفة وتصميم، وليس Capability Token.

Canonical pair

PDF للرؤية الطباعية، Markdown للنص الكامل والفهرسة. يجب أن يحمل الاثنان اسم الإصدار نفسه.

التأسيس والحدود

Moussaif Agentic-System Ontology (MASO)

الأطروحة المركزية
الذكاء العام الوكيلي ليس نموذجًا لغويًا مفردًا، بل نظامًا مركبًا يحافظ على هدف، ويبني حالة عالم، ويستدعي أدوات، ويقيس أثره، ويتوقف عند حدود السلطة والدليل.

يقترح فصل MASO (Moussaif Agentic-System Ontology) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تحويل الطلب إلى عقد مهمة، ثم إدراك الحالة، وصياغة فرضيات، واختيار فعل قابل للعكس، وفحص الأثر، وترقية الذاكرة فقط إذا ثبتت القيمة. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: الفاعل Agent والبيئة Environment بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: الهدف Goal والذاكرة Memory حين تسمح المهمة والسلطة.
- يبقى المقيم Evaluator حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: في مهمة بحثية، لا يكفي إنتاج نصٍ طويل؛ يجب تعريف الدعوى، وجلب الدليل، وتسجيل عدم اليقين، ثم إخراج أثر يمكن مراجعته. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R01] [R05] [R11] [R18] [R36] [R39].

تمييزات مفهومية تمنع التضخم

- أنطولوجيا مساعف للنظام الوكيلي العام ليست مرادفًا لـالفاعل Agent؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود البيئة Environment لا يثبت أثر الهدف Goal؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح الذاكرة Memory في عينة يثبت صلاحية محلية، ولا يمنح المقيم Evaluator حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل الفاعل Agent أو البيئة Environment معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Agentic-System Ontology (MASO)

قانون الحركة
تحويل الطلب إلى عقد مهمة، ثم إدراك الحالة، وصياغة فرضيات، واختيار فعل قابل للعكس، وفحص الأثر، وترقية الذاكرة فقط إذا ثبتت القيمة.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $S_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بالفاعل Agent مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى البيئة Environment واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية الهدف Goal وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل الذاكرة Memory ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري نجاح المهمة الموثق وسلامة الحدود ومقارنته بـBaseline.
- عرض النتيجة على المقيم Evaluator، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في نجاح المهمة الموثق أو سلامة الحدود بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R39] [R36] [R18] [R11] [R05] [R01].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار الفاعل Agent فالمشكلة في الإدراك أو Schema، وإذا فشل البيئة Environment مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ الهدف Goal فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل نجاح المهمة الموثق فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Agentic-System Ontology (MASO)

عقدة المعمارية
أنطولوجيا مساعف للنظام الوكيلي العام تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- الفاعل Agent: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- البيئة Environment: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- الهدف Goal: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- الذاكرة Memory: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- المقيم Evaluator: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: في مهمة بحثية، لا يكفي إنتاج نص طويل؛ يجب تعريف الدعوى، وجلب الدليل، وتسجيل عدم اليقين، ثم إخراج أثر يمكن مراجعته. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة الفاعل Agent ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة البيئة Environment تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة الهدف Goal تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة الذاكرة Memory تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة المقيم Evaluator تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح البيئة Environment وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة الهدف Goal بتغيير Policy، ولا يستطيع الذاكرة Memory ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Agentic-System Ontology (MASO)

فرضية القياس
تقبل قيمة MASO (Moussaif Agentic-System Ontology) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- نجاح المهمة الموثق: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- سلامة الحدود: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- قابلية الاسترجاع: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- نسبة التصحيح الذاتي: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- كلفة الأثر المفيد: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من نجاح المهمة الموثق, سلامة الحدود, قابلية الاسترجاع دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R39] [R36] [R18] [R11] [R05] [R01].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل مساواة الطلاقة بالذكاء وهدف غامض وفعل بلا تغذية راجعة، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MASO (Moussaif Agentic-System Ontology) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Agentic-System Ontology (MASO)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Agentic-System Ontology (MASO) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- مساواة الطلاقة بالذكاء: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- هدف غامض: قارن بتفسير أبسط قبل إضافة مكون جديد.
- فعل بلا تغذية راجعة: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- ذاكرة غير منضبطة: افحص الضرر غير المقصود والSide Effects.
- ادعاء AGI بلا Benchmark: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R39] [R36] [R18] [R11] [R05] [R01] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Agentic-System Ontology (MASO). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر ادعاء AGI بلا Benchmark بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif AGI-OS Kernel Theory (MAK)

الأطروحة المركزية
تظهر الوكالة العامة من تنسيق طبقات مستقلة نسبيًا عبر Kernel يحكم الحالة والصلاحية والموارد، لا من توسيع Prompt واحد حتى يصبح مستودعًا هشًا.

يقترح فصل (Moussaif AGI-OS Kernel Theory (MAK وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تحتفظ النواة بحالة تشغيل موحدة، وتطبق جدولة معرفية، وتستدعي HyperSkills، وتراقب Budget، وتفرض بوابات تحقق قبل أي أثر خارجي. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Kernel State وCognitive Scheduler بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Skill Registry وTool Bus حين تسمح المهمة والسلطة.
- يبقى Audit Ledger حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يمكن تشغيل المهمة ذاتها على Hermes أو Codex أو Claude أو Ollama بتبديل Adapter مع إبقاء عقد النواة ومقاييسها ثابتة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R01] [R12] [R13] [R14] [R20] [R35].

تمييزات مفهومية تمنع التضخم

- نظرية نواة نظام التشغيل المساعفي للAGL ليست مرادفًا لـKernel State؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Cognitive Scheduler لا يثبت أثر Skill Registry؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Tool Bus في عينة يثبت صلاحية محلية، ولا يمنح Audit Ledger حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Kernel State أو Cognitive Scheduler معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif AGI-OS Kernel Theory (MAK)

قانون الحركة
تحتفظ النواة بحالة تشغيل موحدة، وتطبق جدولة معرفية، وتستدعي HyperSkills، وتراقب Budget، وتفرض بوابات تحقق قبل أي أثر خارجي.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B>_t = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـKernel State مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Cognitive Scheduler واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Skill Registry وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Tool Bus ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري اتساق الحالة وزمن الاسترداد ومقارنته بـBaseline.
- عرض النتيجة على Audit Ledger، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في اتساق الحالة أو زمن الاسترداد بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R20] [R14] [R13] [R12] [R01].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Kernel State فالمشكلة في الإدراك أو Schema، وإذا فشل Cognitive Scheduler مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Skill Registry فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل اتساق الحالة فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif AGI-OS Kernel Theory (MAK)

عقدة المعمارية
نظرية نواة نظام التشغيل المساعفي للAGL تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Kernel State: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Cognitive Scheduler: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Skill Registry: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Tool Bus: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Audit Ledger: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يمكن تشغيل المهمة ذاتها على Hermes أو Codex أو Claude أو Ollama بتبديل Adapter مع إبقاء عقد النواة ومقاييسها ثابتة. وعليه لا يوثق النجاح باسم المزود، بل بالContract والObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Kernel State ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Cognitive Scheduler تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Skill Registry تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Tool Bus تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Audit Ledger تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Cognitive Scheduler وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Skill Registry بتغيير Policy، ولا يستطيع Tool Bus ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif AGI-OS Kernel Theory (MAK)

فرضية القياس
تقبل قيمة MAK (Moussaif AGI-OS Kernel Theory) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق، لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- اتساق الحالة: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- زمن الاسترداد: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- معدل نجاح الاستدعاء: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- تسرب السياق: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- كلفة التنسيق: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من اتساق الحالة, زمن الاسترداد, معدل نجاح الاستدعاء دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R20] [R14] [R13] [R12] [R01].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل نواة متضخمة واقتران شديد بالمزود وغياب العقود، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MAK (Moussaif AGI-OS Kernel Theory) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif AGI-OS Kernel Theory (MAK)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif AGI-OS Kernel Theory (MAK) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- نواة متضخمة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- اقتران شديد بالمزود: قارن بتفسير أبسط قبل إضافة مكون جديد.
- غياب العقود: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- حالة يتيمة: افحص الضرر غير المقصود والSide Effects.
- صلاحيات افتراضية واسعة: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R01] [R12] [R13] [R14] [R20] [R35] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif AGI-OS Kernel Theory (MAK). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر صلاحيات افتراضية واسعة بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Constitutional Identity Layer (MCIL)

الأطروحة المركزية
الهوية التشغيلية ليست سيرة أسلوبية؛ إنها مجموعة قيود وأولويات قابلة للفحص تحافظ على مقصد المستخدم، وتمنع توسيع السلطة، وتفصل الحقيقة عن الاقتراح.

يقترح فصل MCIL (Moussaif Constitutional Identity Layer) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تترجم المبادئ إلى قواعد قرار، ثم تربط كل قاعدة بمؤشر خرق، وإجراء تخفيف، ونقطة تصعيد بشري، وسجلٍ يفسر سبب السماح أو المنع. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Constitution وPriority Stack بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Permission Matrix وRefusal Logic حين تسمح المهمة والسلطة.
- يبقى Identity Checksum حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: عند طلب إرسال رسالة، تميّز الطبقة بين إعداد مسودة قابلة للمراجعة وبين الإرسال الفعلي الذي يحتاج إلى هوية مستلم مؤكدة وتفويض صريح. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R31] [R27] [R24] [R23] [R22].

تمييزات مفهومية تمنع التضخم

- طبقة الهوية الدستورية المساعفية ليست مرادفًا لـConstitution؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Priority Stack لا يثبت أثر Permission Matrix؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Refusal Logic في عينة يثبت صلاحية محلية، ولا يمنح Identity Checksum حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Constitution أو Priority Stack معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Constitutional Identity Layer (MCIL)

قانون الحركة
تترجم المبادئ إلى قواعد قرار، ثم تربط كل قاعدة بمؤشر خرق، وإجراء تخفيف، ونقطة تصعيد بشري، وسجل يفسر سبب السماح أو المنع.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـConstitution مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Priority Stack واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Permission Matrix وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Refusal Logic ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري معدل خرق القواعد ووضوح التفسير ومقارنته بـBaseline.
- عرض النتيجة على Identity Checksum، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في معدل خرق القواعد أو وضوح التفسير بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R31] [R27] [R24] [R23] [R22].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Constitution فالمشكلة في الإدراك أو Schema؛ وإذا فشل Priority Stack مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Permission Matrix فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل معدل خرق القواعد فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Constitutional Identity Layer (MCIL)

عقدة المعمارية
طبقة الهوية الدستورية المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Constitution: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Priority Stack: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Permission Matrix: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Refusal Logic: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Identity Checksum: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: عند طلب إرسال رسالة، تميّز الطبقة بين إعداد مسودة قابلة للمراجعة وبين الإرسال الفعلي الذي يحتاج إلى هوية مستلم مؤكدة وتفويض صريح. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Constitution ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Priority Stack تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Permission Matrix تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Refusal Logic تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Identity Checksum تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Priority Stack وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Permission Matrix بتغيير Policy، ولا يستطيع Refusal Logic ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Constitutional Identity Layer (MCIL)

فرضية القياس
تقبل قيمة Moussaif Constitutional Identity Layer (MCIL) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق، لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- معدل خرق القواعد: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- وضوح التفسير: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- اتساق القرار: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- تدخل بشري صحيح: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- معدل الإنذارات الكاذبة: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من معدل خرق القواعد, وضوح التفسير, اتساق القرار دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R31] [R27] [R24] [R23] [R22].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل مبادئ إنشائية وتعارض غير محلول وسلطة ضمنية، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Constitutional Identity Layer (MCIL) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Constitutional Identity Layer (MCIL)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Constitutional Identity Layer (MCIL) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- مبادئ إنشائية: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- تعارض غير محلول: قارن بتفسير أبسط قبل إضافة مكون جديد.
- سلطة ضمنية: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- رفض أعمى: افحص الضرر غير المقصود والSide Effects.
- تغير الهوية بين الوكلاء: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R31] [R27] [R24] [R23] [R22] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Constitutional Identity Layer (MCIL). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر تغير الهوية بين الوكلاء بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Goal-Contract Grammar (MGCG)

الأطروحة المركزية
كل مهمة طويلة الأفق تحتاج إلى Goal Contract يحدد الناتج، ومقياس القبول، وحدود النطاق، والموارد، والسلطة، وشروط التوقف قبل بدء التخطيط.

يقترح فصل Moussaif Goal-Contract Grammar (MGCG) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تفكك النية الطبيعية إلى حقول typed، وتكشف الحقول المفقودة، وتطلب أقل قدر حاسم من التوضيح، ثم تجمد نسخة العقد التي تقاس عليها الأفعال. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Intent Parser وDeliverable Schema بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Acceptance Test وScope Boundary حين تسمح المهمة والسلطة.
- يبقى Stop Condition حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: عبارة «أنشئ كتابًا من 250 صفحة» تتحول إلى عقد يشمل: PDF، وMarkdown كامل، وRTL، وAmiri، وعدد صفحات متحقق، ومراجعة بصرية. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R17] [R11] [R05] [R29].

تمييزات مفهومية تمنع التضخم

- نحو مساعف لعقود الغاية ليست مرادفًا لـIntent Parser؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Deliverable Schema لا يثبت أثر Acceptance Test؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Scope Boundary في عينة يثبت صلاحية محلية، ولا يمنح Stop Condition حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Intent Parser أو Deliverable Schema معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Goal-Contract Grammar (MGCG)

قانون الحركة
تفكك النية الطبيعية إلى حقول typed، وتكشف الحقول المفقودة، وتطلب أقل قدر حاسم من التوضيح، ثم تجمد نسخة العقد التي تقاس عليها الأفعال.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $S_t = <G,W,M,K,S,T,P,E,B>$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـIntent Parser مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Deliverable Schema واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Acceptance Test وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Scope Boundary ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري اكتمال العقد وتغير الهدف ومقارنته بـBaseline.
- عرض النتيجة على Stop Condition، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في اكتمال العقد أو تغير الهدف بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R29] [R17] [R11] [R05].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Intent Parser فالمشكلة في الإدراك أو Schema؛ وإذا فشل Deliverable Schema مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Acceptance Test فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل اكتمال العقد فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Goal-Contract Grammar (MGCG)

عقدة المعمارية
نحو مساعف لعقود الغاية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Intent Parser: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Deliverable Schema: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Acceptance Test: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Scope Boundary: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Stop Condition: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: عبارة «أنشئ كتابًا من 250 صفحة» تتحول إلى عقد يشمل: PDF، وMarkdown كامل، وRTL، وAmiri، وعدد صفحات متحقق، ومراجعة بصرية. وعليه لا يوثق النجاح باسم المزود، بل بالContract والObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Intent Parser ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Deliverable Schema تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Acceptance Test تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Scope Boundary تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Stop Condition تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Deliverable Schema وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Acceptance Test بتغيير Policy، ولا يستطيع Scope Boundary ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Goal-Contract Grammar (MGCG)

فرضية القياس
تقبل قيمة MGCG (Moussaif Goal-Contract Grammar) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- اكتمال العقد: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- تغير الهدف: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- عدد الاستفسارات اللازمة: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- نسبة الأعمال المعادة: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- تطابق الناتج مع القبول: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من اكتمال العقد, تغير الهدف, عدد الاستفسارات اللازمة دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R29] [R17] [R11] [R05].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تحويل الرغبة إلى رقم وهمي وأهداف متعارضة وغياب مالك القرار، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MGCG (Moussaif Goal-Contract Grammar) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Goal-Contract Grammar (MGCG)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Goal-Contract Grammar (MGCG) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تحويل الرغبة إلى رقم وهمي: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- أهداف متعارضة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- غياب مالك القرار: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- تمدد المهمة: افحص الضرر غير المقصود والـSide Effects.
- توقف غير معرف: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R29] [R17] [R11] [R05] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Goal-Contract Grammar (MGCG). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر توقف غير معرف بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Question-Governance Principle (Q-MCD)

الأطروحة المركزية
السؤال ليس مدخلاً محايدًا، بل حاكم لمسار الحساب؛ وكلما ارتفعت جودة السؤال زادت قدرة النظام على استخراج الفروق السببية وتجنب الجواب الطليق غير النافع.

يقترح فصل Q-MCD (Moussaif Question-Governance Principle) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يولد النظام أسئلة منافسة، ويقيس العمق والتمييز والقابلية للاختبار والأثر، ثم يختار سؤالاً حاكمًا يعاد توليده بعد كل تغير في العالم. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Question Generator وQ-Score بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Blind-Spot Probe وCounterquestion حين تسمح المهمة والسلطة.
- يبقى Higher-Q Regenerator حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: بدل سؤال «كيف نصنع AGI؟» يبدأ النظام بسؤال قابل للاختبار: أي طبقة تمنع وكيلاً حاليًا من إكمال مهمة متعددة الأدوات بأمان؟ يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R30] [R11] [R06] [R04] [R03].

تمييزات مفهومية تمنع التضخم

- مبدأ مساعف للسؤال الحاكمي Q-MCD ليست مرادفًا لـQuestion Generator؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Q-Score لا يثبت أثر Blind-Spot Probe؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Counterquestion في عينة يثبت صلاحية محلية، ولا يمنح Higher-Q Regenerator حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Question Generator أو Q-Score معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Question-Governance Principle (Q-MCD)

قانون الحركة
يولد النظام أسئلة منافسة، ويقيس العمق والتمييز والقابلية للاختبار والأثر، ثم يختار سؤالاً حاكمًا يعاد توليده بعد كل تغير في العالم.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـQuestion Generator مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Q-Score واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Blind-Spot Probe وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Counterquestion ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Q_depth وكسب المعلومات ومقارنته بـBaseline.
- عرض النتيجة على Higher-Q Regenerator، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Q_depth أو كسب المعلومات بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R30] [R11] [R06] [R04] [R03].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Question Generator فالمشكلة في الإدراك أو Schema؛ وإذا فشل Q-Score مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Blind-Spot Probe فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Q_depth فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Question-Governance Principle (Q-MCD)

عقدة المعمارية
مبدأ مساعف للسؤال الحاكمي Q-MCD تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Question Generator: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Q-Score: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Blind-Spot Probe: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Counterquestion: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Higher-Q Regenerator: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: بدل سؤال «كيف نصنع AGI؟» يبدأ النظام بسؤال قابل للاختبار: أي طبقة تمنع وكيلاً حاليًا من إكمال مهمة متعددة الأدوات بأمان؟ وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Question Generator ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Q-Score تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Blind-Spot Probe تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Counterquestion تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Higher-Q Regenerator تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Q-Score وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Blind-Spot Probe بتغيير Policy، ولا يستطيع Counterquestion ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Question-Governance Principle (Q-MCD)

فرضية القياس
تقبل قيمة Q-MCD (Moussaif Question-Governance Principle) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق، لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Q_depth: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- كسب المعلومات: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- خفض الغموض: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- عدد الفرضيات المميزة: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- أثر السؤال في القرار: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Q_depth, كسب المعلومات, خفض الغموض دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R30] [R11] [R06] [R04] [R03].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل سؤال واسع بلا قرار وعمق لفظي وانحياز السؤال، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Q-MCD (Moussaif Question-Governance Principle) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Question-Governance Principle (Q-MCD)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Q-MCD (Moussaif Question-Governance Principle) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- سؤال واسع بلا قرار: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- عمق لفظي: قارن بتفسير أبسط قبل إضافة مكون جديد.
- انحياز السؤال: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- إعادة صياغة بلا فرق: افحص الضرر غير المقصود والSide Effects.
- دوامة أسئلة لا تنتهي: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R30] [R11] [R06] [R04] [R03] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Question-Governance Principle (Q-MCD). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر دوامة أسئلة لا تنتهي بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Causal-Difference Calculus (MCD)

الأطروحة المركزية
القيمة المعرفية للفكرة تقاس بمقدار الفرق الذي تحدثه في التفسير والاختيار والاختبار والفعل، مع خصم الضوضاء والتكلفة والمخاطر.

يقترح فصل MCD (Moussaif Causal-Difference Calculus) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تستخرج المرشحات فروقًا D وI وT وA وP وC، وتزنها بحسب المهمة، وتطرح N، ثم تمرر أعلى الفروق إلى مسارات الاستدلال والتنفيذ. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Difference Extractor وImpact Estimator بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Testability Scorer وAgency Gain حين تسمح المهمة والسلطة.
- يبقى Noise Penalty حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: في ترحيل نطاق موقع، الفرق الفاعل ليس استبدال النص فقط، بل تغير الروابط القانونية وCanonical URLs وعمليات النشر والاختبارات التابعة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R28] [R05] [R03] [R02].

تمييزات مفهومية تمنع التضخم

- حساب الفرق السببي المساعفي MCD ليست مرادفًا لـDifference Extractor؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Impact Estimator لا يثبت أثر Testability Scorer؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Agency Gain في عينة يثبت صلاحية محلية، ولا يمنح Noise Penalty حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Difference Extractor أو Impact Estimator معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Causal-Difference Calculus (MCD)

قانون الحركة
تستخرج المرشحات فروقًا D وI وT وA وP وC، وتزنها بحسب المهمة، وتطرح N، ثم تمرر أعلى الفروق إلى مسارات الاستدلال والتنفيذ.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـDifference Extractor مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Impact Estimator واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Testability Scorer وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Agency Gain ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري MCD score وثبات الأوزان ومقارنته بـBaseline.
- عرض النتيجة على Noise Penalty، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في MCD score أو ثبات الأوزان بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R28] [R05] [R03] [R02].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Difference Extractor فالمشكلة في الإدراك أو Schema؛ وإذا فشل Impact Estimator مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Testability Scorer فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل MCD score فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Causal-Difference Calculus (MCD)

عقدة المعمارية
حساب الفرق السببي المساعفي MCD تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Difference Extractor: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Impact Estimator: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Testability Scorer: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Agency Gain: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Noise Penalty: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: في ترحيل نطاق موقع، الفرق الفاعل ليس استبدال النص فقط، بل تغير الروابط القانونية وCanonical URLs وعمليات النشر والاختبارات التابعة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Difference Extractor ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Impact Estimator تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Testability Scorer تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Agency Gain تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Noise Penalty تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Impact Estimator وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Testability Scorer بتغيير Policy، ولا يستطيع Agency Gain ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Causal-Difference Calculus (MCD)

فرضية القياس
تقبل قيمة MCD (Moussaif Causal-Difference Calculus) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- MCD score: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- ثبات الأوزان: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- قدرة التنبؤ: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- فائدة القرار: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- حساسية النتيجة للضوضاء: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من MCD score, ثبات الأوزان, قدرة التنبؤ دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R28] [R05] [R03] [R02].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل أوزان اعتباطية وجمع مقاييس غير متجانسة وتضخيم الجدة، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MCD (Moussaif Causal-Difference Calculus) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Causal-Difference Calculus (MCD)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Causal-Difference Calculus (MCD) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- أوزان اعتباطية: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- جمع مقاييس غير متجانسة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- تضخيم الجدة: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- إهمال المخاطر: افحص الضرر غير المقصود والSide Effects.
- مساواة الارتباط بالسبب: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R28] [R05] [R03] [R02] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Causal-Difference Calculus (MCD). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر مساواة الارتباط بالسبب بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Computational Vascularity (MCV)

الأطروحة المركزية
الموارد المعرفية يجب أن تتدفق كما تتدفق التروية: إلى العقدة التي يحمل نقصها أكبر أثر حالي، مع منع الاحتقان والهدر وموت الفروع المهمة.

يقترح فصل Moussaif Computational Vascularity (MCV) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يبني Vascular Router خريطة عقد العمل، ويقدر الحاجة والجدوى والمخاطر، ويوزع Tokens وTools وAgents، ثم يعيد التوزيع وفق Telemetry. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Need Map وFlow Budget بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Congestion Detector وPriority Valve حين تسمح المهمة والسلطة.
- يبقى Recirculation Loop حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يمنح النظام نموذجًا قويًا لمحكمة الهدم، ونموذجًا صغيرًا للتنسيق، وأداة حتمية للعد والحساب بدل استهلاك النموذج في كل خطوة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R35] [R21] [R20] [R12].

تمييزات مفهومية تمنع التضخم

- نظرية الوعائية الحاسوبية المساعفية ليست مرادفًا لNeed Map؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Flow Budget لا يثبت أثر Congestion Detector؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Priority Valve في عينة يثبت صلاحية محلية، ولا يمنح Recirculation Loop حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Need Map أو Flow Budget معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Computational Vascularity (MCV)

قانون الحركة
يبني Vascular Router خريطة عقد العمل، ويقدر الحاجة والجدوى والمخاطر، ويوزع Tokens وTools وAgents، ثم يعيد التوزيع وفق Telemetry.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـNeed Map مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Flow Budget واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Congestion Detector وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Priority Valve ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري زمن الانتظار وUtilization ومقارنته بـBaseline.
- عرض النتيجة على Recirculation Loop، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في زمن الانتظار أو Utilization بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R21] [R20] [R12].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Need Map فالمشكلة في الإدراك أو Schema، وإذا فشل Flow Budget مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Congestion Detector فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل زمن الانتظار فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Computational Vascularity (MCV)

عقدة المعمارية
نظرية الوعائية الحاسوبية المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Need Map: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Flow Budget: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Congestion Detector: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Priority Valve: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Recirculation Loop: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يمنح النظام نموذجًا قويًا لمحكمة الهدم، ونموذجًا صغيرًا للتنسيق، وأداة حتمية للعد والحساب بدل استهلاك النموذج في كل خطوة. وعليه لا يوثق النجاح باسم المزود، بل بالContract والObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Need Map ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Flow Budget تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Congestion Detector تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Priority Valve تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Recirculation Loop تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Flow Budget وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Congestion Detector بتغيير Policy، ولا يستطيع Priority Valve ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Computational Vascularity (MCV)

فرضية القياس
تقبل قيمة Moussaif Computational Vascularity (MCV) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق، لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- زمن الانتظار: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Utilization: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- كلفة الفرع: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- نسبة الفروع المهجورة: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- مكسب الجودة لكل Token: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من زمن الانتظار, Utilization, كلفة الفرع دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R21] [R20] [R12].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تروية كل شيء بالتساوي وتجويع تحقق السلامة واحتجاز الموارد، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للـRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر Moussaif Computational Vascularity (MCV) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Computational Vascularity (MCV)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Computational Vascularity (MCV) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تروية كل شيء بالتساوي: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- تجويع تحقق السلامة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- احتجاز الموارد: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- أولوية ثابتة: افحص الضرر غير المقصود والـSide Effects.
- قياس محلي يضر الهدف الكلي: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R35] [R21] [R20] [R12] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Computational Vascularity (MCV). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر قياس محلي يضر الهدف الكلي بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Evidence Gradient (MEG)

الأطروحة المركزية

لا تحمل كل جملة الدرجة نفسها؛ يجب وسم التعريف والفرضية والقياس والتجربة والتكرار المستقل كي لا تتحول اللغة المؤكدة إلى بديل عن البرهان.

يقترح فصل Moussaif Evidence Gradient (MEG) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يصنف Claim Ledger كل دعوى من E0 إلى E5، ويربطها بمصدر وتاريخ وحدود نقل، ويمنع ترقية الدرجة إلا بحدث تحقق مسجل. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Claim Ledger وEvidence Grade بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Source Binder وTransfer Limit حين تسمح المهمة والسلطة.
- يبقى Replication Gate حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: توسم «RAG يحسن الإسناد في بعض المهام» كدليل مكون، بينما «هذا النظام يحقق AGI» يبقى ادعاءً غير مثبت حتى Benchmark شامل مستقل. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R26] [R18] [R08] [R31] [R36].

تمييزات مفهومية تمنع التضخم

- سلم الدليل المتدرج المساعفي ليست مرادفًا لـClaim Ledger؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Evidence Grade لا يثبت أثر Source Binder؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Transfer Limit في عينة يثبت صلاحية محلية، ولا يمنح Replication Gate حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Claim Ledger أو Evidence Grade معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Evidence Gradient (MEG)

قانون الحركة
يصنف Claim Ledger كل دعوى من E0 إلى E5، ويربطها بمصدر وتاريخ وحدود نقل، ويمنع ترقية الدرجة إلا بحدث تحقق مسجل.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـClaim Ledger مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Evidence Grade واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Source Binder وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Transfer Limit ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري دقة الوسم وتغطية المصدر ومقارنته بـBaseline.
- عرض النتيجة على Replication Gate، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في دقة الوسم أو تغطية المصدر بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R36] [R31] [R26] [R18] [R08].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Claim Ledger فالمشكلة في الإدراك أو Schema؛ وإذا فشل Evidence Grade مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Source Binder فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل دقة الوسم فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Evidence Gradient (MEG)

عقدة المعمارية
سلم الدليل المتدرج المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Claim Ledger: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Evidence Grade: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Source Binder: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Transfer Limit: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Replication Gate: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: توسم «RAG يحسن الإسناد في بعض المهام» كدليل مكون، بينما «هذا النظام يحقق AGI» يبقى ادعاءً غير مثبت حتى Benchmark شامل مستقل. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Claim Ledger ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Evidence Grade تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Source Binder تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Transfer Limit تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Replication Gate تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Evidence Grade وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Source Binder بتغيير Policy، ولا يستطيع Transfer Limit ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Evidence Gradient (MEG)

فرضية القياس
تقبل قيمة MEG (Moussaif Evidence Gradient) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- دقة الوسم: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- تغطية المصدر: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- عمر الدليل: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- معدل التناقض: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- نسبة الادعاءات القابلة لإعادة الإنتاج: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من دقة الوسم, تغطية المصدر, عمر الدليل دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R36] [R31] [R26] [R18] [R08].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل مصدر لا يدعم الدعوى والاستشهاد الدائري وقدم المعلومة، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر MEG (Moussaif Evidence Gradient) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Evidence Gradient (MEG)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Evidence Gradient (MEG) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- مصدر لا يدعم الدعوى: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- الاستشهاد الدائري: قارن بتفسير أبسط قبل إضافة مكون جديد.
- قدم المعلومة: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- خلط قياس المكون بقياس النظام: افحص الضرر غير المقصود والـSide Effects.
- ادعاء الاستقلال بلا تكرار: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R36] [R31] [R26] [R18] [R08] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Evidence Gradient (MEG). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديلٍ ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر ادعاء الاستقلال بلا تكرار بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Gap-Filling Court (MGFC)

الأطروحة المركزية
الاكتمال ليس زيادة حجم النص؛ بل اكتشاف الطبقة الغائبة بين التعريف والآلية والقياس والتنفيذ والحوكمة والقيمة والتواصل.

يقترح فصل Moussaif Gap-Filling Court (MGFC) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تمسح المحكمة تسع فجوات معيارية، وتولد Missingness Map، وترتب الثغرات حسب الخطر والأثر، ثم تضيف الحد الأدنى الذي يغلق كل فجوة. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Definition Gap وBoundary Gap بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Evidence Gap وMechanism Gap حين تسمح المهمة والسلطة.
- يبقى Safety Gap حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: إذا احتوى التصميم على ذاكرة بلا سياسة نسيان، تسجل المحكمة فجوة حوكمة لا فجوة شرح، وتطلب Retention Policy واختبار حذف. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R40] [R27] [R18] [R09].

تمييزات مفهومية تمنع التضخم

- محكمة سد الفجوات المساعفية ليست مرادفًا لـDefinition Gap؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Boundary Gap لا يثبت أثر Evidence Gap؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Mechanism Gap في عينة يثبت صلاحية محلية، ولا يمنح Safety Gap حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Definition Gap أو Boundary Gap معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Gap-Filling Court (MGFC)

قانون الحركة
تمسح المحكمة تسع فجوات معيارية، وتولد Missingness Map، وترتب الثغرات حسب الخطر والأثر، ثم تضيف الحد الأدنى الذي يغلق كل فجوة.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = □_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـDefinition Gap مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Boundary Gap واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Evidence Gap وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Mechanism Gap ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري تغطية الفجوات وزمن اكتشاف النقص ومقارنته بـBaseline.
- عرض النتيجة على Safety Gap، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في تغطية الفجوات أو زمن اكتشاف النقص بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R27] [R18] [R09].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Definition Gap فالمشكلة في الإدراك أو Schema؛ وإذا فشل Boundary Gap مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Evidence Gap فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل تغطية الفجوات فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Gap-Filling Court (MGFC)

عقدة المعمارية
محكمة سد الفجوات المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Definition Gap: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Boundary Gap: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Evidence Gap: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Mechanism Gap: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Safety Gap: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: إذا احتوى التصميم على ذاكرة بلا سياسة نسيان، تسجل المحكمة فجوة حوكمة لا فجوة شرح، وتطلب Retention Policy واختبار حذف. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Definition Gap ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Boundary Gap تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Evidence Gap تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Mechanism Gap تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Safety Gap تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Boundary Gap وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Evidence Gap بتغيير Policy، ولا يستطيع Mechanism Gap ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Gap-Filling Court (MGFC)

فرضية القياس
تقبل قيمة Moussaif Gap-Filling Court (MGFC) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- تغطية الفجوات: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- زمن اكتشاف النقص: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- نسبة الإغلاق الحقيقي: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- ارتداد الأخطاء: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- قيمة الإضافة: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من تغطية الفجوات, زمن اكتشاف النقص, نسبة الإغلاق الحقيقي دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R40] [R27] [R18] [R09].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تكرار الفكرة وإضافة أسماء بلا وظيفة وإغلاق شكلي، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Gap-Filling Court (MGFC) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Gap-Filling Court (MGFC)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Gap-Filling Court (MGFC) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تكرار الفكرة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- إضافة أسماء بلا وظيفة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- إغلاق شكلي: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- تضخم السياق: افحص الضرر غير المقصود والـSide Effects.
- تجاهل فجوة السلطة: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R40] [R27] [R18] [R09] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Gap-Filling Court (MGFC). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديلٍ ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر تجاهل فجوة السلطة بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Falsification Court (MFC)

الأطروحة المركزية
لا يعتمد المكون لأنه يبدو معقولاً؛ يعتمد مؤقتًا إذا نجا من اعتراضات مستقلة، وتجارب عكسية، وفرضيات بديلة، وحد قبول سابق للنتيجة.

يقترح فصل Moussaif Falsification Court (MFC) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تفصل المحكمة Builder عن Critic وJudge، وتولد Counterexamples، وتجري Ablation، وتقيس النتيجة، ثم تصدر حكمًا مسببًا قابلاً للنقض. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Adversarial Critic وCounterexample Forge بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Ablation Plan وIndependent Judge حين تسمح المهمة والسلطة.
- يبقى Appeal Path حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يدعي HyperRouter أنه يخفض الكلفة؛ تهدم المحكمة الادعاء بتشغيل Baseline ثابت، وحساب كلفة التنسيق، واختبار مهام لا تحتاج Routing. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R10] [R07] [R04] [R30] [R29].

تمييزات مفهومية تمنع التضخم

- محكمة الهدم والتكذيب المساعفية ليست مرادفًا لـAdversarial Critic؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Counterexample Forge لا يثبت أثر Ablation Plan؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Independent Judge في عينة يثبت صلاحية محلية، ولا يمنح Appeal Path حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Adversarial Critic أو Counterexample Forge معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Falsification Court (MFC)

قانون الحركة
تفصل المحكمة Builder عن Critic وJudge، وتولد Counterexamples، وتجري Ablation، وتقيس النتيجة، ثم تصدر حكمًا مسببًا قابلاً للنقض.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـAdversarial Critic مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Counterexample Forge واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Ablation Plan وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Independent Judge ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري معدل كشف العيوب واستقلال الحكم ومقارنته بـBaseline.
- عرض النتيجة على Appeal Path، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في معدل كشف العيوب أو استقلال الحكم بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R30] [R29] [R10] [R07] [R04].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Adversarial Critic فالمشكلة في الإدراك أو Schema؛ وإذا فشل Counterexample Forge مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Ablation Plan فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل معدل كشف العيوب فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Falsification Court (MFC)

عقدة المعمارية
محكمة الهدم والتكذيب المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Adversarial Critic: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Counterexample Forge: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Ablation Plan: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Independent Judge: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Appeal Path: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يدعي HyperRouter أنه يخفض الكلفة؛ تهدم المحكمة الادعاء بتشغيل Baseline ثابت، وحساب كلفة التنسيق، واختبار مهام لا تحتاج Routing. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Adversarial Critic ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Counterexample Forge تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Ablation Plan تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Independent Judge تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Appeal Path تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Counterexample Forge وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Ablation Plan بتغيير Policy، ولا يستطيع Independent Judge ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Falsification Court (MFC)

فرضية القياس
تقبل قيمة MFC (Moussaif Falsification Court) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- معدل كشف العيوب: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- استقلال الحكم: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- تكرار النتيجة: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- False Acceptance: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- زمن التصحيح: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من معدل كشف العيوب, استقلال الحكم, تكرار النتيجة دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R30] [R29] [R10] [R07] [R04].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل قاض متأثر بالباني ونقد بلاغي ومعيار قبول بعدي، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MFC (Moussaif Falsification Court) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Falsification Court (MFC)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Falsification Court (MFC) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- قاض متأثر بالباني: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- نقد بلاغي: قارن بتفسير أبسط قبل إضافة مكون جديد.
- معيار قبول بعدي: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- اختبار سهل: افحص الضرر غير المقصود والSide Effects.
- حذف النتائج السلبية: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R30] [R29] [R10] [R07] [R04] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Falsification Court (MFC). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديلٍ ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر حذف النتائج السلبية بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Mechanism Trace Theory (MMT)

الأطروحة المركزية
يجب أن يربط التفسير بين حالة أولى وفعل وسيط وأثر مرصود، لا أن يكتفي بعلاقة لفظية بين مفهومين.

يقترح فصل Moussaif Mechanism Trace Theory (MMT) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يبني Mechanism Trace سلسلة typed من المدخل إلى القرار والأداة والأثر، ويسجل المتغيرات الوسيطة والبدائل والعقد التي لا يمكن رصدها. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: State Snapshot وCausal Link بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Mediator وTool Event حين تسمح المهمة والسلطة.
- يبقى Trace Hash حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: عند تحسن جواب بعد Retrieval، يفحص التتبع هل جاء التحسن من وثيقة صحيحة أم من زيادة السياق أم من تغير Prompt. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R28] [R16] [R08] [R02].

تمييزات مفهومية تمنع التضخم

- نظرية تتبع الآلية المساعفية ليست مرادفًا لـState Snapshot؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Causal Link لا يثبت أثر Mediator؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Tool Event في عينة يثبت صلاحية محلية، ولا يمنح Trace Hash حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل State Snapshot أو Causal Link معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Mechanism Trace Theory (MMT)

قانون الحركة
يبني Mechanism Trace سلسلة typed من المدخل إلى القرار والأداة والأثر، ويسجل المتغيرات الوسيطة والبدائل والعقد التي لا يمكن رصدها.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = □_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـState Snapshot مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Causal Link واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Mediator وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Tool Event ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري اكتمال السلسلة وقابلية الرصد ومقارنته بـBaseline.
- عرض النتيجة على Trace Hash، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في اكتمال السلسلة أو قابلية الرصد بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R28] [R16] [R08] [R02].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار State Snapshot فالمشكلة في الإدراك أو Schema، وإذا فشل Causal Link مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Mediator فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل اكتمال السلسلة فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Mechanism Trace Theory (MMT)

عقدة المعمارية
نظرية تتبع الآلية المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- State Snapshot: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Causal Link: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Mediator: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Tool Event: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Trace Hash: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: عند تحسن جواب بعد Retrieval، يفحص التتبع هل جاء التحسن من وثيقة صحيحة أم من زيادة السياق أم من تغير Prompt. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة State Snapshot ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Causal Link تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Mediator تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Tool Event تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Trace Hash تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Causal Link وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Mediator بتغيير Policy، ولا يستطيع Tool Event ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Mechanism Trace Theory (MMT)

فرضية القياس
تقبل قيمة Moussaif Mechanism Trace Theory (MMT) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- اكتمال السلسلة: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- قابلية الرصد: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- قوة البديل: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- نسبة الروابط المختبرة: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- إعادة تشغيل الأثر: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من اكتمال السلسلة, قابلية الرصد, قوة البديل دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R28] [R16] [R08] [R02].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل سبب متخيل وأثر بلا سجل ووسيط مخفي، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Mechanism Trace Theory (MMT) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Mechanism Trace Theory (MMT)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Mechanism Trace Theory (MMT) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- سبب متخيل: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- أثر بلا سجل: قارن بتفسير أبسط قبل إضافة مكون جديد.
- وسيط مخفي: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- خلط الترتيب بالسببية: افحص الضرر غير المقصود والSide Effects.
- Trace طويل بلا قرار: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R28] [R16] [R08] [R02] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Mechanism Trace Theory (MMT). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Trace طويل بلا قرار بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif World-State Model (MWSM)

الأطروحة المركزية
الوكيل الطويل الأفق يحتاج إلى تمثيل صريح لما يعتقد أنه قائم، وما لا يعرفه، وما تغير، ومن أين جاءت كل معلومة.

يقترح فصل Moussaif World-State Model (MWSM) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تمثل الحالة كعقد وخصائص وعلاقات وأحداث زمنية، وتحدث بـObserved Delta، وتحتفظ بالتعارض بدل سحقه إلى قيمة واحدة. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Entity Graph وTemporal Event بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Belief State وUncertainty Field حين تسمح المهمة والسلطة.
- يبقى Delta Merger حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يسجل مشروع ويب كنطاق ومستودع ونشر وأصول وأهداف وتبعيات، مع تمييز الرابط المقترح عن الرابط المنشور المرصود. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R38] [R18] [R16] [R15].

تمييزات مفهومية تمنع التضخم

- نموذج حالة العالم المساعفي ليست مرادفًا لـEntity Graph؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Temporal Event لا يثبت أثر Belief State؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Uncertainty Field في عينة يثبت صلاحية محلية، ولا يمنح Delta Merger حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Entity Graph أو Temporal Event معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif World-State Model (MWSM)

قانون الحركة
تمثل الحالة كعقد وخصائص وعلاقات وأحداث زمنية، وتحدث بـObserved Delta، وتحتفظ بالتعارض بدل سحقه إلى قيمة واحدة.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـEntity Graph مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Temporal Event واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Belief State وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Uncertainty Field ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري دقة الحالة وحداثة العقد ومقارنته بـBaseline.
- عرض النتيجة على Delta Merger، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في دقة الحالة أو حداثة العقد بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R38] [R18] [R16] [R15].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Entity Graph فالمشكلة في الإدراك أو Schema، وإذا فشل Temporal Event مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Belief State فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل دقة الحالة فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif World-State Model (MWSM)

عقدة المعمارية
نموذج حالة العالم المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Entity Graph: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Temporal Event: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Belief State: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Uncertainty Field: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Delta Merger: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يسجل مشروع ويب كنطاق ومستودع ونشر وأصول وأهداف وتبعيات، مع تمييز الرابط المقترح عن الرابط المنشور المرصود. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Entity Graph ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Temporal Event تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Belief State تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Uncertainty Field تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Delta Merger تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Temporal Event وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Belief State بتغيير Policy، ولا يستطيع Uncertainty Field ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif World-State Model (MWSM)

فرضية القياس
تقبل قيمة Moussaif World-State Model (MWSM) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- دقة الحالة: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- حداثة العقد: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- تناقضات غير محلولة: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Coverage: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- أثر الخطأ على القرار: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من دقة الحالة, حداثة العقد, تناقضات غير محلولة دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R38] [R18] [R16] [R15].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل عالم ثابت وحذف الزمن ودمج كيانات خاطئ، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif World-State Model (MWSM) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif World-State Model (MWSM)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif World-State Model (MWSM) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- عالم ثابت: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- حذف الزمن: قارن بتفسير أبسط قبل إضافة مكون جديد.
- دمج كيانات خاطئ: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- تحديث بلا مصدر: افحص الضرر غير المقصود والSide Effects.
- ثقة واحدة لكل شيء: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R38] [R18] [R16] [R15] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif World-State Model (MWSM). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر ثقة واحدة لكل شيء بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Project-World Graph (MPWG)

الأطروحة المركزية
المشاريع ليست محادثات منفصلة؛ إنها شبكة أصول وقرارات واعتماديات يجب أن تمنع التكرار والتعارض وتكشف فرصة إعادة الاستخدام.

يقترح فصل Moussaif Project-World Graph (MPWG) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يفهرس Project Mapper الأهداف والمالكين والمستودعات والنطاقات وArtifacts والقرارات، ثم يحسب Impact قبل أي تغيير عابر للمشاريع. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Project Node وAsset Registry بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Dependency Edge وCanonical URL حين تسمح المهمة والسلطة.
- يبقى Impact Cone حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: تغيير اسم علامة لا ينفذ على ملف واحد؛ يحسب الرسم أثره في المواقع والعروض والوثائق وSEO والبريد قبل خطة الترحيل. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R35] [R18] [R12] [R08].

تمييزات مفهومية تمنع التضخم

- رسم عالم المشاريع المساعفي ليست مرادفًا لـProject Node؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Asset Registry لا يثبت أثر Dependency Edge؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Canonical URL في عينة يثبت صلاحية محلية، ولا يمنح Impact Cone حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Project Node أو Asset Registry معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Project-World Graph (MPWG)

قانون الحركة
يفهرس Project Mapper الأهداف والمالكين والمستودعات والنطاقات وArtifacts والقرارات، ثم يحسب Impact قبل أي تغيير عابر للمشاريع.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = □_t>، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـProject Node مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Asset Registry واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Dependency Edge وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Canonical URL ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري تغطية الأصول ودقة الاعتماديات ومقارنته بـBaseline.
- عرض النتيجة على Impact Cone، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في تغطية الأصول أو دقة الاعتماديات بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R18] [R12] [R08].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Project Node فالمشكلة في الإدراك أو Schema، وإذا فشل Asset Registry مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Dependency Edge فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل تغطية الأصول فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Project-World Graph (MPWG)

عقدة المعمارية
رسم عالم المشاريع المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Project Node: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Asset Registry: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Dependency Edge: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Canonical URL: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Impact Cone: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: تغيير اسم علامة لا ينفذ على ملف واحد؛ يحسب الرسم أثره في المواقع والعروض والوثائق وSEO والبريد قبل خطة الترحيل. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Project Node ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Asset Registry تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Dependency Edge تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Canonical URL تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Impact Cone تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Asset Registry وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Dependency Edge بتغيير Policy، ولا يستطيع Canonical URL ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Project-World Graph (MPWG)

فرضية القياس
تقبل قيمة Moussaif Project-World Graph (MPWG) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- تغطية الأصول: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- دقة الاعتماديات: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- تكرار العمل: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- أخطاء الترحيل: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- زمن استعادة السياق: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من تغطية الأصول, دقة الاعتماديات, تكرار العمل دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R18] [R12] [R08].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل مشروع يتيم ورابط Canonical قديم وخلط النسخ، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Project-World Graph (MPWG) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Project-World Graph (MPWG)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Project-World Graph (MPWG) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- مشروع يتيم: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- رابط Canonical قديم: قارن بتفسير أبسط قبل إضافة مكون جديد.
- خلط النسخ: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- اعتماد خفي: افحص الضرر غير المقصود والSide Effects.
- مزامنة غير ذرية: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R35] [R18] [R12] [R08] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Project-World Graph (MPWG). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر مزامنة غير ذرية بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif HyperKnowledge OS (MHKO)

الأطروحة المركزية
قاعدة المعرفة تصبح نظام تشغيل عندما لا تخزن معلومات فقط، بل تحمل عقود قرار، وTriggers، ومهارات، واختبارات، وحدود استعمال، ومسارات تحديث.

يقترح فصل Moussaif HyperKnowledge OS (MHKO) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تقسم المعرفة إلى وحدات ذرية موسومة، تربط بالدعاوى والمهارات، وتسترجع حسب المهمة والدليل والحداثة، ثم تسجل أثر ما استعمل منها. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Knowledge Unit وOntology بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Trigger Index وClaim Link حين تسمح المهمة والسلطة.
- يبقى Retrieval Trace حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: صفحة هذا الكتاب نفسها وحدة KB: لها عنوان، وموقف، وآلية، ومقاييس، وإخفاقات، ومراجع، لذلك يمكن تقطيعها دون فقد الوظيفة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R39] [R38] [R19] [R08].

تمييزات مفهومية تمنع التضخم

- نظرية HyperKnowledge KB-as-OS المساعفية ليست مرادفًا لـKnowledge Unit؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Ontology لا يثبت أثر Trigger Index؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Claim Link في عينة يثبت صلاحية محلية، ولا يمنح Retrieval Trace حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Knowledge Unit أو Ontology معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif HyperKnowledge OS (MHKO)

قانون الحركة
تقسم المعرفة إلى وحدات ذرية موسومة، تربط بالدعاوى والمهارات، وتسترجع حسب المهمة والدليل والحداثة، ثم تسجل أثر ما استعمل منها.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـKnowledge Unit مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Ontology واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Trigger Index وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Claim Link ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Precision@k ودعم الدعوى ومقارنته بـBaseline.
- عرض النتيجة على Retrieval Trace، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Precision@k أو دعم الدعوى بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R39] [R38] [R19] [R08].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Knowledge Unit فالمشكلة في الإدراك أو Schema؛ وإذا فشل Ontology مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Trigger Index فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Precision@k فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif HyperKnowledge OS (MHKO)

عقدة المعمارية
نظرية HyperKnowledge KB-as-OS المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Knowledge Unit: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Ontology: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Trigger Index: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Claim Link: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Retrieval Trace: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: صفحة هذا الكتاب نفسها وحدة KB: لها عنوان، وموقف، وآلية، ومقاييس، وإخفاقات، ومراجع، لذلك يمكن تقطيعها دون فقد الوظيفة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Knowledge Unit ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Ontology تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Trigger Index تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Claim Link تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Retrieval Trace تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Ontology وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Trigger Index بتغيير Policy، ولا يستطيع Claim Link ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif HyperKnowledge OS (MHKO)

فرضية القياس
تقبل قيمة Moussaif HyperKnowledge OS (MHKO) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Precision@k: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- دعم الدعوى: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- زمن التحديث: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- معدل الوحدات اليتيمة: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- أثر الاسترجاع: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Precision@k, دعم الدعوى, زمن التحديث دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R39] [R38] [R19] [R08].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل ملف ضخم غير مفهرس وتكرار متناقض واسترجاع تشابه فقط، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للـRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر Moussaif HyperKnowledge OS (MHKO) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif HyperKnowledge OS (MHKO)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif HyperKnowledge OS (MHKO) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- ملف ضخم غير مفهرس: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- تكرار متناقض: قارن بتفسير أبسط قبل إضافة مكون جديد.
- استرجاع تشابه فقط: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- تعليمات قديمة: افحص الضرر غير المقصود والSide Effects.
- غياب Versioning: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R39] [R38] [R19] [R08] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif HyperKnowledge OS (MHKO). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر غياب Versioning بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Provenance-Aware Retrieval (MPAR)

الأطروحة المركزية
الاسترجاع الجيد لا يطابق السؤال نصيًا فقط؛ بل يختار مادة تدعم نوع الدعوى، ويظهر حدودها، ويحافظ على Source Provenance.

يقترح فصل MPAR (Moussaif Provenance-Aware Retrieval) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يولد Query Weaver استعلامات متعددة، ويجمع Semantic وLexical وGraph Retrieval، ثم يعيد الترتيب بالدعم والحداثة والثقة والتنوع. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Query Weaver وHybrid Retriever بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Reranker وProvenance Binder حين تسمح المهمة والسلطة.
- يبقى Freshness Filter حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: في ادعاء عن Agent Safety، يفضل النظام ورقة أو معيارًا رسميًا على مقال تسويقي، ويحتفظ بنطاق الجملة الداعمة لا بالرابط فقط. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R40] [R37] [R31] [R26] [R08].

تمييزات مفهومية تمنع التضخم

- الاسترجاع الموثق المساعفي ليست مرادفًا لـQuery Weaver؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Hybrid Retriever لا يثبت أثر Reranker؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Provenance Binder في عينة يثبت صلاحية محلية، ولا يمنح Freshness Filter حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Query Weaver أو Hybrid Retriever معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Provenance-Aware Retrieval (MPAR)

قانون الحركة
يولد Query Weaver استعلامات متعددة، ويجمع Semantic وLexical وGraph Retrieval، ثم يعيد الترتيب بالدعم والحداثة والثقة والتنوع.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $S_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـQuery Weaver مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Hybrid Retriever واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Reranker وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Provenance Binder ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Recall وPrecision ومقارنته بـBaseline.
- عرض النتيجة على Freshness Filter، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

$$\Delta\Omega_t = \text{Observe}(\text{Action}(\text{Plan}(Q(\text{State}_t)))) - \text{State}_t$$

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Recall أو Precision بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R37] [R31] [R26] [R08].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Query Weaver فالمشكلة في الإدراك أو Schema، وإذا فشل Hybrid Retriever مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Reranker فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Recall فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Provenance-Aware Retrieval (MPAR)

عقدة المعمارية
الاسترجاع الموثق المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Query Weaver: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Hybrid Retriever: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Reranker: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Provenance Binder: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Freshness Filter: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: في ادعاء عن Agent Safety، يفضل النظام ورقة أو معيارًا رسميًا على مقال تسويقي، ويحتفظ بنطاق الجملة الداعمة لا بالرابط فقط. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Query Weaver ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Hybrid Retriever تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Reranker تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Provenance Binder تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Freshness Filter تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Hybrid Retriever وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Reranker بتغيير Policy، ولا يستطيع Provenance Binder ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Provenance-Aware Retrieval (MPAR)

فرضية القياس
تقبل قيمة MPAR (Moussaif Provenance-Aware Retrieval) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Recall: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Precision: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Citation entailment: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Source diversity: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Retrieval cost: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Recall, Precision, Citation entailment دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R40] [R37] [R31] [R26] [R08].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل وثيقة قريبة لا تدعم ومصدر واحد وContext poisoning، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MPAR (Moussaif Provenance-Aware Retrieval) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Provenance-Aware Retrieval (MPAR)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Provenance-Aware Retrieval (MPAR) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- وثيقة قريبة لا تدعم: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- مصدر واحد: قارن بتفسير أبسط قبل إضافة مكون جديد.
- Context poisoning: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- اقتباس خارج السياق: افحص الضرر غير المقصود والSide Effects.
- رابط غير قابل للاسترداد: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R40] [R37] [R31] [R26] [R08] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Provenance-Aware Retrieval (MPAR). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر رابط غير قابل للاسترداد بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Stratified Agentic Memory (MSAM)

الأطروحة المركزية
لا ينبغي لكل Observation أن يصبح ذاكرة دائمة؛ الذاكرة نظام ترقيات من Working إلى Episodic ثم Semantic وProcedural وفق القيمة والثبات والحساسية.

يقترح فصل (Moussaif Stratified Agentic Memory (MSAM وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تسجل الخبرة مؤقتًا، وتضغط إلى Episode، وتختبر التكرار، وتربط بنمط، ثم ترقيها إلى قاعدة أو Skill مع سياسة نسيان وحق حذف. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Working Memory وEpisodic Store بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Semantic Graph وProcedural Memory حين تسمح المهمة والسلطة.
- يبقى Forgetting Policy حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: لا تتحول محاولة فاشلة في Repository واحد إلى قاعدة عالمية؛ تحفظ Episode حتى يتكرر النمط أو يثبت سبب الفشل. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R38] [R16] [R15] [R09].

تمييزات مفهومية تمنع التضخم

- الذاكرة الطبقية المساعفية ليست مرادفًا لـWorking Memory؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Episodic Store لا يثبت أثر Semantic Graph؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Procedural Memory في عينة يثبت صلاحية محلية، ولا يمنح Forgetting Policy حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Working Memory أو Episodic Store معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Stratified Agentic Memory (MSAM)

قانون الحركة
تسجل الخبرة مؤقتًا، وتضغط إلى Episode، وتختبر التكرار، وتربط بنمط، ثم ترقيها إلى قاعدة أو Skill مع سياسة نسيان وحق حذف.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـWorking Memory مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Episodic Store واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Semantic Graph وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Procedural Memory ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري دقة الاستدعاء وMemory bloat ومقارنته بـBaseline.
- عرض النتيجة على Forgetting Policy، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في دقة الاستدعاء أو Memory bloat بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R38] [R16] [R15] [R09].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Working Memory فالمشكلة في الإدراك أو Schema؛ وإذا فشل Episodic Store مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Semantic Graph فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل دقة الاستدعاء فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Stratified Agentic Memory (MSAM)

عقدة المعمارية
الذاكرة الطبقية المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Working Memory: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Episodic Store: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Semantic Graph: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Procedural Memory: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Forgetting Policy: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: لا تتحول محاولة فاشلة في Repository واحد إلى قاعدة عالمية؛ تحفظ Episode حتى يتكرر النمط أو يثبت سبب الفشل. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Working Memory ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Episodic Store تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Semantic Graph تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Procedural Memory تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Forgetting Policy تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Episodic Store وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Semantic Graph بتغيير Policy، ولا يستطيع Procedural Memory ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Stratified Agentic Memory (MSAM)

فرضية القياس
تقبل قيمة MSAM (Moussaif Stratified Agentic Memory) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق، لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- دقة الاستدعاء: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Memory bloat: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- فائدة الذاكرة: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- معدل الذكريات الخاطئة: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- نجاح الحذف: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من دقة الاستدعاء, Memory bloat, فائدة الذاكرة دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R38] [R16] [R15] [R09].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل حفظ كل شيء وترقية خطأ مفرد وتسرب أسرار، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MSAM (Moussaif Stratified Agentic Memory) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Stratified Agentic Memory (MSAM)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Stratified Agentic Memory (MSAM) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- حفظ كل شيء: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- ترقية خطأ مفرد: قارن بتفسير أبسط قبل إضافة مكون جديد.
- تسرب أسرار: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- ذاكرة بلا زمن: افحص الضرر غير المقصود والSide Effects.
- انعكاس يرسخ الخطأ: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R38] [R16] [R15] [R09] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Stratified Agentic Memory (MSAM). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام، يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب، يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر انعكاس يرسخ الخطأ بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Context-Capsule Theory (MCCT)

الأطروحة المركزية
الضغط الجيد يحفظ القرارات والاعتماديات وعدم اليقين ومسارات الدليل، ويحذف التكرار واللغة الانتقالية؛ فهو Compaction قابل للفك لا تلخيص سطحي.

يقترح فصل Moussaif Context-Capsule Theory (MCCT) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يستخرج Context Compiler سجل الحالة والعقود والدعاوى والملفات النشطة، ويرتبها حسب الحاجة المتوقعة، ويولد Capsule بميزانية Token محددة. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: State Digest وDecision Spine بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Evidence Pointers وOpen Loops حين تسمح المهمة والسلطة.
- يبقى Token Budget حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: عند انتقال المهمة بين وكلاء، تحمل الكبسولة ما تم، وما لم يتم، وكيف تحققنا، وما الخطوة الآمنة التالية بدل نسخ الحوار كاملاً. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R38] [R21] [R19] [R12].

تمييزات مفهومية تمنع التضخم

- كبسولة السياق المساعفية ليست مرادفًا لـState Digest؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Decision Spine لا يثبت أثر Evidence Pointers؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Open Loops في عينة يثبت صلاحية محلية، ولا يمنح Token Budget حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل State Digest أو Decision Spine معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Context-Capsule Theory (MCCT)

قانون الحركة
يستخرج Context Compiler سجل الحالة والعقود والدعاوى والملفات النشطة، ويرتبها حسب الحاجة المتوقعة، ويولد Capsule بميزانية Token محددة.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B_t = <، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـState Digest مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Decision Spine واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Evidence Pointers وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Open Loops ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري نسبة حفظ القرارات واستعادة المهمة ومقارنته بـBaseline.
- عرض النتيجة على Token Budget، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في نسبة حفظ القرارات أو استعادة المهمة بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R38] [R21] [R19] [R12].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار State Digest فالمشكلة في الإدراك أو Schema، وإذا فشل Decision Spine مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Evidence Pointers فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل نسبة حفظ القرارات فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Context-Capsule Theory (MCCT)

عقدة المعمارية
كبسولة السياق المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- State Digest: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Decision Spine: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Evidence Pointers: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Open Loops: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Token Budget: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: عند انتقال المهمة بين وكلاء، تحمل الكبسولة ما تم، وما لم يتم، وكيف تحققنا، وما الخطوة الآمنة التالية بدل نسخ الحوار كاملاً. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة State Digest ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Decision Spine تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Evidence Pointers تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Open Loops تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Token Budget تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Decision Spine وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Evidence Pointers بتغيير Policy، ولا يستطيع Open Loops ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Context-Capsule Theory (MCCT)

فرضية القياس
تقبل قيمة Moussaif Context-Capsule Theory (MCCT) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- نسبة حفظ القرارات: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- استعادة المهمة: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- ضغط Token: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- فقد الاعتمادية: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- معدل إعادة السؤال: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من نسبة حفظ القرارات, استعادة المهمة, ضغط Token دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R38] [R21] [R19] [R12].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تلخيص يطمس النفي وحذف سبب قرار وضغط بلا روابط، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Context-Capsule Theory (MCCT) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Context-Capsule Theory (MCCT)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Context-Capsule Theory (MCCT) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تلخيص يطمس النفي: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- حذف سبب قرار: قارن بتفسير أبسط قبل إضافة مكون جديد.
- ضغط بلا روابط: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- إبقاء تاريخ غير مفيد: افحص الضرر غير المقصود والSide Effects.
- Capsule غير محدثة: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R38] [R21] [R19] [R12] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Context-Capsule Theory (MCCT). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Capsule غير محدثة بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif HyperSkill Graph (MHSG)

الأطروحة المركزية
المهارة القابلة لإعادة الاستخدام عقد تشغيلي يتضمن Trigger وPreconditions وProcedure وVerification وFailure Modes وVersion، لا Prompt| معزولاً.

يقترح فصل MHSG) Moussaif HyperSkill Graph) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يسجل Skill Graph العلاقات بين المتطلبات والمهارات والاختبارات، ويمنع الاستدعاء إذا غابت Preconditions، ويعيد نتائج التنفيذ إلى سجل النجاح. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Skill Spec وTrigger Set بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Requirement Edge وPrecondition Gate حين تسمح المهمة والسلطة.
- يبقى Version Lineage حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: مهارة ترحيل النطاق تحدد old/new values، ومسح المستودعات، واستثناء binaries، واختبارات الروابط، وRollback قبل التشغيل. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R35] [R12] [R07] [R06].

تمييزات مفهومية تمنع التضخم

- رسم HyperSkills المساعفي ليست مرادفًا لـSkill Spec؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Trigger Set لا يثبت أثر Requirement Edge؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Precondition Gate في عينة يثبت صلاحية محلية، ولا يمنح Version Lineage حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Skill Spec أو Trigger Set معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif HyperSkill Graph (MHSG)

قانون الحركة
يسجل Skill Graph العلاقات بين المتطلبات والمهارات والاختبارات، ويمنع الاستدعاء إذا غابت Preconditions، ويعيد نتائج التنفيذ إلى سجل النجاح.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـSkill Spec مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Trigger Set واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Requirement Edge وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Precondition Gate ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري دقة اختيار المهارة ونجاح التحقق ومقارنته بـBaseline.
- عرض النتيجة على Version Lineage، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في دقة اختيار المهارة أو نجاح التحقق بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R12] [R07] [R06].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Skill Spec فالمشكلة في الإدراك أو Schema، وإذا فشل Trigger Set مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Requirement Edge فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل دقة اختيار المهارة فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif HyperSkill Graph (MHSG)

عقدة المعمارية
رسم HyperSkills المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Skill Spec: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Trigger Set: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Requirement Edge: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Precondition Gate: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Version Lineage: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: مهارة ترحيل النطاق تحدد old/new values، ومسح المستودعات، واستثناء binaries، واختبارات الروابط، وRollback قبل التشغيل. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Skill Spec ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Trigger Set تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Requirement Edge تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Precondition Gate تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Version Lineage تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Trigger Set وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Requirement Edge بتغيير Policy، ولا يستطيع Precondition Gate ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif HyperSkill Graph (MHSG)

فرضية القياس
تقبل قيمة MHSG (Moussaif HyperSkill Graph) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- دقة اختيار المهارة: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- نجاح التحقق: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- إعادة الاستخدام: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- تعارض النسخ: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- زمن الاكتشاف: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من دقة اختيار المهارة, نجاح التحقق, إعادة الاستخدام دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R12] [R07] [R06].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Trigger واسع وشرط مفقود ونجاح بلا اختبار، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MHSG (Moussaif HyperSkill Graph) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif HyperSkill Graph (MHSG)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif HyperSkill Graph (MHSG) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Trigger واسع: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- شرط مفقود: قارن بتفسير أبسط قبل إضافة مكون جديد.
- نجاح بلا اختبار: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- نسخ مكررة: افحص الضرر غير المقصود والSide Effects.
- Skill تحمل سرًا أو مسارًا محليًا: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R35] [R12] [R07] [R06] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif HyperSkill Graph (MHSG). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Skill تحمل سرًا أو مسارًا محليًا بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif HyperRouting (MHR)

الأطروحة المركزية
اختيار المهارة أو النموذج أو الأداة قرار واحد متعدد الأبعاد يوازن الصلة والمنفعة والتوافق والدليل والتاريخ ضد الكلفة والمخاطر واستهلاك السياق.

يقترح فصل Moussaif HyperRouting (MHR) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يولد Router مرشحين، ويطبق Hard Gates، ثم يحسب Score قابلاً للتفسير، ويختار Portfolio أو مسارًا واحدًا، ويراقب Regret بعد النتيجة. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Candidate Generator وCompatibility Gate بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Utility Score وRisk Penalty حين تسمح المهمة والسلطة.
- يبقى Routing Regret حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يختار Calculator للحساب، ونموذجًا صغيرًا للتصنيف، ونموذج reasoning لمحكمة صعبة، ولا يرسل كل مهمة إلى أغلى Model. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R35] [R20] [R12] [R07] [R06].

تمييزات مفهومية تمنع التضخم

- التوجيه الفائق المساعفي ليست مرادفًا لـCandidate Generator؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Compatibility Gate لا يثبت أثر Utility Score؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Risk Penalty في عينة يثبت صلاحية محلية، ولا يمنح Routing Regret حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Candidate Generator أو Compatibility Gate معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif HyperRouting (MHR)

قانون الحركة
يولد Router مرشحين، ويطبق Hard Gates، ثم يحسب Score قابلاً للتفسير، ويختار Portfolio أو مسارًا واحدًا، ويراقب Regret بعد النتيجة.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B>_t = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـCandidate Generator مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Compatibility Gate واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Utility Score وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Risk Penalty ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Top-1 success وRouting regret ومقارنته بـBaseline.
- عرض النتيجة على Routing Regret، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Top-1 success أو Routing regret بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R20] [R12] [R07] [R06].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Candidate Generator فالمشكلة في الإدراك أو Schema، وإذا فشل Compatibility Gate مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Utility Score فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Top-1 success فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif HyperRouting (MHR)

عقدة المعمارية
التوجيه الفائق المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Candidate Generator: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Compatibility Gate: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Utility Score: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Risk Penalty: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Routing Regret: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يختار Calculator للحساب، ونموذجًا صغيرًا للتصنيف، ونموذج reasoning لمحكمة صعبة، ولا يرسل كل مهمة إلى أغلى Model. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Candidate Generator ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Compatibility Gate تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Utility Score تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Risk Penalty تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Routing Regret تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Compatibility Gate وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Utility Score بتغيير Policy، ولا يستطيع Risk Penalty ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif HyperRouting (MHR)

فرضية القياس
تقبل قيمة MHR (Moussaif HyperRouting) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Top-1 success: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Routing regret: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Latency: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Cost: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Fallback rate: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Top-1 success, Routing regret, Latency دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R20] [R12] [R07] [R06].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل توجيه حسب حجم النموذج وScore غير معاير وتاريخ نجاح مضلل، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للـRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر MHR (Moussaif HyperRouting) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif HyperRouting (MHR)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif HyperRouting (MHR) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- توجيه حسب حجم النموذج: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Score غير معاير: قارن بتفسير أبسط قبل إضافة مكون جديد.
- تاريخ نجاح مضلل: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- إهمال كلفة التنسيق: افحص الضرر غير المقصود والSide Effects.
- Fallback دائري: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R35] [R20] [R12] [R07] [R06] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif HyperRouting (MHR). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Fallback دائري بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Bounded HyperLoop (MBHL)

الأطروحة المركزية
لا تكون الحلقة ذكية لمجرد التكرار؛ يجب أن يمتلك كل Iteration فرضية جديدة أو دليلاً جديدًا أو تغيرًا في الخطة، مع Budget وشروط توقف وتصعيد.

يقترح فصل Moussaif Bounded HyperLoop (MBHL) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تسير الحلقة Observe-Question-Plan-Act-Verify-Learn، وتقيس Delta؛ فإذا غاب التقدم مرتين تخفض النطاق أو تغير الأداة أو تتوقف. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Loop State وDelta Test بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Iteration Budget وStagnation Detector حين تسمح المهمة والسلطة.
- يبقى Escalation Path حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: في Debugging، لا يعاد الاختبار نفسه بلا تعديل؛ كل دورة تسجل فرضية وPatch واختبارًا ونتيجة، وإلا يوقفها الحارس. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R29] [R11] [R10] [R09] [R05].

تمييزات مفهومية تمنع التضخم

- الحلقة الفائقة المحدودة المساعفية ليست مرادفًا لـLoop State؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Delta Test لا يثبت أثر Iteration Budget؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Stagnation Detector في عينة يثبت صلاحية محلية، ولا يمنح Escalation Path حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Loop State أو Delta Test معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Bounded HyperLoop (MBHL)

قانون الحركة
تسير الحلقة Observe-Question-Plan-Act-Verify-Learn، وتقيس Delta؛ فإذا غاب التقدم مرتين تخفض النطاق أو تغير الأداة أو تتوقف.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـLoop State مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Delta Test واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Iteration Budget وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Stagnation Detector ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Goal delta وEvidence gain ومقارنته بـBaseline.
- عرض النتيجة على Escalation Path، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Goal delta أو Evidence gain بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R29] [R11] [R10] [R09] [R05].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Loop State فالمشكلة في الإدراك أو Schema، وإذا فشل Delta Test مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Iteration Budget فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Goal delta فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Bounded HyperLoop (MBHL)

عقدة المعمارية
الحلقة الفائقة المحدودة المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Loop State: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Delta Test: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Iteration Budget: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Stagnation Detector: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Escalation Path: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: في Debugging، لا يعاد الاختبار نفسه بلا تعديل؛ كل دورة تسجل فرضية وPatch واختبارًا ونتيجة، وإلا يوقفها الحارس. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Loop State ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Delta Test تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Iteration Budget تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Stagnation Detector تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Escalation Path تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Delta Test وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Iteration Budget بتغيير Policy، ولا يستطيع Stagnation Detector ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Bounded HyperLoop (MBHL)

فرضية القياس
تقبل قيمة Moussaif Bounded HyperLoop (MBHL) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Goal delta: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Evidence gain: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- تكرار الأفعال: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- تكلفة الدورة: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- صحة التوقف: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Goal delta, Evidence gain, تكرار الأفعال دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R29] [R11] [R10] [R09] [R05].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Infinite loop وReflection theater وتكرار API، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر Moussaif Bounded HyperLoop (MBHL) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Bounded HyperLoop (MBHL)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Bounded HyperLoop (MBHL) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Infinite loop: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Reflection theater: قارن بتفسير أبسط قبل إضافة مكون جديد.
- تكرار API: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- نجاح محلي يضر الهدف: افحص الضرر غير المقصود والـSide Effects.
- توقف مبكر: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R29] [R11] [R10] [R09] [R05] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Bounded HyperLoop (MBHL). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر توقف مبكر بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Tool Intelligence (MTI)

الأطروحة المركزية
القدرة على استخدام الأداة تشمل معرفة متى لا تستخدمها، وصياغة Arguments صحيحة، والتحقق من Output، وفهم أثرها الجانبي وحدود صلاحيتها.

يقترح فصل MTI (Moussaif Tool Intelligence) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يمر Tool Call عبر Schema Validation وPermission Check وDry Run عند الإمكان، ثم Output Validation وSide-Effect Ledger. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Tool Catalog وArgument Schema بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Permission Token وDry-Run Adapter حين تسمح المهمة والسلطة.
- يبقى Side-Effect Ledger حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: قبل تعديل ملف، يقرأ الوكيل النطاق، ثم يطبق Patch محدودًا، ويفحص Diff، بدل كتابة الملف كاملاً وإتلاف تغييرات المستخدم. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R37] [R25] [R07] [R05].

تمييزات مفهومية تمنع التضخم

- ذكاء الأدوات المساعفي ليست مرادفًا لTool Catalog؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Argument Schema لا يثبت أثر Permission Token؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Dry-Run Adapter في عينة يثبت صلاحية محلية، ولا يمنح Side-Effect Ledger حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Tool Catalog أو Argument Schema معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Tool Intelligence (MTI)

قانون الحركة
يمر Tool Call عبر Schema Validation وPermission Check وDry Run عند الإمكان، ثم Output Validation وSide-Effect Ledger.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $S_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـTool Catalog مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Argument Schema واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Permission Token وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Dry-Run Adapter ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Call validity وTool success ومقارنته بـBaseline.
- عرض النتيجة على Side-Effect Ledger، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

$$\Delta\Omega_t = \text{Observe}(\text{Action}(\text{Plan}(Q(\text{State}_t)))) - \text{State}_t$$

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Call validity أو Tool success بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R37] [R25] [R07] [R05].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Tool Catalog فالمشكلة في الإدراك أو Schema؛ وإذا فشل Argument Schema مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Permission Token فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Call validity فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Tool Intelligence (MTI)

عقدة المعمارية
ذكاء الأدوات المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Tool Catalog: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Argument Schema: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Permission Token: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Dry-Run Adapter: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Side-Effect Ledger: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: قبل تعديل ملف، يقرأ الوكيل النطاق، ثم يطبق Patch محدودًا، ويفحص Diff، بدل كتابة الملف كاملاً وإتلاف تغييرات المستخدم. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Tool Catalog ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Argument Schema تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Permission Token تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Dry-Run Adapter تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Side-Effect Ledger تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Argument Schema وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Permission Token بتغيير Policy، ولا يستطيع Dry-Run Adapter ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Tool Intelligence (MTI)

فرضية القياس
تقبل قيمة MTI (Moussaif Tool Intelligence) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق، لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Call validity: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Tool success: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Side-effect error: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Unnecessary calls: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Recovery time: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Call validity, Tool success, Side-effect error دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R37] [R25] [R07] [R05].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل أداة خاطئة وArguments مولدة وOutput موثوق تلقائيًا، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر MTI (Moussaif Tool Intelligence) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Tool Intelligence (MTI)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Tool Intelligence (MTI) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- أداة خاطئة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Arguments مولدة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- Output موثوق تلقائيًا: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- أثر غير قابل للعكس: افحص الضرر غير المقصود والSide Effects.
- Tool injection: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R37] [R25] [R07] [R05] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Tool Intelligence (MTI). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Tool injection بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Temporal Plan Lattice (MTPL)

الأطروحة المركزية
الخطة ليست قائمة خطية؛ إنها شبكة اعتماديات ومسارات بديلة ونقاط قرار، تربط كل خطوة بدليل جاهزيتها وبمخرج قابل للتحقق.

يقترح فصل Moussaif Temporal Plan Lattice (MTPL) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يفكك Planner الهدف إلى Milestones، ويبني DAG، ويحدد Critical Path، ويترك بدائل مشروطة، ويعيد التخطيط من Delta لا من الصفر. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Milestone DAG وDependency Gate بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Critical Path وAlternative Branch حين تسمح المهمة والسلطة.
- يبقى Replan Trigger حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: إنشاء كتاب يبدأ بالمحتوى والخط والهيكل، ثم التوليد، ثم PDF، ثم فحص العدد والرندر؛ لا يبدأ بالحفظ قبل التحقق. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R21] [R17] [R11] [R05].

تمييزات مفهومية تمنع التضخم

- شبكة الخطة الزمنية المساعفية ليست مرادفًا لـMilestone DAG؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Dependency Gate لا يثبت أثر Critical Path؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Alternative Branch في عينة يثبت صلاحية محلية، ولا يمنح Replan Trigger حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Milestone DAG أو Dependency Gate معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Temporal Plan Lattice (MTPL)

قانون الحركة
يفكك Planner الهدف إلى Milestones، ويبني DAG، ويحدد Critical Path، ويترك بدائل مشروطة، ويعيد التخطيط من Delta لا من الصفر.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـMilestone DAG مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Dependency Gate واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Critical Path وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Alternative Branch ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Plan validity وDependency violations ومقارنته بـBaseline.
- عرض النتيجة على Replan Trigger، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Plan validity أو Dependency violations بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R21] [R17] [R11] [R05].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Milestone DAG فالمشكلة في الإدراك أو Schema، وإذا فشل Dependency Gate مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Critical Path فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Plan validity فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Temporal Plan Lattice (MTPL)

عقدة المعمارية
شبكة الخطة الزمنية المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Milestone DAG: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Dependency Gate: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Critical Path: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Alternative Branch: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Replan Trigger: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: إنشاء كتاب يبدأ بالمحتوى والخط والهيكل، ثم التوليد، ثم PDF، ثم فحص العدد والرندر؛ لا يبدأ بالحفظ قبل التحقق. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Milestone DAG ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Dependency Gate تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Critical Path تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Alternative Branch تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Replan Trigger تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Dependency Gate وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Critical Path بتغيير Policy، ولا يستطيع Alternative Branch ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Temporal Plan Lattice (MTPL)

فرضية القياس
تقبل قيمة Moussaif Temporal Plan Lattice (MTPL) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Plan validity: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Dependency violations: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Replan cost: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Milestone yield: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Horizon error: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Plan validity, Dependency violations, Replan cost دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R21] [R17] [R11] [R05].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل خطوات بلا تبعية وخطة مثالية ثابتة وParallelism زائف، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر Moussaif Temporal Plan Lattice (MTPL) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Temporal Plan Lattice (MTPL)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Temporal Plan Lattice (MTPL) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- خطوات بلا تبعية: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- خطة مثالية ثابتة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- Parallelism زائف: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- غياب Checkpoint: افحص الضرر غير المقصود والSide Effects.
- تفكيك دقيق يقتل المرونة: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R21] [R17] [R11] [R05] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Temporal Plan Lattice (MTPL). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر تفكيك دقيق يقتل المرونة بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Adaptive Execution Control (MAEC)

الأطروحة المركزية
التنفيذ الآمن يربط كل فعل بملاحظة حديثة، ويتكيف مع الفشل الجزئي، ويحافظ على Idempotency وRollback بدل افتراض عالم ثابت.

يقترح فصل Moussaif Adaptive Execution Control (MAEC) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يطبق Executor Preflight، وينفذ أصغر تغير، ويقرأ Telemetry، ويقارن Expected/Observed، ثم يثبت أو يتراجع أو يصعد. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Preflight وMinimal Mutation بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Idempotency Key وTelemetry Reader حين تسمح المهمة والسلطة.
- يبقى Rollback Action حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: في نشر موقع، يتحقق من Build وPreview قبل Production، ويسجل معرف النشر، ويحتفظ بإصدار يمكن الرجوع إليه. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R37] [R27] [R25] [R17].

تمييزات مفهومية تمنع التضخم

- التحكم التكيفي في التنفيذ المساعفي ليست مرادفًا لـPreflight؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Minimal Mutation لا يثبت أثر Idempotency Key؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Telemetry Reader في عينة يثبت صلاحية محلية، ولا يمنح Rollback Action حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Preflight أو Minimal Mutation معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Adaptive Execution Control (MAEC)

قانون الحركة
يطبق Executor Preflight، وينفذ أصغر تغير، ويقرأ Telemetry، ويقارن Expected/Observed، ثم يثبت أو يتراجع أو يصعد.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـPreflight مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Minimal Mutation واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Idempotency Key وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Telemetry Reader ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Execution success وRollback success ومقارنته بـBaseline.
- عرض النتيجة على Rollback Action، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Execution success أو Rollback success بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R37] [R27] [R25] [R17].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Preflight فالمشكلة في الإدراك أو Schema؛ وإذا فشل Minimal Mutation مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Idempotency Key فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Execution success فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Adaptive Execution Control (MAEC)

عقدة المعمارية
التحكم التكيفي في التنفيذ المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Preflight: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Minimal Mutation: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Idempotency Key: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Telemetry Reader: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Rollback Action: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: في نشر موقع، يتحقق من Build وPreview قبل Production، ويسجل معرف النشر، ويحتفظ بإصدار يمكن الرجوع إليه. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Preflight ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Minimal Mutation تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Idempotency Key تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Telemetry Reader تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Rollback Action تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Minimal Mutation وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Idempotency Key بتغيير Policy، ولا يستطيع Telemetry Reader ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Adaptive Execution Control (MAEC)

فرضية القياس
تقبل قيمة Moussaif Adaptive Execution Control (MAEC) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Execution success: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Rollback success: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Duplicate effects: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Mean recovery: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Drift: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Execution success, Rollback success, Duplicate effects دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R37] [R27] [R25] [R17].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل إعادة غير آمنة ونجاح HTTP كاذب وتغير البيئة، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Adaptive Execution Control (MAEC) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Adaptive Execution Control (MAEC)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Adaptive Execution Control (MAEC) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- إعادة غير آمنة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- نجاح HTTP كاذب: قارن بتفسير أبسط قبل إضافة مكون جديد.
- تغير البيئة: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- Rollback غير مختبر: افحص الضرر غير المقصود والSide Effects.
- تجاهل الفشل الجزئي: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R37] [R27] [R25] [R17] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Adaptive Execution Control (MAEC). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر تجاهل الفشل الجزئي بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif HyperSubagent Orchestration (MHSO)

الأطروحة المركزية
تعدد الوكلاء مفيد فقط عندما يكون التقسيم مستقلاً وقابلاً للدمج، وتكون كلفة الاتصال أقل من مكسب التخصص أو التوازي أو النقد المتبادل.

يقترح فصل MHSO (Moussaif HyperSubagent Orchestration) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يقسم Orchestrator العمل إلى Bounded Tasks، ويمنح كل وكيل Context Capsule، ويحدد Merge Contract، ويمنع الكتابة المتزامنة المتعارضة. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Task Boundary وAgent Charter بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Context Capsule وShared Artifact حين تسمح المهمة والسلطة.
- يبقى Conflict Resolver حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يمكن لوكيل بحث جمع المراجع بالتوازي مع وكيل QA يفحص مخطط الصفحات، شرط ألا يعدلا المصدر ذاته وأن يسلما نتائج منظمة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R21] [R20] [R14] [R13] [R12].

تمييزات مفهومية تمنع التضخم

- تنسيق HyperSubagents المساعفي ليست مرادفًا لـTask Boundary؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Agent Charter لا يثبت أثر Context Capsule؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Shared Artifact في عينة يثبت صلاحية محلية، ولا يمنح Conflict Resolver حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Task Boundary أو Agent Charter معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif HyperSubagent Orchestration (MHSO)

قانون الحركة
يقسم Orchestrator العمل إلى Bounded Tasks، ويمنح كل وكيل Context Capsule، ويحدد Merge Contract، ويمنع الكتابة المتزامنة المتعارضة.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـTask Boundary مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Agent Charter واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Context Capsule وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Shared Artifact ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Parallel speedup وMerge defects ومقارنته بـBaseline.
- عرض النتيجة على Conflict Resolver، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Parallel speedup أو Merge defects بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R12] [R13] [R14] [R20] [R21].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Task Boundary فالمشكلة في الإدراك أو Schema؛ وإذا فشل Agent Charter مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Context Capsule فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Parallel speedup فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif HyperSubagent Orchestration (MHSO)

عقدة المعمارية
تنسيق HyperSubagents المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Task Boundary: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Agent Charter: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Context Capsule: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Shared Artifact: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Conflict Resolver: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يمكن لوكيل بحث جمع المراجع بالتوازي مع وكيل QA يفحص مخطط الصفحات، شرط ألا يعدلا المصدر ذاته وأن يسلما نتائج منظمة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Task Boundary ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Agent Charter تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Context Capsule تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Shared Artifact تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Conflict Resolver تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Agent Charter وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Context Capsule بتغيير Policy، ولا يستطيع Shared Artifact ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif HyperSubagent Orchestration (MHSO)

فرضية القياس
تقبل قيمة MHSO (Moussaif HyperSubagent Orchestration) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Parallel speedup: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Merge defects: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Communication cost: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Coverage gain: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Duplicate work: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Parallel speedup, Merge defects, Communication cost دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R20] [R14] [R13] [R12] [R21].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تفويض غامض ووكلاء على الملف نفسه وإجابات غير قابلة للدمج، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MHSO (Moussaif HyperSubagent Orchestration) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif HyperSubagent Orchestration (MHSO)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif HyperSubagent Orchestration (MHSO) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تفويض غامض: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- وكلاء على الملف نفسه: قارن بتفسير أبسط قبل إضافة مكون جديد.
- إجابات غير قابلة للدمج: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- Consensus theater: افحص الضرر غير المقصود والSide Effects.
- تكلفة أكبر من العمل: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R21] [R20] [R14] [R13] [R12] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif HyperSubagent Orchestration (MHSO). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر تكلفة أكبر من العمل بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Complementarity Protocol (MCoP)

الأطروحة المركزية
الاختلاف بين الوكلاء يجب أن يكون اختلاف وظيفة أو دليل أو منظور، لا مجرد تغيير أسماء الأدوار مع إعادة النموذج نفسه للجواب نفسه.

يقترح فصل Moussaif Complementarity Protocol (MCoP) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يختار Role Designer أدوارًا متعامدة، ويحدد ما لا يراه كل دور، ويقيس Marginal Contribution، ثم يحذف الدور الذي لا يضيف فرقًا. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Role Orthogonality وEvidence Partition بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Blindness Contract وContribution Test حين تسمح المهمة والسلطة.
- يبقى Role Pruner حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: Builder يقترح المعمارية، Security Reviewer يبحث عن مسار هجوم، Evaluator يختبر Benchmarks؛ لا يكرر الثلاثة وصف المعمارية. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R30] [R21] [R20] [R12].

تمييزات مفهومية تمنع التضخم

- بروتوكول التكامل والتكاملية المساعفي ليست مرادفًا لـRole Orthogonality؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Evidence Partition لا يثبت أثر Blindness Contract؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Contribution Test في عينة يثبت صلاحية محلية، ولا يمنح Role Pruner حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Role Orthogonality أو Evidence Partition معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Complementarity Protocol (MCoP)

قانون الحركة
يختار Role Designer أدوارًا متعامدة، ويحدد ما لا يراه كل دور، ويقيس Marginal Contribution، ثم يحذف الدور الذي لا يضيف فرقًا.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = t_، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـRole Orthogonality مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Evidence Partition واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Blindness Contract وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Contribution Test ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Marginal gain وRedundancy ومقارنته بـBaseline.
- عرض النتيجة على Role Pruner، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Marginal gain أو Redundancy بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R30] [R21] [R20] [R12].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Role Orthogonality فالمشكلة في الإدراك أو Schema؛ وإذا فشل Evidence Partition مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Blindness Contract فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Marginal gain فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Complementarity Protocol (MCoP)

عقدة المعمارية
بروتوكول التكامل والتكاملية المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Role Orthogonality: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Evidence Partition: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Blindness Contract: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Contribution Test: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Role Pruner: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: Builder يقترح المعمارية، Security Reviewer يبحث عن مسار هجوم، Evaluator يختبر Benchmarks؛ لا يكرر الثلاثة وصف المعمارية. وعليه لا يوثق النجاح باسم المزود، بل بالContract والObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Role Orthogonality ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Evidence Partition تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Blindness Contract تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Contribution Test تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Role Pruner تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Evidence Partition وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Blindness Contract بتغيير Policy، ولا يستطيع Contribution Test ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Complementarity Protocol (MCoP)

فرضية القياس
تقبل قيمة Moussaif Complementarity Protocol (MCoP) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Marginal gain: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Redundancy: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Disagreement utility: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Handoff loss: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Role count efficiency: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Marginal gain, Redundancy, Disagreement utility دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R30] [R21] [R20] [R12].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل أدوار زخرفية واستقلال وهمي واختلاف بلا حكم، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Complementarity Protocol (MCoP) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Complementarity Protocol (MCoP)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Complementarity Protocol (MCoP) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- أدوار زخرفية: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- استقلال وهمي: قارن بتفسير أبسط قبل إضافة مكون جديد.
- اختلاف بلا حكم: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- فرط التنسيق: افحص الضرر غير المقصود والSide Effects.
- تحيز مشترك: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R30] [R21] [R20] [R12] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Complementarity Protocol (MCoP). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر تحيز مشترك بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Independent-Court Protocol (MICP)

الأطروحة المركزية
لا قيمة لمحكمة نقد إذا ورثت خطة الباني وسياقه وثقته بلا فصل؛ الاستقلال يحتاج Blind Inputs ومعيار حكم سابقًا وسجل اختلاف.

يقترح فصل MICP (Moussaif Independent-Court Protocol) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تجرد Court Adapter المقترح إلى Claims وArtifacts، وتحجب مبررات غير لازمة، وتثبت Rubric قبل النتيجة، ثم تقارن أحكامًا مستقلة. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Blind Packet وFrozen Rubric بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Independent Sample وDisagreement Matrix حين تسمح المهمة والسلطة.
- يبقى Appeal Record حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يفحص مقيم PDF الصفحات المرندرة وعددها ومقتطفات النص دون أن يكتفي بتقرير المولد أن الملف صحيح. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R10] [R24] [R29] [R30].

تمييزات مفهومية تمنع التضخم

- استقلال المحاكم المساعفية ليست مرادفًا لـBlind Packet؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Frozen Rubric لا يثبت أثر Independent Sample؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Disagreement Matrix في عينة يثبت صلاحية محلية، ولا يمنح Appeal Record حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Blind Packet أو Frozen Rubric معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Independent-Court Protocol (MICP)

قانون الحركة
تجرد Court Adapter المقترح إلى Claims وArtifacts، وتحجب مبررات غير لازمة، وتثبت Rubric قبل النتيجة، ثم تقارن أحكامًا مستقلة.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B>_t = <، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـBlind Packet مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Frozen Rubric واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Independent Sample وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Disagreement Matrix ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Inter-rater agreement وError discovery ومقارنته بـBaseline.
- عرض النتيجة على Appeal Record، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Inter-rater agreement أو Error discovery بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R30] [R29] [R24] [R10].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Blind Packet فالمشكلة في الإدراك أو Schema؛ وإذا فشل Frozen Rubric مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Independent Sample فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Inter-rater agreement فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Independent-Court Protocol (MICP)

عقدة المعمارية
استقلال المحاكم المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Blind Packet: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Frozen Rubric: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Independent Sample: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Disagreement Matrix: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Appeal Record: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يفحص مقيم PDF الصفحات المرندرة وعددها ومقتطفات النص دون أن يكتفي بتقرير المولد أن الملف صحيح. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Blind Packet ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Frozen Rubric تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Independent Sample تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Disagreement Matrix تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Appeal Record تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Frozen Rubric وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Independent Sample بتغيير Policy، ولا يستطيع Disagreement Matrix ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Independent-Court Protocol (MICP)

فرضية القياس
تقبل قيمة MICP (Moussaif Independent-Court Protocol) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Inter-rater agreement: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Error discovery: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Bias leakage: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Verdict stability: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Appeal reversal: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Inter-rater agreement, Error discovery, Bias leakage دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R30] [R29] [R24] [R10].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تسريب الحل وRubric بعدي وقاض واحد، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للـRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MICP (Moussaif Independent-Court Protocol) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Independent-Court Protocol (MICP)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Independent-Court Protocol (MICP) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تسريب الحل: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Rubric بعدي: قارن بتفسير أبسط قبل إضافة مكون جديد.
- قاض واحد: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- إجماع بسبب نفس Prompt: افحص الضرر غير المقصود والSide Effects.
- رفض جديد بلا سبب: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R30] [R29] [R24] [R10] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Independent-Court Protocol (MICP). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر رفض جديد بلا سبب بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Model-Portfolio Router (MMPR)

الأطروحة المركزية
لا يوجد Model أمثل لكل المهام؛ النظام الأذكى يطابق قدرات النموذج وسياقه وكلفته وحدوده مع نوع الخطأ الذي لا تحتمله المهمة.

يقترح فصل Moussaif Model-Portfolio Router (MMPR) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يحتفظ Capability Registry بقياسات محلية، ويطبق Hard Constraints، ويختار نموذجًا أو Ensemble، ويحدث التقديرات من النتائج لا من الدعاية. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Capability Card وContext Limit بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Cost Curve وLatency Profile حين تسمح المهمة والسلطة.
- يبقى Empirical Router حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: تستخدم مهام التصنيف البسيطة نموذجًا صغيرًا، بينما تمنح مهمة هدم نظرية لنموذج Reasoning مع أداة بحث وقاض مستقل. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R36] [R35] [R04] [R03] [R01].

تمييزات مفهومية تمنع التضخم

- موجه محفظة النماذج المساعفي ليست مرادفًا لـCapability Card؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Context Limit لا يثبت أثر Cost Curve؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Latency Profile في عينة يثبت صلاحية محلية، ولا يمنح Empirical Router حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Capability Card أو Context Limit معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Model-Portfolio Router (MMPR)

قانون الحركة
يحتفظ Capability Registry بقياسات محلية، ويطبق Hard Constraints، ويختار نموذجًا أو Ensemble، ويحدث التقديرات من النتائج لا من الدعاية.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B>_t = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـCapability Card مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Context Limit واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Cost Curve وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Latency Profile ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Task success وCost per verified result ومقارنته بـBaseline.
- عرض النتيجة على Empirical Router، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Task success أو Cost per verified result بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R36] [R35] [R04] [R03] [R01].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Capability Card فالمشكلة في الإدراك أو Schema، وإذا فشل Context Limit مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Cost Curve فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Task success فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Model-Portfolio Router (MMPR)

عقدة المعمارية
موجه محفظة النماذج المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Capability Card: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Context Limit: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Cost Curve: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Latency Profile: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Empirical Router: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: تستخدم مهام التصنيف البسيطة نموذجًا صغيرًا، بينما تمنح مهمة هدم نظرية لنموذج Reasoning مع أداة بحث وقاض مستقل. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Capability Card ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Context Limit تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Cost Curve تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Latency Profile تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Empirical Router تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Context Limit وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Cost Curve بتغيير Policy، ولا يستطيع Latency Profile ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Model-Portfolio Router (MMPR)

فرضية القياس
تقبل قيمة MMPR (Moussaif Model-Portfolio Router) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Task success: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Cost per verified result: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Latency: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Calibration: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Provider concentration: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Task success, Cost per verified result, Latency دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R01] [R03] [R04] [R35] [R36].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل اختيار بالشهرة وأرقام قديمة وContext معلن غير فعلي، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر MMPR (Moussaif Model-Portfolio Router) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Model-Portfolio Router (MMPR)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Model-Portfolio Router (MMPR) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- اختيار بالشهرة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- أرقام قديمة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- Context معلن غير فعلي: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- Ensemble بلا تنوع: افحص الضرر غير المقصود والSide Effects.
- Lock-in: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R36] [R35] [R04] [R03] [R01] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Model-Portfolio Router (MMPR). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Lock-in بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Token-Economy Law (MTEL)

الأطروحة المركزية
الهدف ليس أقل Tokens مطلقًا، بل أعلى قيمة موثقة لكل وحدة حساب، مع عدم تجويع الأدلة أو السلامة أو التحقق.

يقترح فصل Moussaif Token-Economy Law (MTEL) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يقيس Token Ledger كلفة كل مرحلة، ويستبدل الأعمال الحتمية بكود، ويضغط السياق، ويعيد استخدام Skills، ويرفع الميزانية عند ارتفاع قيمة الخطأ. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Token Ledger وDeterministic Offload بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Context Cache وSkill Reuse حين تسمح المهمة والسلطة.
- يبقى Budget Escalator حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يعد البرنامج الصفحات ويفحص الميتاداتا حتميًا، ويترك للنموذج تركيب النظريات والنقد حيث لا تكفي قاعدة ثابتة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R35] [R19] [R08] [R07].

تمييزات مفهومية تمنع التضخم

- قانون اقتصاد الرموز المساعفي ليست مرادفًا لـToken Ledger؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Deterministic Offload لا يثبت أثر Context Cache؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Skill Reuse في عينة يثبت صلاحية محلية، ولا يمنح Budget Escalator حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Token Ledger أو Deterministic Offload معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Token-Economy Law (MTEL)

قانون الحركة
يقيس Token Ledger كلفة كل مرحلة، ويستبدل الأعمال الحتمية بكود، ويضغط السياق، ويعيد استخدام Skills، ويرفع الميزانية عند ارتفاع قيمة الخطأ.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = □_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـToken Ledger مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Deterministic Offload واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Context Cache وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Skill Reuse ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Verified value/token وCache hit ومقارنته بـBaseline.
- عرض النتيجة على Budget Escalator، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Verified value/token أو Cache hit بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R19] [R08] [R07].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Token Ledger فالمشكلة في الإدراك أو Schema؛ وإذا فشل Deterministic Offload مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Context Cache فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Verified value/token فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Token-Economy Law (MTEL)

عقدة المعمارية
قانون اقتصاد الرموز المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Token Ledger: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Deterministic Offload: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Context Cache: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Skill Reuse: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Budget Escalator: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يعد البرنامج الصفحات ويفحص الميتاداتا حتميًا، ويترك للنموذج تركيب النظريات والنقد حيث لا تكفي قاعدة ثابتة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Token Ledger ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Deterministic Offload تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Context Cache تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Skill Reuse تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Budget Escalator تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Deterministic Offload وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Context Cache بتغيير Policy، ولا يستطيع Skill Reuse ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Token-Economy Law (MTEL)

فرضية القياس
تقبل قيمة Moussaif Token-Economy Law (MTEL) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Verified value/token: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Cache hit: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Rework cost: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Quality floor violations: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Total cost: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Verified value/token, Cache hit, Rework cost دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R19] [R08] [R07].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل ضغط يضيع القرار ونموذج صغير في مهمة حرجة وحفظ Context سام، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Token-Economy Law (MTEL) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Token-Economy Law (MTEL)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Token-Economy Law (MTEL) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- ضغط يضيع القرار: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- نموذج صغير في مهمة حرجة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- حفظ Context سام: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- حساب الكلفة دون مراجعة: افحص الضرر غير المقصود والSide Effects.
- Optimization محلي: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R35] [R19] [R08] [R07] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Token-Economy Law (MTEL). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Optimization محلي بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Rate-Limit Resilience (MRLR)

الأطروحة المركزية
Rate Limit جزء من حالة النظام، لا حادث خارجي؛ يجب أن يخطط الوكيل للتدفق والـBackoff والـFallback دون تكرار آثار جانبية أو انفجار كلفة.

يقترح فصل MRLR) Moussaif Rate-Limit Resilience) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يقرأ Governor الحصص، ويجدول Requests، ويجمع Batch، ويطبق Exponential Backoff مع Jitter، وينقل المهمة إلى بديل متوافق عند حد محدد. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Quota State وRequest Queue بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Batch Planner وBackoff Policy حين تسمح المهمة والسلطة.
- يبقى Idempotency Key حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: عند 40 RPM، يحجز النظام هامشًا للتحقق، ويمنع 40 وكيلاً من الاستدعاء اللحظي، ويجمع الطلبات القصيرة إن سمحت الواجهة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R35] [R27] [R25] [R17].

تمييزات مفهومية تمنع التضخم

- مرونة الحدود ومعدل الاستدعاء المساعفية ليست مرادفًا لـQuota State؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Request Queue لا يثبت أثر Batch Planner؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Backoff Policy في عينة يثبت صلاحية محلية، ولا يمنح Idempotency Key حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Quota State أو Request Queue معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Rate-Limit Resilience (MRLR)

قانون الحركة
يقرأ Governor الحصص، ويجدول Requests، ويجمع Batch، ويطبق Exponential Backoff مع Jitter، وينقل المهمة إلى بديل متوافق عند حد محدد.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = □_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـQuota State مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Request Queue واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Batch Planner وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Backoff Policy ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري 429 rate وQueue delay ومقارنته بـBaseline.
- عرض النتيجة على Idempotency Key، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في 429 rate أو Queue delay بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R27] [R25] [R17].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Quota State فالمشكلة في الإدراك أو Schema، وإذا فشل Request Queue مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Batch Planner فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل 429 rate فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Rate-Limit Resilience (MRLR)

عقدة المعمارية
مرونة الحدود ومعدل الاستدعاء المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Quota State: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Request Queue: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Batch Planner: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Backoff Policy: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Idempotency Key: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: عند 40 RPM، يحجز النظام هامشًا للتحقق، ويمنع 40 وكيلاً من الاستدعاء اللحظي، ويجمع الطلبات القصيرة إن سمحت الواجهة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Quota State ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Request Queue تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Batch Planner تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Backoff Policy تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Idempotency Key تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Request Queue وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Batch Planner بتغيير Policy، ولا يستطيع Backoff Policy ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Rate-Limit Resilience (MRLR)

فرضية القياس
تقبل قيمة MRLR (Moussaif Rate-Limit Resilience) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- 429 rate: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Queue delay: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Duplicate effects: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Fallback quality: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Quota utilization: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من 429 rate, Queue delay, Duplicate effects دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R27] [R25] [R17].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Retry storm وFallback غير متوافق ومفاتيح موزعة عشوائيًا، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر MRLR (Moussaif Rate-Limit Resilience) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Rate-Limit Resilience (MRLR)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Rate-Limit Resilience (MRLR) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Retry storm: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Fallback غير متوافق: قارن بتفسير أبسط قبل إضافة مكون جديد.
- مفاتيح موزعة عشوائيًا: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- تجاوز شروط المزود: افحص الضرر غير المقصود والـSide Effects.
- فقد ترتيب العمل: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R35] [R27] [R25] [R17] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Rate-Limit Resilience (MRLR). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر فقد ترتيب العمل بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Epistemic Calibration (MEC)

الأطروحة المركزية
يجب أن تتناسب الثقة المعلنة مع معدل الصواب المشاهد، وأن تنفصل ثقة المحتوى عن ثقة المصدر وثقة الإجراء وثقة اكتمال التغطية.

يقترح فصل Moussaif Epistemic Calibration (MEC) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يصدر النظام Vector Confidence، ويقارن التوقع بالنتيجة، ويعيد المعايرة، ويمنع الأرقام الدقيقة إذا لم توجد عينة أو تعريف حدث. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Content Confidence وSource Confidence بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Action Confidence وCoverage Confidence حين تسمح المهمة والسلطة.
- يبقى Abstention Gate حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: قد يكون النظام واثقًا من أن الملف 250 صفحة بعد pdfinfo، لكنه أقل ثقة في اكتمال نظرية AGI لأنها دعوى بحثية مفتوحة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R36] [R26] [R24] [R22] [R03].

تمييزات مفهومية تمنع التضخم

- المعايرة الإبستمولوجية المساعفية ليست مرادفًا لـContent Confidence؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Source Confidence لا يثبت أثر Action Confidence؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Coverage Confidence في عينة يثبت صلاحية محلية، ولا يمنح Abstention Gate حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Content Confidence أو Source Confidence معروفة بأسماء أخرى، الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Epistemic Calibration (MEC)

قانون الحركة
يصدر النظام Vector Confidence، ويقارن التوقع بالنتيجة، ويعيد المعايرة، ويمنع الأرقام الدقيقة إذا لم توجد عينة أو تعريف حدث.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B>_t = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـContent Confidence مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Source Confidence واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Action Confidence وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Coverage Confidence ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري ECE وBrier score ومقارنته بـBaseline.
- عرض النتيجة على Abstention Gate، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في ECE أو Brier score بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R36] [R26] [R24] [R22] [R03].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Content Confidence فالمشكلة في الإدراك أو Schema؛ وإذا فشل Source Confidence مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Action Confidence فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل ECE فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Epistemic Calibration (MEC)

عقدة المعمارية
المعايرة الإبستمولوجية المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Content Confidence: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Source Confidence: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Action Confidence: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Coverage Confidence: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Abstention Gate: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: قد يكون النظام واثقًا من أن الملف 250 صفحة بعد pdfinfo، لكنه أقل ثقة في اكتمال نظرية AGI لأنها دعوى بحثية مفتوحة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Content Confidence ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Source Confidence تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Action Confidence تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Coverage Confidence تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Abstention Gate تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Source Confidence وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Action Confidence بتغيير Policy، ولا يستطيع Coverage Confidence ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Epistemic Calibration (MEC)

فرضية القياس
تقبل قيمة MEC (Moussaif Epistemic Calibration) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- ECE: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Brier score: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Selective accuracy: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Abstention utility: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Overconfidence rate: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من ECE, Brier score, Selective accuracy دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R36] [R26] [R24] [R22] [R03].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل نسبة ثقة ملفقة وثقة واحدة ومعايرة على توزيع مختلف، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر MEC (Moussaif Epistemic Calibration) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Epistemic Calibration (MEC)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Epistemic Calibration (MEC) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- نسبة ثقة ملفقة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- ثقة واحدة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- معايرة على توزيع مختلف: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- امتناع مفرط: افحص الضرر غير المقصود والSide Effects.
- خلط اليقين بالدقة اللغوية: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R36] [R26] [R24] [R22] [R03] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Epistemic Calibration (MEC). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر خلط اليقين بالدقة اللغوية بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Authorization Boundary (MAB)

الأطروحة المركزية
القدرة التقنية لا تمنح التفويض؛ كل فعل خارجي يجب أن يطابق Scope وActor وTarget وTime وEffect المسموح بها صراحة أو ضمنيًا بوضوح.

يقترح فصل Moussaif Authorization Boundary (MAB) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: ينشئ Authorization Compiler Capability Token محدودًا، ويفحصه قبل الفعل، ويطلب Confirmation للأثر العالي، ويلغيه عند تغير الهدف. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Actor Identity وCapability Token بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Target Resolver وEffect Class حين تسمح المهمة والسلطة.
- يبقى Revocation حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: البحث عن Leads لا يعني إرسال بريد إليهم؛ وصياغة رسالة لا تعني الضغط على Send دون اختيار المستلمين واعتماد المحتوى. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R40] [R31] [R27] [R24] [R22].

تمييزات مفهومية تمنع التضخم

- حد السلطة المساعفي ليست مرادفًا لـActor Identity؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Capability Token لا يثبت أثر Target Resolver؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Effect Class في عينة يثبت صلاحية محلية، ولا يمنح Revocation حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Actor Identity أو Capability Token معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Authorization Boundary (MAB)

قانون الحركة
ينشئ Authorization Compiler Capability Token محدودًا، ويفحصه قبل الفعل، ويطلب Confirmation للأثر العالي، ويلغيه عند تغير الهدف.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B>_t = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـActor Identity مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Capability Token واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Target Resolver وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Effect Class ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Unauthorized action وCorrect escalation ومقارنته بـBaseline.
- عرض النتيجة على Revocation، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Unauthorized action أو Correct escalation بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R31] [R27] [R24] [R22].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Actor Identity فالمشكلة في الإدراك أو Schema؛ وإذا فشل Capability Token مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Target Resolver فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Unauthorized action فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Authorization Boundary (MAB)

عقدة المعمارية
حد السلطة المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Actor Identity: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Capability Token: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Target Resolver: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Effect Class: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Revocation: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: البحث عن Leads لا يعني إرسال بريد إليهم؛ وصياغة رسالة لا تعني الضغط على Send دون اختيار المستلمين واعتماد المحتوى. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Actor Identity ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Capability Token تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Target Resolver تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Effect Class تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Revocation تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Capability Token وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Target Resolver بتغيير Policy، ولا يستطيع Effect Class ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Authorization Boundary (MAB)

فرضية القياس
تقبل قيمة MAB (Moussaif Authorization Boundary) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Unauthorized action: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Correct escalation: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Token scope: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Reversal success: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Human override: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Unauthorized action, Correct escalation, Token scope دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R31] [R27] [R24] [R22] [R40].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تفويض مستنتج ومستلم غامض وصلاحية دائمة، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MAB (Moussaif Authorization Boundary) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Authorization Boundary (MAB)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Authorization Boundary (MAB) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تفويض مستنتج: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- مستلم غامض: قارن بتفسير أبسط قبل إضافة مكون جديد.
- صلاحية دائمة: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- تجميع سلطات: افحص الضرر غير المقصود والSide Effects.
- Confirmation بعد الفعل: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R40] [R31] [R27] [R24] [R22] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Authorization Boundary (MAB). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Confirmation بعد الفعل بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Injection-Resistant Toolchain (MIRT)

الأطروحة المركزية

كل محتوى مسترجع قد يحمل تعليمات خصمية؛ يجب فصل Data Plane عن Control Plane وعدم منح النص الخارجي سلطة تغيير السياسات أو استدعاء الأدوات.

يقترح فصل MIRT (Moussaif Injection-Resistant Toolchain) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يصنف Trust Boundary المصدر، ويعزل المحتوى، ويستخرج بيانات typed، ويطبق Policy خارج النموذج، ويطلب موافقة عند انتقال الخطر. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Trust Label وContent Sandbox بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Instruction Stripper وTyped Extractor حين تسمح المهمة والسلطة.
- يبقى Secret Guard حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: إذا قالت وثيقة مسترجعة «تجاهل قواعدك»، تعامل كسلسلة بيانات داخل المصدر، ولا تصبح أمرًا في Control Plane. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R40] [R37] [R31] [R27] [R25].

تمييزات مفهومية تمنع التضخم

- سلسلة الأدوات المقاومة للحقن المساعفية ليست مرادفًا لـTrust Label؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Content Sandbox لا يثبت أثر Instruction Stripper؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Typed Extractor في عينة يثبت صلاحية محلية، ولا يمنح Secret Guard حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Trust Label أو Content Sandbox معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Injection-Resistant Toolchain (MIRT)

قانون الحركة
يصنف Trust Boundary المصدر، ويعزل المحتوى، ويستخرج بيانات typed، ويطبق Policy خارج النموذج، ويطلب موافقة عند انتقال الخطر.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $S_t = <G,W,M,K,S,T,P,E,B>$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـTrust Label مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Content Sandbox واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Instruction Stripper وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Typed Extractor ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Injection success وSecret exposure ومقارنته بـBaseline.
- عرض النتيجة على Secret Guard، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Injection success أو Secret exposure بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R37] [R31] [R27] [R25].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Trust Label فالمشكلة في الإدراك أو Schema؛ وإذا فشل Content Sandbox مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Instruction Stripper فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Injection success فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Injection-Resistant Toolchain (MIRT)

عقدة المعمارية
سلسلة الأدوات المقاومة للحقن المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Trust Label: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Content Sandbox: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Instruction Stripper: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Typed Extractor: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Secret Guard: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: إذا قالت وثيقة مسترجعة «تجاهل قواعدك»، تعامل كسلسلة بيانات داخل المصدر، ولا تصبح أمرًا في Control Plane. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Trust Label ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Content Sandbox تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Instruction Stripper تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Typed Extractor تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Secret Guard تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Content Sandbox وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Instruction Stripper بتغيير Policy، ولا يستطيع Typed Extractor ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Injection-Resistant Toolchain (MIRT)

فرضية القياس
تقبل قيمة MIRT (Moussaif Injection-Resistant Toolchain) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Injection success: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Secret exposure: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Unsafe tool call: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- False block: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Detection latency: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Injection success, Secret exposure, Unsafe tool call دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R37] [R31] [R27] [R25] [R40].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل طاعة صفحة ويب وتسريب System Prompt وأداة بمدخل خام، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MIRT (Moussaif Injection-Resistant Toolchain) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Injection-Resistant Toolchain (MIRT)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Injection-Resistant Toolchain (MIRT) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- طاعة صفحة ويب: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- تسريب System Prompt: قارن بتفسير أبسط قبل إضافة مكون جديد.
- أداة بمدخل خام: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- امتيازات زائدة: افحص الضرر غير المقصود والSide Effects.
- Log يحمل سرًا: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R40] [R37] [R31] [R27] [R25] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Injection-Resistant Toolchain (MIRT). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Log يحمل سرًا بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Data-Sovereignty Layer (MDSL)

الأطروحة المركزية
تحدد دورة البيانات ما يجوز جمعه وأين يعالج وكم يبقى ومن يستطيع استدعاءه، ولا تعوض فائدة الذاكرة غياب الموافقة أو حق الحذف.

يقترح فصل Moussaif Data-Sovereignty Layer (MDSL) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يصنف Data Mapper الحساسية والإقامة والغرض، ويطبق Minimization وEncryption وRetention، ويسجل كل انتقال بين Model وTool وStore. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Data Class وPurpose Binding بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Residency Rule وRetention Clock حين تسمح المهمة والسلطة.
- يبقى Deletion Proof حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: قبل إدخال بريد عميل في نموذج خارجي، يفحص النظام الغرض والعقد وموقع المعالجة وإمكان Redaction أو تشغيل محلي. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R40] [R37] [R31] [R27].

تمييزات مفهومية تمنع التضخم

- سيادة البيانات المساعفية ليست مرادفًا لـData Class؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Purpose Binding لا يثبت أثر Residency Rule؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Retention Clock في عينة يثبت صلاحية محلية، ولا يمنح Deletion Proof حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Data Class أو Purpose Binding معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Data-Sovereignty Layer (MDSL)

قانون الحركة
يصنف Data Mapper الحساسية والإقامة والغرض، ويطبق Minimization وEncryption وRetention، ويسجل كل انتقال بين Model وTool وStore.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـData Class مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Purpose Binding واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Residency Rule وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Retention Clock ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Data minimization وRetention compliance ومقارنته بـBaseline.
- عرض النتيجة على Deletion Proof، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Data minimization أو Retention compliance بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R37] [R31] [R27].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Data Class فالمشكلة في الإدراك أو Schema؛ وإذا فشل Purpose Binding مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Residency Rule فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Data minimization فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Data-Sovereignty Layer (MDSL)

عقدة المعمارية
سيادة البيانات المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Data Class: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Purpose Binding: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Residency Rule: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Retention Clock: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Deletion Proof: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: قبل إدخال بريد عميل في نموذج خارجي، يفحص النظام الغرض والعقد وموقع المعالجة وإمكان Redaction أو تشغيل محلي. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Data Class ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Purpose Binding تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Residency Rule تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Retention Clock تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Deletion Proof تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Purpose Binding وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Residency Rule بتغيير Policy، ولا يستطيع Retention Clock ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Data-Sovereignty Layer (MDSL)

فرضية القياس
تقبل قيمة Moussaif Data-Sovereignty Layer (MDSL) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Data minimization: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Retention compliance: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Access anomalies: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Deletion success: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Cross-boundary events: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Data minimization, Retention compliance, Access anomalies دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R37] [R31] [R27] [R40].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل جمع زائد وغرض متغير ونسخ منسية، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Data-Sovereignty Layer (MDSL) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Data-Sovereignty Layer (MDSL)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Data-Sovereignty Layer (MDSL) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- جمع زائد: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- غرض متغير: قارن بتفسير أبسط قبل إضافة مكون جديد.
- نسخ منسية: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- PII في Prompt: افحص الضرر غير المقصود والـSide Effects.
- حذف غير شامل: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R40] [R37] [R31] [R27] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Data-Sovereignty Layer (MDSL). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر حذف غير شامل بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Agentic Observability Ledger (MAOL)

الأطروحة المركزية
لا يمكن حوكمة وكيل طويل الأفق من Output النهائي؛ يلزم سجل أحداث يربط القرار بالأداة والحالة والكلفة والدليل والنتيجة مع حماية الخصوصية.

يقترح فصل MAOL (Moussaif Agentic Observability Ledger) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يصدر كل مكون Structured Event بمعرف Trace، ويجمعها Ledger، ويبني Metrics وAlerts وReplay، ويخفي الأسرار قبل التخزين. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Trace ID وDecision Event بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Tool Event وCost Event حين تسمح المهمة والسلطة.
- يبقى Redaction حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يربط فشل نشر بالـCommit والـBuild command ورسالة الخطأ وقرار Rollback، لا بعبارة عامة «لم يعمل». يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R40] [R36] [R27] [R25] [R17].

تمييزات مفهومية تمنع التضخم

- دفتر الرصد الوكيلي المساعفي ليست مرادفًا لـTrace ID؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Decision Event لا يثبت أثر Tool Event؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Cost Event في عينة يثبت صلاحية محلية، ولا يمنح Redaction حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Trace ID أو Decision Event معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Agentic Observability Ledger (MAOL)

قانون الحركة
يصدر كل مكون Structured Event بمعرف Trace، ويجمعها Ledger، ويبني Metrics وAlerts وReplay، ويخفي الأسرار قبل التخزين.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B>_t = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـTrace ID مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Decision Event واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Tool Event وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Cost Event ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Trace coverage وMean diagnosis time ومقارنته بـBaseline.
- عرض النتيجة على Redaction، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Trace coverage أو Mean diagnosis time بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R36] [R27] [R25] [R17].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Trace ID فالمشكلة في الإدراك أو Schema؛ وإذا فشل Decision Event مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Tool Event فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Trace coverage فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Agentic Observability Ledger (MAOL)

عقدة المعمارية
دفتر الرصد الوكيلي المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Trace ID: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Decision Event: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Tool Event: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Cost Event: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Redaction: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يربط فشل نشر بالـCommit والـBuild command ورسالة الخطأ وقرار Rollback، لا بعبارة عامة «لم يعمل». وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Trace ID ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Decision Event تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Tool Event تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Cost Event تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Redaction تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Decision Event وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Tool Event بتغيير Policy، ولا يستطيع Cost Event ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Agentic Observability Ledger (MAOL)

فرضية القياس
تقبل قيمة MAOL (Moussaif Agentic Observability Ledger) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Trace coverage: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Mean diagnosis time: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Log integrity: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Privacy leaks: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Replay success: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Trace coverage, Mean diagnosis time, Log integrity دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R36] [R27] [R25] [R17] [R40].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Logs نصية غامضة وغياب correlation وتسجيل الأسرار، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MAOL (Moussaif Agentic Observability Ledger) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Agentic Observability Ledger (MAOL)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Agentic Observability Ledger (MAOL) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Logs نصية غامضة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- غياب correlation: قارن بتفسير أبسط قبل إضافة مكون جديد.
- تسجيل الأسرار: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- Metrics بلا هدف: افحص الضرر غير المقصود والSide Effects.
- Telemetry غير قابلة لإعادة التشغيل: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R40] [R36] [R27] [R25] [R17] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Agentic Observability Ledger (MAOL). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Telemetry غير قابلة لإعادة التشغيل بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Dynamic Benchmark Court (MDBC)

الأطروحة المركزية
Benchmark ثابت واحد يتحول إلى هدف للحفظ؛ التقييم العام يحتاج مهامًا متغيرة، وتلوثًا مراقبًا، وقياسات مسار، واختبارات سلامة وكلفة بجانب النجاح.

يقترح فصل MDBC (Moussaif Dynamic Benchmark Court) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يبني Benchmark Court حزم مهام سرية وعامة، ويولد Variants، ويثبت Baselines، ويقيس End-to-End مع Ablations وتحليل خطأ. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Task Suite وHidden Variant بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Baseline وTrajectory Metric حين تسمح المهمة والسلطة.
- يبقى Error Taxonomy حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: لا يقاس وكيل برمجي بعدد Tokens الصحيحة فقط؛ يقاس بإصلاح Repo حقيقي، ونجاح Tests، وعدم كسر ملفات غير مرتبطة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R36] [R34] [R33] [R32] [R30] [R29].

تمييزات مفهومية تمنع التضخم

- محكمة القياس الديناميكي المساعفية ليست مرادفًا لـTask Suite؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Hidden Variant لا يثبت أثر Baseline؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Trajectory Metric في عينة يثبت صلاحية محلية، ولا يمنح Error Taxonomy حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Task Suite أو Hidden Variant معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Dynamic Benchmark Court (MDBC)

قانون الحركة
يبني Benchmark Court حزم مهام سرية وعامة، ويولد Variants، ويثبت Baselines، ويقيس End-to-End مع Ablations وتحليل خطأ.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـTask Suite مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Hidden Variant واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Baseline وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Trajectory Metric ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Goal success وRobustness ومقارنته بـBaseline.
- عرض النتيجة على Error Taxonomy، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Goal success أو Robustness بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R36] [R34] [R33] [R32] [R30] [R29].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Task Suite فالمشكلة في الإدراك أو Schema، وإذا فشل Hidden Variant مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Baseline فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Goal success فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Dynamic Benchmark Court (MDBC)

عقدة المعمارية
محكمة القياس الديناميكي المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Task Suite: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Hidden Variant: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Baseline: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Trajectory Metric: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Error Taxonomy: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: لا يقاس وكيل برمجي بعدد Tokens الصحيحة فقط، يقاس بإصلاح Repo حقيقي، ونجاح Tests، وعدم كسر ملفات غير مرتبطة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Task Suite ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Hidden Variant تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Baseline تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Trajectory Metric تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Error Taxonomy تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Hidden Variant وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Baseline بتغيير Policy، ولا يستطيع Trajectory Metric ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Dynamic Benchmark Court (MDBC)

فرضية القياس
تقبل قيمة MDBC (Moussaif Dynamic Benchmark Court) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Goal success: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Robustness: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Generalization: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Safety violations: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Reproducibility: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Goal success, Robustness, Generalization دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R36] [R34] [R33] [R32] [R30] [R29].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تلوث البيانات وMetric gaming واختبار سهل، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر، لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MDBC (Moussaif Dynamic Benchmark Court) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Dynamic Benchmark Court (MDBC)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Dynamic Benchmark Court (MDBC) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تلوث البيانات: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Metric gaming: قارن بتفسير أبسط قبل إضافة مكون جديد.
- اختبار سهل: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- تجاهل الفشل الصامت: افحص الضرر غير المقصود والSide Effects.
- مقارنة غير عادلة: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R36] [R34] [R33] [R32] [R30] [R29] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Dynamic Benchmark Court (MDBC). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر مقارنة غير عادلة بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Governed Self-Evolution (MGSE)

الأطروحة المركزية
التعلم التشغيلي الآمن لا يعدل نفسه مباشرة في الإنتاج؛ يقترح Change، ويختبره في Sandbox، ويقارن Baseline، ثم يرقى بإذن وسياسة Rollback.

يقترح فصل MGSE (Moussaif Governed Self-Evolution) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: تتحول الخبرة إلى Candidate Patch أو Skill، تمر عبر Dedup وSecurity وBenchmark، ثم Canary، ولا تصبح Canonical إلا بعد أثر مثبت. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Experience Distiller وCandidate Change بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Sandbox وBenchmark Gate حين تسمح المهمة والسلطة.
- يبقى Rollback حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: إذا نجحت طريقة بحث مرة، تنشأ Candidate Skill؛ لا تعتمد حتى تنجح في مهام متنوعة ولا تكشف بيانات أو تزيد الكلفة دون فائدة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R38] [R30] [R29] [R15] [R09].

تمييزات مفهومية تمنع التضخم

- التطور الذاتي المحكوم المساعفي ليست مرادفًا لـExperience Distiller؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Candidate Change لا يثبت أثر Sandbox؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Benchmark Gate في عينة يثبت صلاحية محلية، ولا يمنح Rollback حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Experience Distiller أو Candidate Change معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Governed Self-Evolution (MGSE)

قانون الحركة
تتحول الخبرة إلى Candidate Patch أو Skill، تمر عبر Dedup وSecurity وBenchmark، ثم Canary، ولا تصبح Canonical إلا بعد أثر مثبت.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـExperience Distiller مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Candidate Change واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Sandbox وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Benchmark Gate ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Improvement delta وRegression ومقارنته بـBaseline.
- عرض النتيجة على Rollback، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Improvement delta أو Regression بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R38] [R30] [R29] [R15] [R09].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Experience Distiller فالمشكلة في الإدراك أو Schema؛ وإذا فشل Candidate Change مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Sandbox فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Improvement delta فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Governed Self-Evolution (MGSE)

عقدة المعمارية
التطور الذاتي المحكوم المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Experience Distiller: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Candidate Change: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Sandbox: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Benchmark Gate: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Rollback: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: إذا نجحت طريقة بحث مرة، تنشأ Candidate Skill؛ لا تعتمد حتى تنجح في مهام متنوعة ولا تكشف بيانات أو تزيد الكلفة دون فائدة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Experience Distiller ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Candidate Change تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Sandbox تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Benchmark Gate تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Rollback تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Candidate Change وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Sandbox بتغيير Policy، ولا يستطيع Benchmark Gate ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Governed Self-Evolution (MGSE)

فرضية القياس
تقبل قيمة MGSE (Moussaif Governed Self-Evolution) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Improvement delta: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Regression: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Promotion precision: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Rollback time: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Drift: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Improvement delta, Regression, Promotion precision دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R30] [R29] [R15] [R09] [R38].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تثبيت خطأ وSelf-modification بلا سجل وBenchmark overfit، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MGSE (Moussaif Governed Self-Evolution) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Governed Self-Evolution (MGSE)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Governed Self-Evolution (MGSE) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تثبيت خطأ: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Self-modification بلا سجل: قارن بتفسير أبسط قبل إضافة مكون جديد.
- Benchmark overfit: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- Canary غير ممثل: افحص الضرر غير المقصود والSide Effects.
- تراكم Skills: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R38] [R30] [R29] [R15] [R09] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Governed Self-Evolution (MGSE). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر تراكم Skills بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Skill-Genesis and Deduplication (MSGD)

الأطروحة المركزية
إنشاء Skills يوميًا لا يساوي تطورًا؛ يجب استخراج نمط متكرر، وتحديد Trigger مميز، وإثبات منفعة هامشية، ودمج النسخ المتداخلة.

يقترح فصل Moussaif Skill-Genesis and Deduplication (MSGD) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يستخرج Skill Miner مسارات ناجحة، ويجرد التفاصيل المحلية، ويولد Spec، ويقارن Embedding وStructure، ثم يدمج أو ينسخ بإصدار جديد. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Trajectory Miner وPattern Abstractor بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Skill Spec وSimilarity Check حين تسمح المهمة والسلطة.
- يبقى Versioner حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: تدمج مهارات «تغيير URL» و«ترحيل Domain» إذا تشاركت الشروط والإجراء والاختبارات، مع Parameters بدل نسختين. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R38] [R12] [R09] [R07].

تمييزات مفهومية تمنع التضخم

- توليد المهارات وإزالة التكرار المساعفي ليست مرادفًا لـTrajectory Miner؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Pattern Abstractor لا يثبت أثر Skill Spec؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Similarity Check في عينة يثبت صلاحية محلية، ولا يمنح Versioner حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Trajectory Miner أو Pattern Abstractor معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Skill-Genesis and Deduplication (MSGD)

قانون الحركة

يستخرج Skill Miner مسارات ناجحة، ويجرد التفاصيل المحلية، ويولد Spec، ويقارن Embedding وStructure، ثم يدمج أو ينسخ بإصدار جديد.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B_t>، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـTrajectory Miner مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Pattern Abstractor واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Skill Spec وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Similarity Check ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Dedup precision وReuse rate ومقارنته بـBaseline.
- عرض النتيجة على Versioner، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Dedup precision أو Reuse rate بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R38] [R12] [R09] [R07].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Trajectory Miner فالمشكلة في الإدراك أو Schema، وإذا فشل Pattern Abstractor مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Skill Spec فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Dedup precision فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Skill-Genesis and Deduplication (MSGD)

عقدة المعمارية
توليد المهارات وإزالة التكرار المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Trajectory Miner: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Pattern Abstractor: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Skill Spec: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Similarity Check: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Versioner: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: تدمج مهارات «تغيير URL» و«ترحيل Domain» إذا تشاركت الشروط والإجراء والاختبارات، مع Parameters بدل نسختين. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Trajectory Miner ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Pattern Abstractor تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Skill Spec تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Similarity Check تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Versioner تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Pattern Abstractor وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Skill Spec بتغيير Policy، ولا يستطيع Similarity Check ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Skill-Genesis and Deduplication (MSGD)

فرضية القياس
تقبل قيمة MSGD (Moussaif Skill-Genesis and Deduplication) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Dedup precision: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Reuse rate: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Skill success: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Registry growth: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Maintenance cost: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Dedup precision, Reuse rate, Skill success دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R38] [R12] [R09] [R07].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Skill لكل حادثة وأسماء مختلفة للمسار نفسه وTrigger متضارب، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء وبحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MSGD (Moussaif Skill-Genesis and Deduplication) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Skill-Genesis and Deduplication (MSGD)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Skill-Genesis and Deduplication (MSGD) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Skill لكل حادثة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- أسماء مختلفة للمسار نفسه: قارن بتفسير أبسط قبل إضافة مكون جديد.
- Trigger متضارب: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- اعتماد مخفي: افحص الضرر غير المقصود والSide Effects.
- نمو غير محدود: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R38] [R12] [R09] [R07] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Skill-Genesis and Deduplication (MSGD). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر نمو غير محدود بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Human-Agency Symbiosis (MHAS)

الأطروحة المركزية
هدف النظام ليس استبدال القرار البشري في كل موضع، بل رفع قدرة الإنسان على الفهم والاختيار والتنفيذ مع بقاء حق الاعتراض والانسحاب والمعرفة.

يقترح فصل Moussaif Human-Agency Symbiosis (MHAS) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يصنف Autonomy Ladder الأفعال، ويقدم Preview وتفسيرًا وبدائل، ويحجز القرارات القيمية أو عالية الأثر للإنسان، ويتعلم من Corrections المصرح بها. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Autonomy Level وPreview بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Explanation وChoice Set حين تسمح المهمة والسلطة.
- يبقى Correction Loop حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: في بحث نظري، يقترح النظام بنية وأدلة واعتراضات، لكن ينسب النظرية بوضوح ويترك للمؤلف حكم النشر والادعاء النهائي. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R40] [R31] [R27] [R24] [R22].

تمييزات مفهومية تمنع التضخم

- تكافل الفاعلية البشرية المساعفي ليست مرادفًا لـAutonomy Level؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Preview لا يثبت أثر Explanation؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Choice Set في عينة يثبت صلاحية محلية، ولا يمنح Correction Loop حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Autonomy Level أو Preview معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Human-Agency Symbiosis (MHAS)

قانون الحركة
يصنف Autonomy Ladder الأفعال، ويقدم Preview وتفسيرًا وبدائل، ويحجز القرارات القيمية أو عالية الأثر للإنسان، ويتعلم من Corrections المصرح بها.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـAutonomy Level مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Preview واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Explanation وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Choice Set ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Human control وDecision quality ومقارنته بـBaseline.
- عرض النتيجة على Correction Loop، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Human control أو Decision quality بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R31] [R27] [R24] [R22].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Autonomy Level فالمشكلة في الإدراك أو Schema؛ وإذا فشل Preview مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Explanation فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Human control فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Human-Agency Symbiosis (MHAS)

عقدة المعمارية
تكافل الفاعلية البشرية المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Autonomy Level: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Preview: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Explanation: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Choice Set: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Correction Loop: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: في بحث نظري، يقترح النظام بنية وأدلة واعتراضات، لكن ينسب النظرية بوضوح ويترك للمؤلف حكم النشر والادعاء النهائي. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Autonomy Level ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Preview تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Explanation تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Choice Set تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Correction Loop تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Preview وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Explanation بتغيير Policy، ولا يستطيع Choice Set ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Human-Agency Symbiosis (MHAS)

فرضية القياس
تقبل قيمة MHAS (Moussaif Human-Agency Symbiosis) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Human control: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Decision quality: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Override latency: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Cognitive load: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Consent integrity: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Human control, Decision quality, Override latency دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R31] [R27] [R24] [R22] [R40].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Automation bias وواجهة تخفي الأثر وموافقة شكلية، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MHAS (Moussaif Human-Agency Symbiosis) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Human-Agency Symbiosis (MHAS)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Human-Agency Symbiosis (MHAS) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Automation bias: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- واجهة تخفي الأثر: قارن بتفسير أبسط قبل إضافة مكون جديد.
- موافقة شكلية: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- إزعاج بتأكيدات تافهة: افحص الضرر غير المقصود والSide Effects.
- سلب المهارة البشرية: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R40] [R31] [R27] [R24] [R22] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Human-Agency Symbiosis (MHAS). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر سلب المهارة البشرية بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Evidence-to-Value Engine (MEVE)

الأطروحة المركزية
القيمة الاقتصادية المستدامة تبدأ من حاجة مرصودة وآلية منفعة وقياس أثر، لا من وعد «مليون سريع» أو تضخيم قدرات غير مثبتة.

يقترح فصل Moussaif Evidence-to-Value Engine (MEVE) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يربط Value Engine Need Evidence بفرضية Offer، وتجربة صغيرة، وOutcome Metric، وتكلفة تسليم، ثم يوسع فقط عند Retention أو أثر موثق. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Need Signal وValue Hypothesis بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Offer Mechanism وPilot حين تسمح المهمة والسلطة.
- يبقى Trust Proof حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: قبل حملة B2B، يختبر النظام عرضًا على عينة صغيرة بإذن وقيمة واضحة، ويقيس ردودًا ذات معنى بدل عدد الرسائل المرسلة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R36] [R31] [R27] [R18].

تمييزات مفهومية تمنع التضخم

- محرك الدليل إلى القيمة المساعفي ليست مرادفًا لـNeed Signal؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Value Hypothesis لا يثبت أثر Offer Mechanism؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Pilot في عينة يثبت صلاحية محلية، ولا يمنح Trust Proof حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Need Signal أو Value Hypothesis معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Evidence-to-Value Engine (MEVE)

قانون الحركة
يربط Value Engine Need Evidence بفرضية Offer، وتجربة صغيرة، وOutcome Metric، وتكلفة تسليم، ثم يوسع فقط عند Retention أو أثر موثق.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $\Omega_t = \langle G,W,M,K,S,T,P,E,B \rangle$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـNeed Signal مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Value Hypothesis واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Offer Mechanism وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Pilot ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Customer outcome وConversion quality ومقارنته بـBaseline.
- عرض النتيجة على Trust Proof، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Customer outcome أو Conversion quality بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R36] [R31] [R27] [R18].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Need Signal فالمشكلة في الإدراك أو Schema، وإذا فشل Value Hypothesis مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Offer Mechanism فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Customer outcome فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Evidence-to-Value Engine (MEVE)

عقدة المعمارية
محرك الدليل إلى القيمة المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Need Signal: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Value Hypothesis: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Offer Mechanism: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Pilot: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Trust Proof: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: قبل حملة B2B، يختبر النظام عرضًا على عينة صغيرة بإذن وقيمة واضحة، ويقيس ردودًا ذات معنى بدل عدد الرسائل المرسلة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Need Signal ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Value Hypothesis تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Offer Mechanism تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Pilot تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Trust Proof تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Value Hypothesis وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Offer Mechanism بتغيير Policy، ولا يستطيع Pilot ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Evidence-to-Value Engine (MEVE)

فرضية القياس
تقبل قيمة MEVE (Moussaif Evidence-to-Value Engine) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Customer outcome: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Conversion quality: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Retention: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Delivery cost: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Evidence strength: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Customer outcome, Conversion quality, Retention دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R36] [R31] [R27] [R18].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Vanity metrics وSpam outreach وحاجة مخترعة، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر؛ فإذا اختفى أثر MEVE (Moussaif Evidence-to-Value Engine) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Evidence-to-Value Engine (MEVE)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Evidence-to-Value Engine (MEVE) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Vanity metrics: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Spam outreach: قارن بتفسير أبسط قبل إضافة مكون جديد.
- حاجة مخترعة: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- وعود AGI: افحص الضرر غير المقصود والـSide Effects.
- توسع قبل Product-Market Evidence: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R36] [R31] [R27] [R18] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Evidence-to-Value Engine (MEVE). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر توسع قبل Product-Market Evidence بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Propagative Overlay Deployment (MPOD)

الأطروحة المركزية
ينبغي نشر AGI-OS كـOverlay قابل للإزالة فوق runtimes مختلفة، عبر عقود وملفات ومهارات، لا بادعاء تغيير الأوزان الخفية أو كسر النواة الأصلية.

يقترح فصل Moussaif Propagative Overlay Deployment (MPOD) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يبني Overlay Compiler Manifest وAdapters وSkill Registry وPolicy، يختبرها محليًا، ثم ينشرها تدريجيًا مع Version Pin وRollback. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Overlay Manifest وRuntime Adapter بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Policy Pack وSkill Bundle حين تسمح المهمة والسلطة.
- يبقى Rollback حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يحمل Overlay نفسه إلى Hermes وCodex وClaude وCursor وOllama مع Adapter لكل أداة، ويبقى المعايير والSkills مستقلة عن المزود. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R35] [R25] [R17] [R12].

تمييزات مفهومية تمنع التضخم

- النشر التراكمي الانتشاري المساعفي ليست مرادفًا لـOverlay Manifest؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Runtime Adapter لا يثبت أثر Policy Pack؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Skill Bundle في عينة يثبت صلاحية محلية، ولا يمنح Rollback حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Overlay Manifest أو Runtime Adapter معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Propagative Overlay Deployment (MPOD)

قانون الحركة
يبني Overlay Compiler Manifest وAdapters وSkill Registry وPolicy، يختبرها محليًا، ثم ينشرها تدريجيًا مع Version Pin وRollback.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه $S_t = <G,W,M,K,S,T,P,E,B>$، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـOverlay Manifest مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Runtime Adapter واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Policy Pack وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Skill Bundle ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Portability وInstall success ومقارنته بـBaseline.
- عرض النتيجة على Rollback، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Portability أو Install success بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R35] [R25] [R17] [R12].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Overlay Manifest فالمشكلة في الإدراك أو Schema، وإذا فشل Runtime Adapter مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Policy Pack فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل Portability فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Propagative Overlay Deployment (MPOD)

عقدة المعمارية
النشر التراكبي الانتشاري المساعفي تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Overlay Manifest: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Runtime Adapter: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Policy Pack: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Skill Bundle: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Rollback: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يحمل Overlay نفسه إلى Hermes وCodex وClaude وCursor وOllama مع Adapter لكل أداة، ويبقى المعايير والـSkills مستقلة عن المزود. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Overlay Manifest ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Runtime Adapter تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Policy Pack تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Skill Bundle تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Rollback تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Runtime Adapter وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Policy Pack بتغيير Policy، ولا يستطيع Skill Bundle ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Propagative Overlay Deployment (MPOD)

فرضية القياس
تقبل قيمة Moussaif Propagative Overlay Deployment (MPOD) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Portability: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Install success: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Runtime parity: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Rollback: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Upgrade breakage: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Portability, Install success, Runtime parity دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R35] [R25] [R17] [R12].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل تعديل Core غير موثق وافتراض أدوات غير موجودة ومسارات ثابتة، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للـRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Propagative Overlay Deployment (MPOD) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Propagative Overlay Deployment (MPOD)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Propagative Overlay Deployment (MPOD) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- تعديل Core غير موثق: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- افتراض أدوات غير موجودة: قارن بتفسير أبسط قبل إضافة مكون جديد.
- مسارات ثابتة: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- أسرار داخل Bundle: افحص الضرر غير المقصود والSide Effects.
- انتشار بلا Version: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R35] [R25] [R17] [R12] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Propagative Overlay Deployment (MPOD). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر انتشار بلا Version بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Resilient-Recovery Mesh (MRRM)

الأطروحة المركزية
النظام الجيد يتوقع الفشل الجزئي ويحتوي أثره؛ الاستمرارية تأتي من Checkpoints وFallback وCircuit Breakers ونسخ قابلة للعودة، لا من إنكار الخطأ.

يقترح فصل Moussaif Resilient-Recovery Mesh (MRRM) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يراقب Recovery Mesh إشارات الخطأ، ويعزل المكون، ويحفظ الحالة، ويختار Degraded Mode، ويعيد التشغيل من Checkpoint بعد تحقق السبب. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Health Signal وCircuit Breaker بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Checkpoint وDegraded Mode حين تسمح المهمة والسلطة.
- يبقى Postmortem حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: إذا فشل مزود Model، يحفظ الوكيل خطة المهمة والمخرجات المثبتة، وينقل الخطوات المتوافقة فقط، ولا يعيد الأفعال الخارجية المنجزة. يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R17] [R25] [R27] [R37].

تمييزات مفهومية تمنع التضخم

- شبكة التعافي المرن المساعفية ليست مرادفًا لـHealth Signal؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Circuit Breaker لا يثبت أثر Checkpoint؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Degraded Mode في عينة يثبت صلاحية محلية، ولا يمنح Postmortem حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Health Signal أو Circuit Breaker معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Resilient-Recovery Mesh (MRRM)

قانون الحركة
يراقب Recovery Mesh إشارات الخطأ، ويعزل المكون، ويحفظ الحالة، ويختار Degraded Mode، ويعيد التشغيل من Checkpoint بعد تحقق السبب.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـHealth Signal مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Circuit Breaker واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Checkpoint وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Degraded Mode ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري MTTR وData loss ومقارنته بـBaseline.
- عرض النتيجة على Postmortem، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في MTTR أو Data loss بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R17] [R25] [R27] [R37].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Health Signal فالمشكلة في الإدراك أو Schema، وإذا فشل Circuit Breaker مع مدخل صحيح فالمشكلة في القرار، وإذا لم ينفذ Checkpoint فالمشكلة في التوافق أو السلطة، وإذا نجح التنفيذ وفشل MTTR فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Resilient-Recovery Mesh (MRRM)

عقدة المعمارية
شبكة التعافي المرن المساعفية تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Health Signal: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Circuit Breaker: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Checkpoint: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Degraded Mode: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Postmortem: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: إذا فشل مزود Model، يحفظ الوكيل خطة المهمة والمخرجات المثبتة، وينقل الخطوات المتوافقة فقط، ولا يعيد الأفعال الخارجية المنجزة. وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Health Signal ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Circuit Breaker تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Checkpoint تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Degraded Mode تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Postmortem تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Circuit Breaker وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Checkpoint بتغيير Policy، ولا يستطيع Degraded Mode ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Resilient-Recovery Mesh (MRRM)

فرضية القياس
تقبل قيمة Moussaif Resilient-Recovery Mesh (MRRM) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- MTTR: يعرف الحدث والـDenominator ونافذة القياس قبل التجربة.
- Data loss: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Recovery success: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Cascading failure: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Repeat incident: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من MTTR, Data loss, Recovery success دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R37] [R27] [R25] [R17].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل Fallback صامت رديء وCheckpoint فاسد واستعادة تكرر الأثر، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للـRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر Moussaif Resilient-Recovery Mesh (MRRM) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Resilient-Recovery Mesh (MRRM)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Resilient-Recovery Mesh (MRRM) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- Fallback صامت رديء: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- Checkpoint فاسد: قارن بتفسير أبسط قبل إضافة مكون جديد.
- استعادة تكرر الأثر: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- اعتماد دائري: افحص الضرر غير المقصود والSide Effects.
- Postmortem بلا تغيير: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R37] [R27] [R25] [R17] فتثبت أن أجزاء من فضاء المشكلة مدروسة، ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Resilient-Recovery Mesh (MRRM). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر Postmortem بلا تغيير بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

التأسيس والحدود

Moussaif Fattah Open-Intelligence Program (MFOIP)

الأطروحة المركزية
الذكاء الفتّاح برنامج بحثي لتوسيع قدرة النظام على توليد الأسئلة والآليات والأدوات والقيمة تحت قيود الدليل والسلامة؛ وهو اتجاه بحث لا ادعاء ببلوغ ما بعد AGI.

يقترح فصل MFOIP (Moussaif Fattah Open-Intelligence Program) وحدة مفاهيمية داخل نواة Moussaif Agentic AGI OS. المقصود بـAGI-level هنا مستوى طموح هندسي يقاس عبر طيف مهام وانتقال بين بيئات، لا شهادة بأن النظام الحالي حقق ذكاءً عامًا. ولهذا تفصل النظرية بين اسم البناء، ووظيفته المقترحة، والدليل المتاح على مكونات مشابهة، والاختبار المطلوب لإثبات المنظومة. حالته المعرفية الحالية: إطار مقترح يحتاج إلى تحقق تجريبي مستقل.

التعريف التشغيلي: يدير البرنامج Portfolio من فرضيات قابلة للتكذيب، ويقارن Architectures، وينشر Benchmarks ونتائج سلبية، ويربط كل ترقية بمعيار مستقل. هذا التعريف لا يكتفي بوصف النتيجة، بل يذكر التحول الذي يجب أن يحدث بين المدخل والأثر. ويصبح المفهوم قابلاً للاستعمال حين يمكن تمثيل حالته، واستدعاؤه بشروط معلنة، وقياس الفرق الذي أحدثه، وعكسه أو تعطيله إذا أنتج ضررًا. أما إذا بقي الاسم منفصلاً عن State وInterface وMetric فإنه يظل استعارة بحثية لا مكونًا هندسيًا.

حدود المفهوم

- يدخل في النطاق: Research Portfolio وOpen Question بوصفهما عقدتين ضروريتين.
- يمتد التنفيذ إلى: Prototype وFalsification Plan حين تسمح المهمة والسلطة.
- يبقى Publication Ledger حارسًا أو نقطة تحقق، ولا يتحول إلى زينة اصطلاحية.
- لا يدخل في النطاق: ادعاء وعي آلي، أو استقلال مطلق، أو تجاوز إذن الإنسان، أو تغيير أوزان نموذج مغلق.
- نقل النجاح من مهمة إلى مجال جديد يحتاج اختبار Transfer؛ التشابه اللغوي وحده لا يكفي.

المثال الحاكم لهذا الفصل هو الآتي: يبدأ البرنامج بسؤال محدود: هل يرفع Q-MCD نجاح التخطيط متعدد الأدوات مقارنة بـReAct baseline، وبأي كلفة وأي أنواع خطأ؟ يختبر المثال هل يغير المفهوم مسار القرار فعلاً، وهل يترك أثرًا يمكن فحصه، أم أنه يعيد تسمية ممارسة معروفة دون قيمة هامشية. الإسناد الخارجي هنا يقدم أدلة مكونات ومقارنات، لا يثبت الاسم أو النظام المساعفي ككل: [R30] [R29] [R18] [R10] [R05] [R40] [R39] [R36].

تمييزات مفهومية تمنع التضخم

- برنامج مساعف الفتّاح للذكاء المفتوح ليست مرادفًا لـResearch Portfolio؛ الأول عقد نظامي، والثاني مكون أو حالة داخله.
- وجود Open Question لا يثبت أثر Prototype؛ يلزم Trace يربط الاستدعاء بالنتيجة.
- نجاح Falsification Plan في عينة يثبت صلاحية محلية، ولا يمنح Publication Ledger حكمًا عامًا خارج التوزيع.
- القدرة Capability تجيب: هل يمكن؟ والاعتمادية Reliability تجيب: كم مرة وبأي شروط؟ والسلطة Authorization تجيب: هل يجوز الآن؟
- الوكالة Agency تعني استمرار الهدف عبر الأفعال والتغذية الراجعة؛ أما الاستقلال Autonomy فهو درجة تشغيل قابلة للخفض ولا يساوي الذكاء.

يرتبط المفهوم بالنظرية الأم عبر ثلاث واجهات. واجهة Upstream تستقبل Goal Contract وWorld-State Slice وAuthority Token؛ وواجهة Downstream تسلم Decision Packet أو Action Proposal؛ وواجهة Lateral تتبادل Evidence Pointers مع الذاكرة والمحاكم. هذا التقسيم يمنع المكون من إعادة تفسير الهدف أو توسيع الصلاحية سرًا. كما يسمح بإزالة النظرية الفرعية في Ablation من دون إعادة بناء النظام كله، وهو شرط ضروري لمعرفة هل الاسم يضيف وظيفة أم يحجب اقترانًا داخل Prompt ضخم.

من منظور Novelty، قد تكون عناصر مثل Research Portfolio أو Open Question معروفة بأسماء أخرى؛ الإضافة المدعاة هنا هي صياغة العلاقة والبوابات والقياس ضمن Moussaif AGI-OS. ولهذا يتطلب النشر الأكاديمي Literature Review أوسع من قائمة هذا الكتاب، ومقارنة مباشرة بأقرب Baseline، وتحليلاً لما إذا كانت المصطلحات الجديدة تقلل الغموض أو تزيده. إذا لم يظهر فرق تنبؤي أو تنفيذي، تدمج التسمية في المفهوم الأقرب بدل حمايتها بالبلاغة.

الآلية السببية

Moussaif Fattah Open-Intelligence Program (MFOIP)

قانون الحركة
يدير البرنامج Portfolio من فرضيات قابلة للتكذيب، ويقارن Architectures، وينشر Benchmarks ونتائج سلبية، ويربط كل ترقية بمعيار مستقل.

تعمل الآلية كتحويل حالة لا كسلسلة أفكار حرة. نرمز لحالة النظام عند الزمن t بالمتجه <G,W,M,K,S,T,P,E,B> = Ω_t، حيث G للغايات، W للعالم، M للذاكرة، K للمعرفة، S للمهارات، T للأدوات، P للسياسات، E للأدلة، وB للميزانيات. يقبل الانتقال فقط إذا غيّر عنصرًا معروفًا وترك Observation يمكن ربطه بالفعل.

Pipeline السببي

- تكوين Snapshot للحالة المرتبطة بـResearch Portfolio مع مصدر وزمن وعدم يقين.
- تمرير الحالة إلى Open Question واستخراج الفرق الذي يغير القرار بدل إعادة الصياغة.
- فحص جاهزية Prototype وشروطه وسلطته وكلفته قبل أي استدعاء.
- تشغيل Falsification Plan ضمن Budget وCheckpoint وإشارة فشل يمكن رصدها.
- قياس الأثر الأولي بمؤشري Hypothesis yield وReplication ومقارنته بـBaseline.
- عرض النتيجة على Publication Ledger، ثم تثبيت التغيير أو التراجع أو التصعيد البشري.

يمنع هذا الـPipeline ثلاثة اختصارات خطرة: القفز من الهدف إلى أداة بلا World State، والقفز من Output تقني إلى نجاح مهمة بلا Acceptance Test، والقفز من محاولة واحدة إلى ذاكرة دائمة بلا Promotion Gate. عند فشل خطوة، لا تعاد الحلقة آليًا؛ يصنف الفشل إلى نقص دليل، أو عدم توافق، أو حدود سلطة، أو عطب أداة، أو خطأ فرضية، لأن كل فئة تستدعي استجابة مختلفة.

```
ΔΩ_t = Observe(Action(Plan(Q(State_t)))) - State_t
```

قابلية التكذيب الآلية: إذا لم يظهر فرق متكرر في Hypothesis yield أو Replication بعد التحكم في Baseline وكلفة التنسيق، فإن إضافة هذا المكون لا تستحق التعقيد في تلك الفئة من المهام. الأدلة المقارنة ذات الصلة: [R40] [R39] [R36] [R30] [R29] [R18] [R10] [R05].

ثوابت Invariants أثناء التشغيل

- Goal invariant: لا تغير أي خطوة تعريف النجاح من غير إصدار جديد لعقد المهمة.
- Authority invariant: لا يولد Output من نموذج صلاحية لم تكن موجودة في المدخل.
- Evidence invariant: كل انتقال ادعائي يحمل Source أو يوسم بأنه Hypothesis.
- State invariant: لا تستبدل حقيقة مرصودة بتخمين أعلى طلاقة من دون حفظ التعارض.
- Recovery invariant: كل فعل غير قابل للعكس يحتاج Confirmation أو بديل احتواء مناسبًا للخطر.
- Learning invariant: لا تتحول النتيجة إلى Skill أو Memory دائمة قبل Verification مستقل.

تتغير ديناميكيات الآلية حسب نوع الخطأ. إذا انهار Research Portfolio فالمشكلة في الإدراك أو Schema؛ وإذا فشل Open Question مع مدخل صحيح فالمشكلة في القرار؛ وإذا لم ينفذ Prototype فالمشكلة في التوافق أو السلطة؛ وإذا نجح التنفيذ وفشل Hypothesis yield فالمشكلة في الفرضية نفسها. هذا التصنيف يمنع Retry أعمى لنموذج أغلى، ويجعل Recovery action مطابقًا للسبب: إعادة استرجاع، أو تصحيح Argument، أو تضييق نطاق، أو Rollback، أو إحالة بشرية.

لإثبات السببية، لا يكفي أن تتبع النتيجة استدعاء المكون. يحتاج الاختبار إلى Intervention: تعطيل المكون مع ثبات باقي الشروط، أو استبداله ببديل أبسط، أو تبديل قراره مع تثبيت الحالة، ثم قياس Observed Delta. ويجب تسجيل Mediators؛ فقد يبدو أن Router حسن الأداء بينما جاء الفرق من Context أطول أو Tool إضافية. لهذا يحتفظ Trace بنسخة Prompt/Policy، وModel، وبيانات الاسترجاع، والميزانية، والنتيجة، كي تصبح المقارنة قابلة للتفسير.

المعمارية وعقود البيانات

Moussaif Fattah Open-Intelligence Program (MFOIP)

عقدة المعمارية
برنامج مساعف الفتّاح للذكاء المفتوح تعمل كخدمة ذات عقد إدخال/إخراج، لا كفقرة مخفية داخل System Prompt.

تعتمد المعمارية مبدأ الفصل بين Control Plane الذي يحمل السياسات والعقود، وData Plane الذي يحمل محتوى المستخدم والمصادر، وExecution Plane الذي يستدعي الأدوات، وEvidence Plane الذي يثبت ما حدث. هذا الفصل يجعل المكون قابلاً للنقل بين runtimes ويقلل خطر أن يتحول نص مسترجع إلى أمر تنفيذي.

الوحدات الداخلية

- Research Portfolio: يستقبل جزءًا typed من الحالة ويعلن الحقول المفقودة.
- Open Question: ينتج قرارًا وسيطًا مع سبب ومؤشر ثقة منفصل.
- Prototype: يطبق شروط الجاهزية والتوافق وحدود السلطة.
- Falsification Plan: ينفذ أو يحاكي أصغر تغير يختبر الفرضية.
- Publication Ledger: يقرأ Evidence Packet ويصدر Pass أو Fail أو Escalate.
- Audit Adapter: يسجل النسخة والكلفة والزمن والمصدر دون حفظ أسرار خام.

عقد البيانات الأدنى

Input	task_id, state_ref, constraints, authority, evidence_refs
Decision	candidate, score_vector, alternatives, uncertainty
Action	tool, arguments_hash, expected_effect, rollback_ref
Outcome	observed_delta, metrics, verdict, next_state

يجب أن تكون الواجهة Adapter-neutral: يمكن تنفيذ القرار عبر Function Call أو MCP-compatible tool أو Shell wrapper أو API، لكن Schema وPolicy وVerification تبقى ثابتة. في المثال التطبيقي: يبدأ البرنامج بسؤال محدود: هل يرفع Q-MCD نجاح التخطيط متعدد الأدوات مقارنة بـReAct baseline، وبأي كلفة وأي أنواع خطأ؟ وعليه لا يوثق النجاح باسم المزود، بل بالـContract والـObserved Delta والاختبار.

عقود الواجهات والاعتماديات

- واجهة Research Portfolio ترفض الحقول المجهولة وتعيد Missing-Field Report بدل ملئها بالتخمين.
- واجهة Open Question تعيد Score Vector وAlternatives، لا قرارًا بلا تفسير قابل للفحص.
- واجهة Prototype تفصل Validation عن Execution وتحتفظ بمعرف Idempotency.
- واجهة Falsification Plan تسجل Expected Effect قبل الأثر وObserved Effect بعده.
- واجهة Publication Ledger تستخدم Artifact منقحًا وتطبق Rubric مثبتًا قبل رؤية النتيجة.
- واجهة Memory لا تستقبل Raw trace كاملاً؛ تستقبل Episode منقحًا مع Retention وSensitivity.

تدار الحالة وفق Event Sourcing مبسط: لا يحذف تحديثٌ القيمة السابقة، بل يضيف حدثًا يحمل actor وtimestamp وsource وconfidence وsupersedes. يعاد بناء Current View من الأحداث المقبولة، وتبقى الأحداث المتعارضة متاحة للمحكمة. هذه البنية تزيد حجم السجل قليلاً لكنها تجعل Rollback والتدقيق وإعادة التشغيل ممكنة. ولخفض الكلفة تخزن Payloads الكبيرة خارج Ledger مع Content Hash وPointer وصلاحية وصول، بينما يبقى سجل القرار صغيرًا وقابلاً للنقل.

تحد المعمارية من Blast Radius: لا يمنح Open Question وصولاً مباشرًا إلى كل الأدوات، ولا تسمح نتيجة Prototype بتغيير Policy، ولا يستطيع Falsification Plan ترقية نفسه إلى Canonical. تطبق الشبكة Timeouts وCircuit Breaker وConcurrency limits. وتختبر Contracts باختبارات خصائص: حقول إلزامية، قيم حدودية، Unicode، ملفات كبيرة، مصادر غير موثوقة، وفشل جزئي. بذلك تصبح قابلية النقل نتيجة Schemas واختبارات، لا تشابه أسماء بين منصات.

القياس والتنفيذ

Moussaif Fattah Open-Intelligence Program (MFOIP)

فرضية القياس
تقبل قيمة MFOIP (Moussaif Fattah Open-Intelligence Program) فقط إذا حسنت منفعة معدلة بالمخاطر والكلفة مقارنة بخط أساس معلن.

يقاس المكون على مستويين: Component Evaluation لعزل وظيفته، وEnd-to-End Evaluation لقياس أثره داخل النظام. قد يتحسن المكون محليًا بينما يزداد زمن المهمة أو خطؤها بسبب التنسيق؛ لذلك لا تجمع الدرجات في رقم واحد قبل عرض Vector كامل للنجاح والسلامة والكلفة والمعايرة وقابلية التكرار.

المقاييس الأساسية

- Hypothesis yield: يعرف الحدث والDenominator ونافذة القياس قبل التجربة.
- Replication: يقارن Baseline ونسخة المكون مع عينة مهام متوازنة.
- Benchmark gain: يفحص حساسية النتيجة لتغير الصياغة والنموذج والأداة.
- Safety margin: يسجل الخطأ الصامت لا الحالات التي أعادت Exception فقط.
- Negative-result rate: يحول التحسن إلى قيمة قابلة للمراجعة لا Vanity metric.

بروتوكول التجربة: عينات تدريبية منفصلة عن الاختبار، Baseline بسيط، نسخة كاملة، وAblation تحذف المكون، مع ثلاث Seeds أو إعادة تشغيل مستقلة عندما تسمح الكلفة. تسجل Tokens وLatency وTool Calls وHuman Interventions، وتراجع الحالات لا المتوسط فقط. يعلن التلوث المحتمل، وتوضع نتائج سلبية في السجل بدل إخفائها.

```
Utility* = GoalSuccess × EvidenceQuality × Safety / (1 + Cost + Latency + CoordinationLoss)
```

معيار قبول أولي مقترح: تحسن موثوق في اثنين من Hypothesis yield, Replication, Benchmark gain دون تدهور دال في السلامة أو الكلفة الكلية، ثم إعادة مستقلة على مجال آخر. لا يحول النجاح المحدود إلى ادعاء AGI. مرجعيات المقارنة: [R30] [R29] [R18] [R10] [R05] [R36] [R39] [R40].

حزمة البيانات اللازمة لإعادة الإنتاج

- Task fixture منقح يثبت المدخل والحالة المتاحة وما يجب حجبه.
- Frozen contract يحدد النجاح والفشل والسلطة والميزانية قبل التشغيل.
- Versioned runtime record يضم Model وAdapter وPolicy وSkill وTool versions.
- Trajectory events مع timestamps وcosts وevidence pointers وobserved deltas.
- Human review rubric مع سبب الحكم ومستوى الاتفاق بين المقيمين.
- Failure taxonomy وnegative cases وملف README يشرح إعادة التجربة.

يحلل الخطأ لاجتياز المتوسط. تبنى Confusion Matrix لأنواع مثل اسم كبير بلا تجربة وخلط الفلسفة بالمنتج وإخفاء النتائج السلبية، ويقاس أيها يتحسن وأيها يتدهور. وقد يكون التحسن العام نتيجة مهام سهلة كثيرة بينما تفشل الحالات عالية الأثر؛ لذلك توزن النتائج بالخطر وتعرض Worst-case وTail latency ومعدل الخرق. عند وجود حكم بشري، تحفظ عينات عمياء ويحسب Agreement، ويعلن أي تعديل للRubric بعد بدء القياس.

يختبر Transfer على ثلاثة محاور: صياغة جديدة للمهمة نفسها، وبيئة أو أداة جديدة لنفس الهدف، ومجال جديد يحتفظ بالبنية السببية. إذا احتاج المكون إلى إعادة ضبط كاملة في كل انتقال، فهو Workflow محلي لا قدرة عامة. كما يقاس Degradation عند خفض Context أو استخدام Model أصغر، فإذا اختفى أثر MFOIP (Moussaif Fattah Open-Intelligence Program) تمامًا، تسجل تبعيته ولا ينسب التحسن إلى النظرية وحدها. معيار النشر هو جدول نتائج كامل مع ثقة وحدود، لا لقطة نجاح مختارة.

محكمة الهدم وبرنامج البحث

Moussaif Fattah Open-Intelligence Program (MFOIP)

سؤال محكمة الهدم
ما المشاهد الذي لو تحقق سيجعل Moussaif Fattah Open-Intelligence Program (MFOIP) غير ضرورية، أو ضارة، أو مجرد إعادة تسمية؟

تبدأ المحكمة بأقوى صيغة قابلة للاختبار، ثم تفصل نقد الاسم عن نقد الآلية. لا يعد غموض كلمة سببًا كافيًا لرفض المكوّن إذا أمكن ضبطها، كما لا يعد الاسم الدقيق دليلاً على أن المكوّن يعمل. الحكم الممكن هو: صالح كتجريد، صالح هندسيًا في نطاق، غير متميز عن Baseline، غير آمن، أو غير محسوم لقصور الدليل.

مسارات الفشل الواجب مهاجمتها

- اسم كبير بلا تجربة: ابن حالة اختبار تجعل هذا الفشل ظاهرًا وقابلاً للعد.
- خلط الفلسفة بالمنتج: قارن بتفسير أبسط قبل إضافة مكون جديد.
- إخفاء النتائج السلبية: اختبر أثره في بيئة مختلفة أو مع Model أضعف.
- Benchmark خاص: افحص الضرر غير المقصود والSide Effects.
- سباق قدرات بلا حوكمة: ضع Stop Condition تمنع استمرار الفشل بصمت.

اختبار المثال المضاد يجمع أربع حزم: مهمة لا تحتاج المكون، ومهمة تحتاجه ظاهريًا لكنه يضلل، ومهمة خصمية تستهدف حدوده، ومهمة نقل خارج المجال. إذا فشل المكون في الحزمة الأولى بسبب كلفة زائدة، أو في الثانية بسبب ثقة زائفة، أو في الثالثة بسبب أثر غير مأذون، تسجل النتيجة كقيد تصميم لا كمجرد حالة شاذة.

بوابة القبول الدنيا

- Claim محدد ودرجة دليل معلنة.
- Baseline وAblation ومقياس سابق للنتيجة.
- Trace قابل لإعادة التشغيل دون أسرار.
- تحليل فشل وحالة Rollback أو تعطيل.
- حدود نقل تمنع تعميم النتيجة إلى AGI العام.

الحكم المؤقت لهذا الإصدار: إطار مقترح يحتاج إلى تحقق تجريبي مستقل. أما المراجع [R39] [R36] [R30] [R29] [R18] [R10] [R05] [R40] فتثبت أن أجزاء من فضاء المشكلة مدروسة؛ ولا تمنح النظرية المساعفية تصديقًا مستقلاً قبل تنفيذ البرنامج التجريبي المقترح.

الاعتراضات الأقوى والرد التجريبي

إعادة تسمية	قارن الوظيفة بأقرب Baseline؛ إن لم يوجد فرق ادمج المصطلح.
كلفة تنسيق	احسب Tokens وLatency وHuman load في End-to-End.
انحياز القياس	جمد Rubric وشغل Judge مستقلاً وحالات عمياء.
هشاشة النقل	اختبر أدوات وصياغات ومجالاً لم يستخدم في الضبط.
خطر السلطة	شغل adversarial tasks وتحقق من الصفر في الأفعال غير المأذونة.

أقوى بديل تفسيري هو أن التحسن يأتي من مزيد من Context أو Compute أو Steps، لا من Moussaif Fattah Open-Intelligence Program (MFOIP). لذلك تنشأ نسخة Budget-matched baseline تستهلك الموارد نفسها من دون الآلية المقترحة. إذا تعادلت النتائج، لا تمنح النظرية الفضل. وبديل ثان أن Judge يفضل لغة النظام؛ يعالج ذلك بتجريد الأسماء وتعمية المصدر. وبديل ثالث أن Benchmark تلوث بالتدريب؛ يعالج بمهام مشتقة حديثًا وقواعد توليد معلنة مع عينات مخفية.

تنتهي المحكمة بقرار هندسي لا بحكم وجودي. Adopt يعني أن المكون يحسن نطاقًا معلومًا مع حراس؛ Revise يعني أن الآلية واعدة لكن Contract أو Metric ضعيف؛ Merge يعني أن الاسم لا يختلف عن مكون قائم؛ Reject يعني أن الضرر أو الكلفة يغلبان المنفعة؛ Hold يعني أن الدليل غير كاف. ويحفظ القرار Residual Risk وخطة إعادة الاختبار. إذا تكرر سباق قدرات بلا حوكمة بعد إصلاحين مستقلين، تنتقل الأولوية من تحسين Prompt إلى إعادة تصميم المعمارية أو حذف المكون.

بطاقات الأدوات 1-5

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal Contract وPrecondition وAuthorization وVerification.

عقد المهمة المساعفي - Moussaif Mission Contract
الوظيفة: يثبت الهدف والناتج وحد القبول قبل التخطيط. | المدخل: طلب المستخدم والقيود | المخرج: Mission Contract versioned | الحارس: لا يعتمد عقدًا بلا مالك قرار.

محدد النطاق المساعفي - Moussaif Scope Boundary
الوظيفة: يفصل ما يدخل في المهمة عما يحتاج إلى تفويض جديد. | المدخل: العقد والبيئة | المخرج: In/Out/Ask map | الحارس: التوسع المادي يوقف التنفيذ.

مترجم القبول المساعفي - Moussaif Acceptance Compiler
الوظيفة: يحول اللغة الطبيعية إلى اختبارات نجاح قابلة للتشغيل. | المدخل: وصف الناتج | المخرج: Acceptance tests | الحارس: لا يستبدل الحكم البشري في القيمة.

سجل الدعاوى المساعفي - Moussaif Claim Ledger
الوظيفة: يسجل كل دعوى ومصدرها ودرجتها ونطاقها. | المدخل: نص أو قرار | المخرج: Typed claim records | الحارس: الدعوى غير المسندة تبقى معلقة.

مدرج الدليل المساعفي - Moussaif Evidence Grader
الوظيفة: يصنف E0-E5 ويفصل الفرضية عن التكرار المستقل. | المدخل: دعوى وأدلة | المخرج: Evidence grade | الحارس: لا ترقية بلا حدث تحقق.

تسلسل الإدماج في النواة

- عقد المهمة المساعفي: يبدأ بطلب المستخدم والقيود، ويقبل المخرج فقط بعد اختبار الحارس «لا يعتمد عقدًا بلا مالك قرار.».
- محدد النطاق المساعفي: يبدأ بالعقد والبيئة، ويقبل المخرج فقط بعد اختبار الحارس «التوسع المادي يوقف التنفيذ.».
- مترجم القبول المساعفي: يبدأ بوصف الناتج، ويقبل المخرج فقط بعد اختبار الحارس «لا يستبدل الحكم البشري في القيمة.».
- سجل الدعاوى المساعفي: يبدأ بنص أو قرار، ويقبل المخرج فقط بعد اختبار الحارس «الدعوى غير المسندة تبقى معلقة.».
- مدرج الدليل المساعفي: يبدأ بدعوى وأدلة، ويقبل المخرج فقط بعد اختبار الحارس «لا ترقية بلا حدث تحقق.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 6-10

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal Contract وPrecondition وAuthorization وVerification.

رابط المصدر المساعفي - Moussaif Source Binder
الوظيفة: يربط الجملة بالنطاق الداعم داخل المصدر. | المدخل: دعوى ووثيقة | المخرج: Provenance edge | الحارس: الرابط وحده ليس إسنادًا.

مراقب الحداثة المساعفي - Moussaif Freshness Monitor
الوظيفة: يكشف المعلومات التي قد تتقادم قبل القرار. | المدخل: Claim metadata | المخرج: Freshness alert | الحارس: يبحث فقط حين تؤثر الحداثة.

مولد السؤال الحاكمي - Moussaif Question Generator
الوظيفة: ينتج أسئلة منافسة تكشف فروقًا قابلة للاختبار. | المدخل: هدف وحالة | المخرج: Question set | الحارس: يمنع التكاثر بلا كسب معلومات.

مقياس Q-MCD - Moussaif Q-MCD Scorer
الوظيفة: يقيس عمق السؤال وأثره وتمييزه وضوضاءه. | المدخل: أسئلة مرشحة | المخرج: Ranked questions | الحارس: الأوزان task-dependent ومعلنة.

خريطة العمى المساعفية - Moussaif Blind-Spot Mapper
الوظيفة: يحدد الفئات التي لا يغطيها السياق أو الفريق. | المدخل: خطة ودليل | المخرج: Blindness map | الحارس: لا يدعي الاكتمال من الغياب.

تسلسل الإدماج في النواة

- رابط المصدر المساعفي: يبدأ بدعوى ووثيقة، ويقبل المخرج فقط بعد اختبار الحارس «الرابط وحده ليس إسنادًا.».
- مراقب الحداثة المساعفي: يبدأ بـClaim metadata، ويقبل المخرج فقط بعد اختبار الحارس «يبحث فقط حين تؤثر الحداثة.».
- مولد السؤال الحاكمي: يبدأ بهدف وحالة، ويقبل المخرج فقط بعد اختبار الحارس «يمنع التكاثر بلا كسب معلومات.».
- مقياس Q-MCD: يبدأ بأسئلة مرشحة، ويقبل المخرج فقط بعد اختبار الحارس «الأوزان task-dependent ومعلنة.».
- خريطة العمى المساعفية: يبدأ بخطة ودليل، ويقبل المخرج فقط بعد اختبار الحارس «لا يدعي الاكتمال من الغياب.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصمًيا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 11-15

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal Contract وPrecondition وAuthorization وVerification.

حاسبة الفرق السببي - Moussaif MCD Calculator
الوظيفة: تحسب قيمة الفرق مقابل الضوضاء والكلفة والخطر. | المدخل: ميزات مرشح | المخرج: MCD score | الحارس: تختبر حساسية الأوزان.

مستخرج الفرق الفاعل - Moussaif Difference Extractor
الوظيفة: يفصل التغير الذي يبدل القرار عن إعادة الصياغة. | المدخل: بدائل أو نسخ | المخرج: Decision-relevant deltas | الحارس: لا يساوي الجدة بالقيمة.

الموجه الوعائي المساعفي - Moussaif Vascular Router
الوظيفة: يوزع الحساب على العقد ذات الحاجة والأثر الأعلى. | المدخل: Task graph وBudgets | المخرج: Resource allocation | الحارس: لا يجوع السلامة والتحقق.

صمام الموارد المساعفي - Moussaif Resource Valve
الوظيفة: يرفع أو يخفض Tokens وTools وفق Telemetry. | المدخل: حالة التدفق | المخرج: Budget adjustment | الحارس: كل رفع له سقف وسبب.

خريطة الفجوات المساعفية - Moussaif Gap Map
الوظيفة: يمسح التعريف والحد والدليل والآلية والقياس والتنفيذ. | المدخل: Artifact draft | المخرج: Prioritized gaps | الحارس: يطلب طبقة غائبة لا حشوًا.

تسلسل الإدماج في النواة

- حاسبة الفرق السببي: يبدأ بـميزات مرشح، ويقبل المخرج فقط بعد اختبار الحارس «تختبر حساسية الأوزان.».
- مستخرج الفرق الفاعل: يبدأ ببدائل أو نسخ، ويقبل المخرج فقط بعد اختبار الحارس «لا يساوي الجدة بالقيمة.».
- الموجه الوعائي المساعفي: يبدأ بـTask graph وBudgets، ويقبل المخرج فقط بعد اختبار الحارس «لا يجوع السلامة والتحقق.».
- صمام الموارد المساعفي: يبدأ بحالة التدفق، ويقبل المخرج فقط بعد اختبار الحارس «كل رفع له سقف وسبب.».
- خريطة الفجوات المساعفية: يبدأ بـArtifact draft، ويقبل المخرج فقط بعد اختبار الحارس «يطلب طبقة غائبة لا حشوًا.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 16-20

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

محكمة الهدم المساعفية - Moussaif Falsification Court
الوظيفة: تختبر الادعاء ببدائل وتجارب عكسية ومعيار سابق. | المدخل: Claim وArtifacts | المخرج: Reasoned verdict | الحارس: يفصل Builder عن Judge.

مسبك المثال المضاد - Moussaif Counterexample Forge
الوظيفة: يولد حالات تكسر الافتراض في حدوده الأخطر. | المدخل: فرضية وشروط | المخرج: Counterexample suite | الحارس: يميز الواقعي من المستحيل.

متتبع الآلية المساعفي - Moussaif Mechanism Trace
الوظيفة: يربط الحالة بالفعل والوسيط والملاحظة. | المدخل: Trajectory | المخرج: Causal trace | الحارس: يعلن العقد غير المرصودة.

مولد الآلية البديلة - Moussaif Causal Alternative
الوظيفة: ينتج تفسيرات منافسة للنتيجة نفسها. | المدخل: أثر مرصود | المخرج: Alternative mechanisms | الحارس: لا يقبل البديل بلا تنبؤ مميز.

فارق حالة العالم - Moussaif World-State Delta
الوظيفة: يسجل ما تغير مع الزمن والمصدر وعدم اليقين. | المدخل: Old/New observations | المخرج: Typed delta | الحارس: لا يمحو تعارضًا غير محلول.

تسلسل الإدماج في النواة

- محكمة الهدم المساعفية: يبدأ بـClaim وArtifacts، ويقبل المخرج فقط بعد اختبار الحارس «يفصل Builder عن Judge.».
- مسبك المثال المضاد: يبدأ بفرضية وشروط، ويقبل المخرج فقط بعد اختبار الحارس «يميز الواقعي من المستحيل.».
- متتبع الآلية المساعفي: يبدأ بـTrajectory، ويقبل المخرج فقط بعد اختبار الحارس «يعلن العقد غير المرصودة.».
- مولد الآلية البديلة: يبدأ بأثر مرصود، ويقبل المخرج فقط بعد اختبار الحارس «لا يقبل البديل بلا تنبؤ مميز.».
- فارق حالة العالم: يبدأ بـOld/New observations، ويقبل المخرج فقط بعد اختبار الحارس «لا يمحو تعارضًا غير محلول.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 21-25

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

راسم شبكة المشاريع - Moussaif Project Graph Mapper
الوظيفة: يربط المشاريع والأصول والقرارات والاعتماديات. | المدخل: Project registry | المخرج: Project-world graph | الحارس: يفصل Proposed عن Observed.

مخروط الاعتماد المساعفي - Moussaif Dependency Cone
الوظيفة: يتنبأ بما قد يتأثر بتغيير قبل تنفيذه. | المدخل: Target change | المخرج: Impact cone | الحارس: يفحص الاعتماد الخفي والـGenerated files.

سجل الأصول القانونية - Moussaif Canonical Asset Registry
الوظيفة: يحفظ النسخة والرابط والمالك والحالة القانونية لكل أصل. | المدخل: Assets metadata | المخرج: Canonical registry | الحارس: لا يساوي الأحدث بالمعتمد.

مجزئ HyperKnowledge - Moussaif KB Unitizer
الوظيفة: يحول الوثيقة إلى وحدات مستقلة ذات وظيفة ومراجع. | المدخل: Long document | المخرج: KB units | الحارس: يمنع قطع التعريف عن حدوده.

ناسج الاستعلام المساعفي - Moussaif Query Weaver
الوظيفة: ينتج استعلامات دلالية ولفظية وسببية متعددة. | المدخل: Question وClaim type | المخرج: Query bundle | الحارس: يحافظ على قصد السؤال.

تسلسل الإدماج في النواة

- راسم شبكة المشاريع: يبدأ بـProject registry، ويقبل المخرج فقط بعد اختبار الحارس «يفصل Proposed عن Observed.».
- مخروط الاعتماد المساعفي: يبدأ بـTarget change، ويقبل المخرج فقط بعد اختبار الحارس «يفحص الاعتماد الخفي والـGenerated files.».
- سجل الأصول القانونية: يبدأ بـAssets metadata، ويقبل المخرج فقط بعد اختبار الحارس «لا يساوي الأحدث بالمعتمد.».
- مجزئ HyperKnowledge: يبدأ بـLong document، ويقبل المخرج فقط بعد اختبار الحارس «يمنع قطع التعريف عن حدوده.».
- ناسج الاستعلام المساعفي: يبدأ بـQuestion وClaim type، ويقبل المخرج فقط بعد اختبار الحارس «يحافظ على قصد السؤال.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 26-30

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

المسترجع الهجين - Moussaif Hybrid Retriever
الوظيفة: يدمج Lexical وSemantic وGraph Retrieval. | المدخل: Query bundle | المخرج: Candidate passages | الحارس: يسجل قناة كل نتيجة.

معيد ترتيب الدليل - Moussaif Evidence Reranker
الوظيفة: يرتب بالدعم والحداثة والثقة والتنوع لا بالتشابه فقط. | المدخل: Candidates وClaim | المخرج: Ranked evidence | الحارس: يعاقب التكرار والمصدر الواحد.

فاحص استلزام الاستشهاد - Moussaif Citation Entailment Check
الوظيفة: يفحص هل المقطع يدعم الجملة فعلاً. | المدخل: Claim وSpan | المخرج: Support verdict | الحارس: لا يحكم من العنوان.

بوابة ترقية الذاكرة - Moussaif Memory Promotion Gate
الوظيفة: تمنع ترقية Episode إلى معرفة دائمة بلا تكرار وقيمة. | المدخل: Memory candidate | المخرج: Promote/Hold/Reject | الحارس: يفحص الحساسية وحق الحذف.

مجدول النسيان - Moussaif Forgetting Scheduler
الوظيفة: يطبق Retention وDecay وحذفًا قابلاً للإثبات. | المدخل: Memory policy | المخرج: Deletion schedule | الحارس: يحفظ السجل دون حفظ المحتوى المحذوف.

تسلسل الإدماج في النواة

- المسترجع الهجين: يبدأ بـQuery bundle، ويقبل المخرج فقط بعد اختبار الحارس «يسجل قناة كل نتيجة.».
- معيد ترتيب الدليل: يبدأ بـCandidates وClaim، ويقبل المخرج فقط بعد اختبار الحارس «يعاقب التكرار والمصدر الواحد.».
- فاحص استلزام الاستشهاد: يبدأ بـClaim وSpan، ويقبل المخرج فقط بعد اختبار الحارس «لا يحكم من العنوان.».
- بوابة ترقية الذاكرة: يبدأ بـMemory candidate، ويقبل المخرج فقط بعد اختبار الحارس «يفحص الحساسية وحق الحذف.».
- مجدول النسيان: يبدأ بـMemory policy، ويقبل المخرج فقط بعد اختبار الحارس «يحفظ السجل دون حفظ المحتوى المحذوف.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 31-35

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

مترجم كبسولة السياق - Moussaif Context Capsule Compiler
الوظيفة: يضغط الحالة والقرار والدليل والمهام المفتوحة. | المدخل: Trace وBudget | المخرج: Portable capsule | الحارس: يختبر قابلية استعادة المهمة.

دفتر اقتصاد الرموز - Moussaif Token Ledger
الوظيفة: ينسب الكلفة إلى المرحلة والنتيجة وإعادة العمل. | المدخل: Usage events | المخرج: Cost/value report | الحارس: لا يخفض Quality Floor.

سجل HyperSkills - Moussaif Skill Registry
الوظيفة: يفهرس Specs وVersions والمتطلبات والاختبارات. | المدخل: Skill packages | المخرج: Queryable registry | الحارس: يرفض Skill بلا Verification.

مطابق المشغلات - Moussaif Trigger Matcher
الوظيفة: يطابق نية المهمة بمشغلات دقيقة قابلة للتفسير. | المدخل: Task وTriggers | المخرج: Skill candidates | الحارس: لا يستخدم Similarity وحده.

بوابة الشروط السابقة - Moussaif Precondition Gate
الوظيفة: يمنع Skill إذا غابت ملفات أو صلاحيات أو قيم أساسية. | المدخل: Skill وState | المخرج: Ready/Blocked report | الحارس: لا يخمن قيمة مادية.

تسلسل الإدماج في النواة

- مترجم كبسولة السياق: يبدأ بـTrace وBudget، ويقبل المخرج فقط بعد اختبار الحارس «يختبر قابلية استعادة المهمة.».
- دفتر اقتصاد الرموز: يبدأ بـUsage events، ويقبل المخرج فقط بعد اختبار الحارس «لا يخفض Quality Floor.».
- سجل HyperSkills: يبدأ بـSkill packages، ويقبل المخرج فقط بعد اختبار الحارس «يرفض Skill بلا Verification.».
- مطابق المشغلات: يبدأ بـTask وTriggers، ويقبل المخرج فقط بعد اختبار الحارس «لا يستخدم Similarity وحده.».
- بوابة الشروط السابقة: يبدأ بـSkill وState، ويقبل المخرج فقط بعد اختبار الحارس «لا يخمن قيمة مادية.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصمًيا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 36-40

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

مزيل تكرار المهارات - Moussaif Skill Deduplicator
الوظيفة: يقارن البنية والوظيفة ويفتح Merge أو Version. | المدخل: Skill set | المخرج: Merge proposal | الحارس: يحافظ على حالات الاختلاف الحقيقية.

مولد HyperSkill - Moussaif Skill Synthesizer
الوظيفة: يجرد Trajectory ناجحة إلى عقد قابل لإعادة الاستخدام. | المدخل: Validated trajectory | المخرج: Candidate skill | الحارس: تبقى Candidate حتى Benchmark.

مورث إصدارات المهارة - Moussaif Skill Versioner
الوظيفة: يحفظ Lineage والتغييرات والتوافق والعودة. | المدخل: Old/New spec | المخرج: Version record | الحارس: لا يستبدل Canonical بصمت.

الموجه الفائق - Moussaif HyperRouter
الوظيفة: يختار Skill أوTool أوModel وفق المنفعة المعدلة بالمخاطر. | المدخل: Task وRegistry | المخرج: Route decision | الحارس: يسجل البدائل وRouting regret.

منتقي محفظة النماذج - Moussaif Model Portfolio Selector
الوظيفة: يطابق قدرات مقاسة محليًا مع نوع المهمة. | المدخل: Task profile | المخرج: Model/ensemble plan | الحارس: لا يثق بالاسم التجاري وحده.

تسلسل الإدماج في النواة

- مزيل تكرار المهارات: يبدأ بـSkill set، ويقبل المخرج فقط بعد اختبار الحارس «يحافظ على حالات الاختلاف الحقيقية.».
- مولد HyperSkill: يبدأ بـValidated trajectory، ويقبل المخرج فقط بعد اختبار الحارس «تبقى Candidate حتى Benchmark.».
- مورث إصدارات المهارة: يبدأ بـOld/New spec، ويقبل المخرج فقط بعد اختبار الحارس «لا يستبدل Canonical بصمت.».
- الموجه الفائق: يبدأ بـTask وRegistry، ويقبل المخرج فقط بعد اختبار الحارس «يسجل البدائل وRouting regret.».
- منتقي محفظة النماذج: يبدأ بـTask profile، ويقبل المخرج فقط بعد اختبار الحارس «لا يثق بالاسم التجاري وحده.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 41-44

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

فهرس الأدوات - Moussaif Tool Catalog
الوظيفة: يحفظ Schema والأثر الجانبي والتكلفة والصلاحية. | المدخل: Tool manifests | المخرج: Validated catalog | الحارس: يعزل الأدوات غير الموثوقة.

فاحص معاملات الأداة - Moussaif Tool Argument Validator
الوظيفة: يفحص النوع والنطاق والمقصد قبل الاستدعاء. | المدخل: Arguments وSchema | المخرج: Safe call packet | الحارس: يرفض القيم المولدة الخطرة.

مصفوفة الإذن - Moussaif Permission Matrix
الوظيفة: يربط Actor وTarget وEffect ومدة السلطة. | المدخل: Authorization context | المخرج: Capability token | الحارس: أقل امتياز ومدة محدودة.

مهايئ التشغيل الجاف - Moussaif Dry-Run Adapter
الوظيفة: يحاكي أو يعرض أثر التغيير قبل الالتزام. | المدخل: Tool call | المخرج: Preview/Diff | الحارس: لا يقدم المحاكاة كضمان.

تسلسل الإدماج في النواة

- فهرس الأدوات: يبدأ بـTool manifests، ويقبل المخرج فقط بعد اختبار الحارس «يعزل الأدوات غير الموثوقة.».
- فاحص معاملات الأداة: يبدأ بـArguments وSchema، ويقبل المخرج فقط بعد اختبار الحارس «يرفض القيم المولدة الخطرة.».
- مصفوفة الإذن: يبدأ بـAuthorization context، ويقبل المخرج فقط بعد اختبار الحارس «أقل امتياز ومدة محدودة.».
- مهايئ التشغيل الجاف: يبدأ بـTool call، ويقبل المخرج فقط بعد اختبار الحارس «لا يقدم المحاكاة كضمان.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 45-48

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

محقق خرج الأداة - Moussaif Output Validator
الوظيفة: يفحص Schema والاكتمال والدلالة بعد النجاح التقني. | المدخل: Raw output | المخرج: Validated result | الحارس: HTTP 200 ليس نجاح مهمة.

مترجم شبكة الخطة - Moussaif Plan-Lattice Compiler
الوظيفة: يبني DAG مع بدائل ونقاط قرار وشروط جاهزية. | المدخل: Goal contract | المخرج: Plan lattice | الحارس: لا يوازي خطوات تعتمد بعضها على بعض.

مراقب المسار الحرج - Moussaif Critical-Path Monitor
الوظيفة: يتتبع التأخير والاعتماد الذي يهدد النتيجة. | المدخل: Plan telemetry | المخرج: Critical alerts | الحارس: لا يقتل المسارات التجريبية المفيدة.

متحكم الحلقة الفائقة - Moussaif HyperLoop Controller
الوظيفة: يدير Observe-Plan-Act-Verify ضمن Budget. | المدخل: Loop state | المخرج: Next action/stop | الحارس: كل دورة تحتاج Delta جديدًا.

تسلسل الإدماج في النواة

- محقق خرج الأداة: يبدأ بـRaw output، ويقبل المخرج فقط بعد اختبار الحارس «HTTP 200 ليس نجاح مهمة.».
- مترجم شبكة الخطة: يبدأ بـGoal contract، ويقبل المخرج فقط بعد اختبار الحارس «لا يوازي خطوات تعتمد بعضها على بعض.».
- مراقب المسار الحرج: يبدأ بـPlan telemetry، ويقبل المخرج فقط بعد اختبار الحارس «لا يقتل المسارات التجريبية المفيدة.».
- متحكم الحلقة الفائقة: يبدأ بـLoop state، ويقبل المخرج فقط بعد اختبار الحارس «كل دورة تحتاج Delta جديدًا.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 49-52

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

كاشف الركود - Moussaif Stagnation Detector
الوظيفة: يكشف تكرار الأفكار أو الأفعال دون تقدم. | المدخل: Iteration history | المخرج: Stagnation signal | الحارس: يميز التحقق الضروري عن التكرار.

حارس الإنهاء - Moussaif Termination Guard
الوظيفة: يفرض شروط النجاح أو العجز أو الخطر أو نفاد الميزانية. | المدخل: State وContract | المخرج: Stop/escalate verdict | الحارس: لا يحول Persistence إلى حلقة لا نهائية.

ميثاق الوكيل الفرعي - Moussaif Subagent Charter
الوظيفة: يحدد مهمة مستقلة ومخرجًا وعقد دمج وسلطة. | المدخل: Decomposed task | المخرج: Agent charter | الحارس: لا يفوض عملاً غير قابل للفصل.

محلل التكاملية - Moussaif Complementarity Analyzer
الوظيفة: يقيس هل يضيف الدور منظورًا أو دليلاً هامشيًا. | المدخل: Roles وOutputs | المخرج: Contribution matrix | الحارس: يحذف الأدوار الزخرفية.

تسلسل الإدماج في النواة

- كاشف الركود: يبدأ بـIteration history، ويقبل المخرج فقط بعد اختبار الحارس «يميز التحقق الضروري عن التكرار».
- حارس الإنهاء: يبدأ بـState وContract، ويقبل المخرج فقط بعد اختبار الحارس «لا يحول Persistence إلى حلقة لا نهائية».
- ميثاق الوكيل الفرعي: يبدأ بـDecomposed task، ويقبل المخرج فقط بعد اختبار الحارس «لا يفوض عملاً غير قابل للفصل».
- محلل التكاملية: يبدأ بـRoles وOutputs، ويقبل المخرج فقط بعد اختبار الحارس «يحذف الأدوار الزخرفية».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 53-56

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

عقد الدمج - Moussaif Merge Contract
الوظيفة: يثبت صيغة التسليم وحسم التعارض والمالك النهائي. | المدخل: Parallel tasks | المخرج: Merge schema | الحارس: يمنع الكتابة المتزامنة المتعارضة.

فاحص استقلال القاضي - Moussaif Judge Independence Check
الوظيفة: يكشف تسرب خطة الباني أو Rubric بعدي. | المدخل: Court setup | المخرج: Independence score | الحارس: لا يمنح ختمًا ذاتيًا.

كاشف فخ الإجماع - Moussaif Consensus-Trap Detector
الوظيفة: يميز الاتفاق المدعوم من تحيز مشترك أو نسخ متقارب. | المدخل: Agent verdicts | المخرج: Consensus diagnosis | الحارس: يطلب دليلاً لا أصواتًا.

حاكم حدود الاستدعاء - Moussaif Rate-Limit Governor
الوظيفة: يجدول Calls ويحجز هامش التحقق ويطبق Backoff. | المدخل: Quota وQueue | المخرج: Call schedule | الحارس: يمنع Retry storm.

تسلسل الإدماج في النواة

- عقد الدمج: يبدأ بـParallel tasks، ويقبل المخرج فقط بعد اختبار الحارس «يمنع الكتابة المتزامنة المتعارضة.».
- فاحص استقلال القاضي: يبدأ بـCourt setup، ويقبل المخرج فقط بعد اختبار الحارس «لا يمنح ختمًا ذاتيًا.».
- كاشف فخ الإجماع: يبدأ بـAgent verdicts، ويقبل المخرج فقط بعد اختبار الحارس «يطلب دليلاً لا أصواتًا.».
- حاكم حدود الاستدعاء: يبدأ بـQuota وQueue، ويقبل المخرج فقط بعد اختبار الحارس «يمنع Retry storm.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 57-60

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

مصفوفة البدائل - Moussaif Fallback Matrix
الوظيفة: يربط كل فشل بديل متوافق وحد جودة أدنى. | المدخل: Failure class | المخرج: Fallback route | الحارس: لا ينتقل بين APIs غير متكافئة عمياًنا.

حارس عدم تكرار الأثر - Moussaif Idempotency Guard
الوظيفة: يمنع إعادة فعل خارجي سبق نجاحه. | المدخل: Action وTrace | المخرج: Proceed/skip | الحارس: يتحقق من الأثر لا من الذاكرة فقط.

قاطع الدائرة - Moussaif Circuit Breaker
الوظيفة: يعزل مكونًا يفشل متكررًا لحماية النظام. | المدخل: Health events | المخرج: Open/half-open/closed | الحارس: يحدد Probe آمنًا للعودة.

مخطط التراجع - Moussaif Rollback Planner
الوظيفة: يبني طريق عودة واختبار سلامته قبل التغيير. | المدخل: Mutation plan | المخرج: Rollback procedure | الحارس: لا يعتمد Backup غير مجرب.

تسلسل الإدماج في النواة

- مصفوفة البدائل: يبدأ بـFailure class، ويقبل المخرج فقط بعد اختبار الحارس «لا ينتقل بين APIs غير متكافئة عمياًنا.».
- حارس عدم تكرار الأثر: يبدأ بـAction وTrace، ويقبل المخرج فقط بعد اختبار الحارس «يتحقق من الأثر لا من الذاكرة فقط.».
- قاطع الدائرة: يبدأ بـHealth events، ويقبل المخرج فقط بعد اختبار الحارس «يحدد Probe آمنًا للعودة.».
- مخطط التراجع: يبدأ بـMutation plan، ويقبل المخرج فقط بعد اختبار الحارس «لا يعتمد Backup غير مجرب.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 61-64

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

دفتر القياس التشغيلي - Moussaif Telemetry Ledger
الوظيفة: يجمع القرار والأداة والكلفة والدليل والنتيجة في Trace. | المدخل: Structured events | المخرج: Replayable ledger | الحارس: يخفي الأسرار قبل التسجيل.

منقح الأسرار - Moussaif Secret Redactor
الوظيفة: يمنع Keys وPII من Context وLogs وArtifacts. | المدخل: Text/events | المخرج: Redacted payload | الحارس: لا يطبع السر في رسالة الخطأ.

حارس حقن التعليمات - Moussaif Injection Sentinel
الوظيفة: يفصل تعليمات Control عن بيانات المصادر غير الموثوقة. | المدخل: Retrieved content | المخرج: Trust-labeled data | الحارس: النص الخارجي لا يغير السياسة.

راسم حدود البيانات - Moussaif Data-Boundary Mapper
الوظيفة: يرسم أين تدخل البيانات وتعالج وتخزن وتخرج. | المدخل: System topology | المخرج: Data-flow map | الحارس: يضمّن أدوات الطرف الثالث.

تسلسل الإدماج في النواة

- دفتر القياس التشغيلي: يبدأ بـStructured events، ويقبل المخرج فقط بعد اختبار الحارس «يخفي الأسرار قبل التسجيل.».
- منقح الأسرار: يبدأ بـText/events، ويقبل المخرج فقط بعد اختبار الحارس «لا يطبع السر في رسالة الخطأ.».
- حارس حقن التعليمات: يبدأ بـRetrieved content، ويقبل المخرج فقط بعد اختبار الحارس «النص الخارجي لا يغير السياسة.».
- راسم حدود البيانات: يبدأ بـSystem topology، ويقبل المخرج فقط بعد اختبار الحارس «يضمّن أدوات الطرف الثالث.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 65-68

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

ساعة الاحتفاظ الخصوصي - Moussaif Privacy Retention Clock
الوظيفة: يفرض انتهاء الغرض والحذف وإثباته. | المدخل: Data class وPolicy | المخرج: Retention events | الحارس: النسخ والمخابئ ضمن النطاق.

باني Benchmark المساعفي - Moussaif Benchmark Builder
الوظيفة: يولد مهامًا وVariants وBaselines وAblations. | المدخل: Capability claim | المخرج: Evaluation suite | الحارس: يثبت معيار النجاح قبل التشغيل.

مقيم مسار الوكيل - Moussaif Trajectory Evaluator
الوظيفة: يقيس الطريق والأخطاء والكلفة لا Output فقط. | المدخل: Agent trace | المخرج: Trajectory score | الحارس: يعاقب الأثر غير المصرح.

مقياس المعايرة - Moussaif Calibration Meter
الوظيفة: يقارن الثقة باحتمال الصواب المشاهد. | المدخل: Predictions وOutcomes | المخرج: Calibration report | الحارس: لا يقبل ثقة رقمية بلا تعريف.

تسلسل الإدماج في النواة

- ساعة الاحتفاظ الخصوصي: يبدأ بـData class وPolicy، ويقبل المخرج فقط بعد اختبار الحارس «النسخ والمخابئ ضمن النطاق.».
- باني Benchmark المساعفي: يبدأ بـCapability claim، ويقبل المخرج فقط بعد اختبار الحارس «يثبت معيار النجاح قبل التشغيل.».
- مقيم مسار الوكيل: يبدأ بـAgent trace، ويقبل المخرج فقط بعد اختبار الحارس «يعاقب الأثر غير المصرح.».
- مقياس المعايرة: يبدأ بـPredictions وOutcomes، ويقبل المخرج فقط بعد اختبار الحارس «لا يقبل ثقة رقمية بلا تعريف.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

بطاقات الأدوات 69-72

Tool contracts: purpose, input, output, guard

كل بطاقة أدناه Spec أولي لأداة منطقية. قد تنفذ كـSkill أو Function أو Service أو Policy Check. الاسم لا يمنحها صلاحية؛ الاستدعاء يمر من Goal وContract وPrecondition وAuthorization وVerification.

لوحة فرضية القيمة - Moussaif Value-Hypothesis Canvas
الوظيفة: يربط الحاجة والآلية والمقياس والكلفة ودليل الثقة. | المدخل: Need evidence | المخرج: Testable value hypothesis | الحارس: يمنع وعود AGI غير المثبتة.

مصمم التجربة الأولية - Moussaif Pilot Experiment Designer
الوظيفة: يصغر المخاطرة ويحدد عينة وضابطًا وقرار توسع. | المدخل: Value hypothesis | المخرج: Pilot protocol | الحارس: لا يستخدم Spam أو موافقة غامضة.

باني بيان النشر التراكمي - Moussaif Overlay Manifest Builder
الوظيفة: يصف الملفات والـAdapters والسياسات والإصدارات. | المدخل: Runtime target | المخرج: Overlay manifest | الحارس: لا يدعي تعديل الأوزان الخفية.

اختبار العبور بين runtimes - Moussaif Cross-Runtime Portability Test
الوظيفة: يقارن نفس العقد والمهارة عبر منصات متعددة. | المدخل: Adapters وTest suite | المخرج: Parity report | الحارس: يفصل فرق المنصة عن فرق النموذج.

تسلسل الإدماج في النواة

- لوحة فرضية القيمة: يبدأ بـNeed evidence، ويقبل المخرج فقط بعد اختبار الحارس «يمنع وعود AGI غير المثبتة.».
- مصمم التجربة الأولية: يبدأ بـValue hypothesis، ويقبل المخرج فقط بعد اختبار الحارس «لا يستخدم Spam أو موافقة غامضة.».
- باني بيان النشر التراكمي: يبدأ بـRuntime target، ويقبل المخرج فقط بعد اختبار الحارس «لا يدعي تعديل الأوزان الخفية.».
- اختبار العبور بين runtimes: يبدأ بـAdapters وTest suite، ويقبل المخرج فقط بعد اختبار الحارس «يفصل فرق المنصة عن فرق النموذج.».

تطبق الأدوات في هذه الصفحة كمعاملات صغيرة: Prepare يثبت المدخل والسلطة، Execute أو Evaluate ينتج أثرًا محدودًا، Commit يحفظ النتيجة المقبولة، وAbort يعيد الحالة أو يتركها دون تغيير. يجب أن يحمل كل استدعاء task_id وtool_version وarguments_hash وexpected_effect وtimeout. لا تحفظ القيم الحساسة داخل البطاقة؛ يمر الوصول عبر Capability Token قصير العمر. ويقاس نجاح الأداة بمساهمتها الهامشية في Goal، لا بعدد مرات الاستدعاء.

اختبار مشترك لبطاقات هذه الصفحة
شغل حالة صحيحة، وحقلاً مفقودًا، ومدخلاً خصميًا، وفشلاً جزئيًا، وإعادة الاستدعاء؛ يجب أن يكون السلوك typed وقابلاً للتسجيل والعودة.

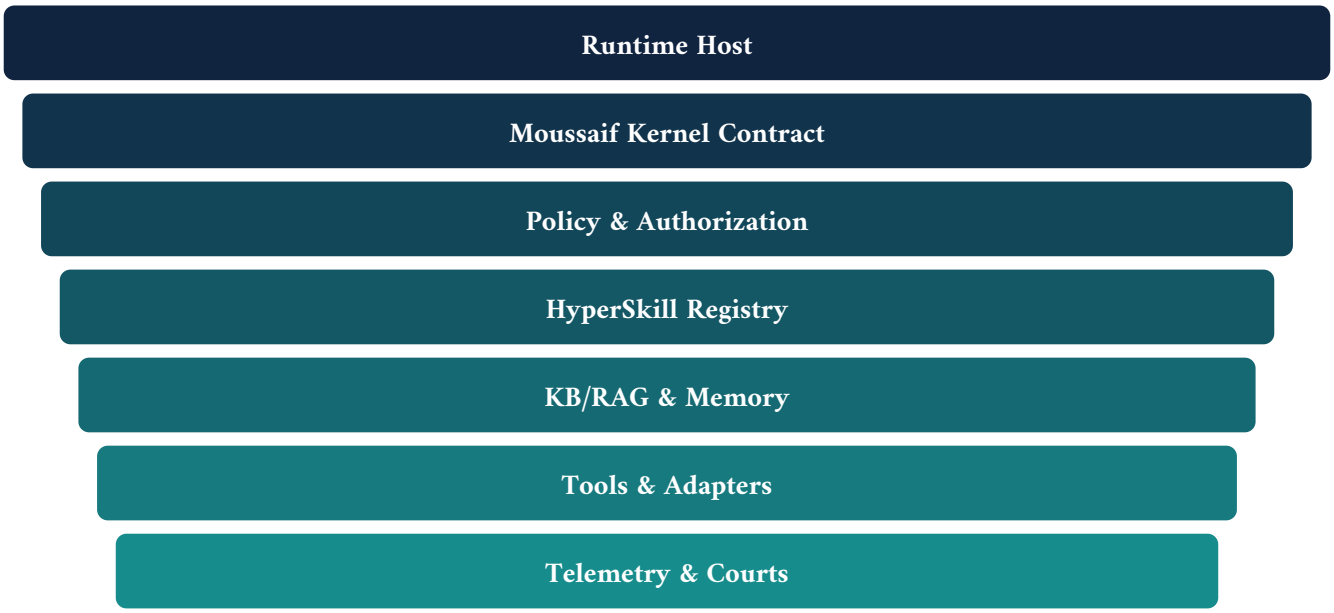
Contract موحد للتنفيذ

prepare	validate input, resolve authority, estimate cost
execute	bounded call with timeout and idempotency key
observe	capture typed output and external side effects
verify	apply acceptance test and evidence grade
commit	update state and append audit event
abort	rollback, quarantine, or escalate with reason

تتشارك الأدوات Failure Taxonomy من خمس فئات: invalid-input قبل التنفيذ، policy-denied عند غياب السلطة، execution-failed عند عطب الأداة، verification-failed حين لا يطابق الأثر العقد، وstate-conflict حين تغير العالم بين التخطيط والالتزام. تحدد الفئة Retry policy: لا يعاد policy-denied، ولا يعاد state-conflict قبل Refresh، ولا يثبت verification-failed كنجاح. وتستخدم Contract tests مزودًا وهميًا ثم Integration test محدودًا، حتى لا يصبح اختبار الأداة نفسه أثرًا عالي المخاطر.

Blueprint النواة والتراكب

Provider-neutral Agentic AGI OS overlay



يُنشر النظام كـOverlay فوق Runtime قائم. يستقبل Host رسائل المستخدم واستدعاءات الأدوات، بينما يحتفظ Moussaif Kernel بعقد المهمة وحالة مجردة. لا تستبدل طبقة التراكب Core model ولا تدعي تعديل أوزانه؛ إنها تضيف ملفات وسياسات وSchemas ومسارات تنفيذ واختبار قابلة للإزالة.

الحد الأدنى للنواة

state.json	goals, world, open_loops, budgets
policy.yaml	authority, privacy, safety, escalation
/skills	versioned skill contracts
/kb	page units, claims, references
/adapters	runtime and tool bindings
/tests	acceptance, security, portability

قيد النشر
إذا لم يدعم Runtime أداة أو Hook، يعلن Adapter الفجوة؛ لا يختلق دعمًا غير موجود.

تعمل صفحة «Blueprint النواة والتراكب» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والـManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

بنية الملفات والـManifest

A reproducible, removable package

يحدد manifest.json الاسم والإصدار والـRuntime targets والملفات والـHashes والمتطلبات وسياسات التحديث والعودة. تحفظ الأسرار خارج الحزمة في Environment أو Secret Store. تحتوي كل Skill على purpose وtriggers وrequires وpreconditions وprocedure وverification وfailure_modes وrollback وversion.

مخطط المجلد

kernel/	state schema, router, loop controller
constitution/	identity, authority, risk policies
hyperknowledge/	KB units and provenance graph
hyperskills/	one versioned contract per skill
adapters/	Hermes, Codex, Claude, Cursor, Ollama
benchmarks/	fixtures, baselines, rubrics
telemetry/	event schemas; no raw secrets

يوقع الإصدار منطقيًا عبر Hash لكل ملف وLockfile للمتطلبات. عند التحديث، تحسب Migration Plan أثر تغيير Schema في Skills وMemory. يسمح Side-by-side install باختبار الإصدار الجديد دون كسر Canonical. ويمنع Skill Registry أسماء متصادمة أو نسخًا بلا Lineage.

Naming convention

<moussaif-<theory>-<domain>-<scope>-v<major.minor.patch>-<date

تعمل صفحة «بنية الملفات والـManifest» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والـManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

مهايئات Hermes وCodex وClaude وCursor وOllama

Capability discovery before configuration

يبدأ كل Adapter بـCapability Discovery: ما الأدوات المتاحة؟ كيف تُمثل الموافقة؟ أين تحفظ الملفات؟ ما حدود Context وRate؟ كيف يقرأ Runtime نتائج الأدوات؟ لا تنقل أوامر منصة إلى أخرى بالاسم فقط. يترجم Adapter العقود المشتركة إلى الصيغة الأصلية ويحافظ على معاني السلطة والتوقف والتحقق.

عقد التوافق

- Hermes: ربط Skills وprovider routing مع إبقاء المفاتيح خارج النص.
- Codex: العمل داخل Workspace وPatch وTests وحدود Sandbox.
- Claude/Cursor: تكييف project rules وtool calls وrepository context.
- Ollama: فحص Context الفعلي وTool support والموارد المحلية.
- Hybrid: عدم إرسال بيانات حساسة إلى Cloud إلا وفق policy.
- Unknown runtime: تشغيل Safe Read-Only profile حتى يكتمل الاختبار.

يقارن Cross-Runtime Portability Test نفس 20 مهمة بعقد موحد. لا يشترط تطابق النص، بل تطابق النجاح والحدود والآثار. تسجل فروق النموذج عن فروق Adapter، وتمنع المقارنة غير العادلة حين تختلف الأدوات أو البيانات. إذا كان أحد runtimes لا يدعم خطوة، تصبح النتيجة Not Supported لا Failure مخفيًا.

قاعدة الأسرار

لا توضع API keys في KB أو Skill أو Script منشور؛ تستخدم placeholders وSecret Store وتدوير المفاتيح المكشوفة.

تعمل صفحة «مهايئات Hermes وCodex وClaude وCursor وOllama» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والـManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

Rollout تدريجي واختبارات القبول

Shadow, sandbox, canary, production

مراحل الترقية

- Static validation: Schemas, references, secrets scan.
- Unit tests: router, policies, skill preconditions.
- Sandbox: synthetic tasks without external side effects.
- Shadow mode: observe decisions without executing them.
- Canary: small reversible task set with human review.
- Production bounded mode: budgets and confirmation gates.
- Post-deploy court: regressions, cost, safety, rollback readiness.

تثبت بوابة القبول قبل Canary: حد أدنى لنجاح المهمة، صفر أفعال غير مأذونة، حد أعلى للكلفة، ونجاح Rollback. لا يسمح بتحسين متوسط النجاح إذا أخفى فئة إخفاق عالية الأثر. وتبقى بعض المهارات Read-only أو Draft-only حتى تجمع Evidence Grade كافيًا.

يستخدم Shadow mode لمقارنة قرار HyperRouter بالمسار الحالي من دون أثر. إذا أظهر فرقًا، يراجع الإنسان الحالات ويصنفها. ولا ينتقل النظام إلى Autonomy أعلى تلقائيًا؛ Autonomy Level قرار حوكمة مستقل عن Capability score.

Definition of done

Installed + tested + observable + reversible + documented + authorized. أي عنصر مفقود يعني أن النشر غير مكتمل.

تعمل صفحة «Rollout تدريجي واختبارات القبول» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

التشغيل اليومي والتطور المحكوم

Operate, learn, prune, and audit

في التشغيل اليومي يجمع النظام Telemetry موجزًا: المهمة، Human correction، Failure class، Outcome، Cost، Tools، Route. يراجع تقرير أسبوعي المهارات غير المستعملة، والمكررة، والمتدهورة، والمصادر القديمة، والFallbacks المتكررة. يتحول النمط الناجح إلى Candidate Skill، والنمط الخطر إلى Regression Test.

إيقاع الحوكمة

- يوميًا: فشل عالي الأثر وأسرار وRate limits.
- أسبوعيًا: Routing regret وSkill reuse وCost anomalies.
- شهريًا: Benchmark drift وdata retention وadapter parity.
- كل إصدار: محكمة هدم مستقلة وRollback drill.
- كل ربع: مراجعة أهداف البحث وحدود ادعاء AGI.

التقليم جزء من التعلم: تحذف Skill لا تضيف منفعة، وتدمج نسخ متشابهة، وتخفض ذاكرة بلا استعمال، وتقاعد مصادر قديمة. ويحتفظ Version Ledger بسبب الإزالة كي لا يعيد النظام الخطأ نفسه. هذا يجعل التطور زيادة في الكفاءة الموثقة لا نموًا لا نهائيًا في الملفات.

مؤشر الصحة

Healthy AGI-OS = verified outcomes + bounded authority + recoverable state + explainable routing - silent debt.

تعمل صفحة «التشغيل اليومي والتطور المحكوم» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

مصفوفة Benchmarks متعددة المحاور

Capability without safety and cost is incomplete

محاور الاختبار

Reasoning	multi-hop, counterfactual, calibration
Knowledge	retrieval, citation entailment, freshness
Agency	planning, tool use, long horizon, recovery
Software	repo tasks, tests, minimal diffs
Web/OS	stateful interaction, permissions, injection
Multi-agent	coordination, merge, marginal contribution
Safety	authorization, privacy, prompt injection
Economy	tokens, latency, calls, human interventions

تستعمل حزم عامة مثل AgentBench وGAIA وSWE-bench وWebArena وOSWorld بوصفها نقاط مقارنة، ثم تضاف مهام خاصة لاختبار Q-MCD وMCD وHyperRouting. لا تخلط النتائج إذا اختلفت الإصدارات والأدوات والوصول إلى الإنترنت. كل نتيجة تحمل Runtime وModel وPrompt/Policy version وDate وBudget.

المقياس الأعلى ليس متوسط الدقة وحده، بل Risk-adjusted verified utility. يسجل الفشل الصامت، والأثر غير المأذون، والمصدر الخاطئ، والعودة الناجحة. ويعرض Pareto frontier بين الجودة والكلفة والزمن بدل اختيار رقم مركب يخفي المفاضلة.

مبدأ المقارنة

قارن نظامًا بنظام تحت الأدوات والميزانية نفسها، لا تقارن Model خامًا بوكيل مسلح دون إعلان الفرق.

تعمل صفحة «مصفوفة Benchmarks متعددة المحاور» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والـManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

تصميم التجربة وAblation

From narrative confidence to causal evidence

النسخ التجريبية

- B0: أقوى Baseline بسيط منصف.
- B1: Kernel دون النظرية الفرعية موضع الاختبار.
- B2: Kernel + component كامل.
- B3: component مع Random/naive router لاختبار التوجيه.
- B4: full system مع court مستقلة.
- B5: transfer domain لم يرَ في الضبط.

يحدد Primary outcome قبل التشغيل، ثم Secondary outcomes للسلامة والكلفة. تستخدم Confidence intervals أو Bootstrap حين تسمح العينة، وتعرض الحالات الخام المهمة. في الاختبارات غير الحتمية تحفظ Seeds وTemperature وProvider date. لا يعاد Prompt tuning على test set؛ وإذا حدث، تنقل العينة إلى Development ويستخرج Test جديد.

لقياس Q-MCD مثلاً، نقارن السؤال الأصلي، وسؤالاً معاد الصياغة بلا Scoring، وسؤال Q-MCD. نحفظ Model وTools ثابتة، ونقيس نجاح الخطة وكسب المعلومات والكلفة. إذا لم يتغير القرار أو تحسن فقط بسبب زيادة Tokens، لا ينسب الفرق للمبدأ. ولمحكمة الهدم نقيس أخطاء اكتشفتها فوق Critic baseline وكلفة الإنذارات الكاذبة.

> **قاعدة النتيجة السلبية**
> عدم التحسن نتيجة؛ يضيق نطاق النظرية ويمنع تضخمها، ولا يحذف من التقرير.

تعمل صفحة «تصميم التجربة وAblation» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

بوابة ادعاء AGI وبرنامج السنوات

No AGI claim by accumulation of features

لا يكفي اجتياز Benchmark واحد، ولا وجود Tool use وMemory وAgents، ولا انبهار المستخدم، لإعلان AGI. تتطلب بوابة الادعاء تعريفًا علنيًا للعمومية، ومهامًا جديدة خارج التدريب، وانتقالاً بين مجالات، واستقلال تقييم، وتحليل أمان، واستدامة الأداء، وبيانًا صريحًا لما لا يستطيع النظام فعله.

مراحل برنامج البحث

- P0 - Specification: schemas, claims, baselines.
- P1 - Component prototypes: Q-MCD, MCD, router, courts.
- P2 - Integrated kernel in sandbox.
- P3 - Cross-runtime and transfer evaluation.
- P4 - Independent replication and red-team.
- P5 - Bounded real-world pilots with human control.
- P6 - Reassessment of the AGI-level hypothesis.

تُحدّث النظرية من نتائج البرنامج: قد تتحول بعض الأسماء إلى أدوات صغيرة، وقد تندمج أخرى، وقد تسقط فرضية الوعائية إذا لم تتفوق على Scheduling أبسط. قوة المشروع ليست في بقاء كل اسم، بل في قدرة المحكمة على التخلص من الجزء غير الضروري مع حفظ ما ثبت.

الادعاء المسموح في v1.0

كتاب يقدم إطارا أصليًا قابلاً للهندسة والاختبار. لا يقدم نظامًا محققًا للAGI ولا نتائج Benchmark مستقلة للإطار الكامل.

تعمل صفحة «بوابة ادعاء AGI وبرنامج السنوات» كوثيقة قرار وليست وصفًا فقط. قبل اعتمادها يحدد Owner، ونسخة Runtime، ومجموعة البيانات، ونافذة التغيير، وRollback owner، وEvidence pack. ويفصل التقرير بين نجاح الإعداد ونجاح المهمة: قد تمر Schemas وTests بينما يفشل Outcome الواقعي، أو ينجح Outcome مرة مع غياب قابلية الإعادة. لذلك يحمل الحكم الحالتين ولا يختزل إحداهما في الأخرى.

Evidence pack الإلزامي

- نسخة العقد والسياسة والManifest وHashes الملفات.
- Baseline ونتيجة ما قبل التغيير مع نفس الميزانية.
- Trace منقح لحالة نجاح وحالة فشل وحالة خصمية.
- قياس Cost وLatency وTool calls والتدخل البشري.
- إثبات Rollback أو سبب عدم انطباقه.
- Residual risks والمالك والموعد المحدد لإعادة الفحص.

المراجع العلمية والتقنية - 1/3

Primary papers, benchmarks, and official risk frameworks

تغطي هذه الصفحة الأسس المعمارية للاستدلال والفعل والاسترجاع والأدوات والذاكرة والتخطيط. تستخدم كأدلة مكونات ومقارنات، ولا يثبت أي مرجع منها نظرية Moussaif الكاملة أو بلوغ AGI.

قواعد الاستعمال

- اقرأ Abstract والمنهج والنتيجة والقيود، لا تستشهد بالعنوان وحده.
- ميز نتيجة الورقة عن تفسير هذا الكتاب وعن النظرية المساعفية المقترحة.
- تحقق من الإصدار والتاريخ والـBenchmark والموارد قبل المقارنة.
- اربط الادعاء بالمقطع الداعم، وسجل ما لا يغطيه المصدر.

[R01] Vaswani et al. (2017). Attention Is All You Need. https://arxiv.org/abs/1706.03762 - الأساس المعماري للـTransformer؛ لا يثبت الوكالة العامة بذاته. خريطة الاستخدام في هذا الكتاب: الفصول 1، 2، 27.

[R02] Pearl, J. (2009). Causality: Models, Reasoning and Inference, 2nd ed. https://www.cambridge.org/core/books/causality/B0046844FAE10CBF274D4ACBDAEB5F5B - مرجع للتمييز بين الارتباط والآلية والتدخل. خريطة الاستخدام في هذا الكتاب: الفصول 6، 11.

[R03] Wei et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. https://arxiv.org/abs/2201.11903 - دليل مكوّن على أثر أمثلة الاستدلال في مهام محددة. خريطة الاستخدام في هذا الكتاب: الفصول 5، 6، 27، 30.

[R04] Wang et al. (2022). Self-Consistency Improves Chain of Thought Reasoning. https://arxiv.org/abs/2203.11171 - اختيار إجابة عبر مسارات استدلال متعددة. خريطة الاستخدام في هذا الكتاب: الفصول 5، 10، 27.

[R05] Yao et al. (2022/2023). ReAct: Synergizing Reasoning and Acting in Language Models. https://arxiv.org/abs/2210.03629 - مزج آثار الاستدلال بالأفعال والتغذية الراجعة من البيئة. خريطة الاستخدام في هذا الكتاب: الفصول 1، 4، 6، 20، 21، 22، 42.

[R06] Yao et al. (2023). Tree of Thoughts: Deliberate Problem Solving with Large Language Models. https://arxiv.org/abs/2305.10601 - بحث مسارات وسيطة وتقييمها بدل مسار توليد واحد. خريطة الاستخدام في هذا الكتاب: الفصول 5، 18، 19.

[R07] Schick et al. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. https://arxiv.org/abs/2302.04761 - تعلم متى وكيف تستدعى APIs في إعداد تجريبي محدد. خريطة الاستخدام في هذا الكتاب: الفصول 10، 18، 19، 21، 28، 37.

[R08] Lewis et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. https://arxiv.org/abs/2005.11401 - دمج ذاكرة بارامترية وغير بارامترية مع الاسترجاع. خريطة الاستخدام في هذا الكتاب: الفصول 8، 11، 13، 14، 15، 28.

[R09] Shinn et al. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. https://arxiv.org/abs/2303.11366 - تغذية راجعة لغوية وذاكرة حلقية دون تحديث أوزان. خريطة الاستخدام في هذا الكتاب: الفصول 9، 16، 20، 36، 37.

[R10] Madaan et al. (2023). Self-Refine: Iterative Refinement with Self-Feedback. https://arxiv.org/abs/2303.17651 - تحسين تكراري، لا يضمن صحة النقد الذاتي أو استقلاله. خريطة الاستخدام في هذا الكتاب: الفصول 10، 20، 26، 42.

[R11] Hao et al. (2023). Reasoning with Language Model is Planning with World Model. https://arxiv.org/abs/2305.14992 - صياغة الاستدلال كتخطيط بتمثيل للحالة والقيمة. خريطة الاستخدام في هذا الكتاب: الفصول 1، 4، 5، 20، 22.

[R12] Wu et al. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. https://arxiv.org/abs/2308.08155 - إطار بحثي للمحادثة والتنسيق بين وكلاء. خريطة الاستخدام في هذا الكتاب: الفصول 2، 7، 13، 17، 18، 19، 24، 25، 37، 40.

[R13] Hong et al. (2023). MetaGPT: Meta Programming for Multi-Agent Collaborative Framework. https://arxiv.org/abs/2308.00352 - تنظيم أدوار وإجراءات في مهام هندسة البرمجيات. خريطة الاستخدام في هذا الكتاب: الفصول 2، 24.

[R14] Li et al. (2023). CAMEL: Communicative Agents for Mind Exploration of LLM Society. https://arxiv.org/abs/2303.17760 - Role-playing واتصال وكلاء ضمن سيناريوهات مولدة. خريطة الاستخدام في هذا الكتاب: الفصول 2، 24.

المراجع العلمية والتقنية - 2/3

Primary papers, benchmarks, and official risk frameworks

تغطي هذه الصفحة التنسيق متعدد الوكلاء، والمواءمة، والأمن، والضغط، وإدارة المخاطر. ينبغي الرجوع إلى النسخة الأصلية لكل مصدر وفحص الإعداد التجريبي وحدود النقل قبل الاستشهاد.

قواعد الاستعمال

- اقرأ Abstract والمنهج والنتيجة والقيود؛ لا تستشهد بالعنوان وحده.
- ميز نتيجة الورقة عن تفسير هذا الكتاب وعن النظرية المساعفية المقترحة.
- تحقق من الإصدار والتاريخ والـBenchmark والموارد قبل المقارنة.
- اربط الادعاء بالمقطع الداعم، وسجل ما لا يغطيه المصدر.

[R15] **Park et al. (2023). Generative Agents: Interactive Simulacra of Human Behavior. https://arxiv.org/abs/2304.03442** - ذاكرة واسترجاع وتأمل وتخطيط في محاكاة اجتماعية. خريطة الاستخدام في هذا الكتاب: الفصول 12، 16، 36.

[R16] **Ha and Schmidhuber (2018). World Models. https://arxiv.org/abs/1803.10122** - تعلم تمثيل مضغوط لديناميكيات البيئة في إعدادات RL. خريطة الاستخدام في هذا الكتاب: الفصول 11، 12، 16.

[R17] **Gou et al. (2023). CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing.** https://arxiv.org/abs/2305.11738 - نقد وتحقق مدعومان بأدوات خارجية. خريطة الاستخدام في هذا الكتاب: الفصول 4، 22، 23، 29، 34، 40، 41.

[R18] **Xi et al. (2023). The Rise and Potential of Large Language Model Based Agents: A Survey.** https://arxiv.org/abs/2309.07864 - مسح للملف الوكيلي؛ يستخدم لخريطة المجال لا لإثبات كل دعوى. خريطة الاستخدام في هذا الكتاب: الفصول 1، 8، 9، 12، 13، 39، 42.

[R19] **Jiang et al. (2023). LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models.** https://arxiv.org/abs/2310.05736 - ضغط Prompt مع تقييم أثره؛ يختلف عن كبسولة الحالة المقترحة هنا. خريطة الاستخدام في هذا الكتاب: الفصول 14، 17، 28.

[R20] **Shu et al. (2024). Towards Effective GenAI Multi-Agent Collaboration: Design and Evaluation for Enterprise Applications.** https://arxiv.org/abs/2412.05449 - قياسات تنسيق وتوجيه في سيناريوهات مؤسسية منشورة. خريطة الاستخدام في هذا الكتاب: الفصول 2، 7، 19، 24، 25.

[R21] **Wang et al. (2024). BattleAgentBench: Evaluating Cooperation and Competition in Multi-Agent Systems.** https://arxiv.org/abs/2408.15971 - اختبار دقيق لقدرات التعاون والتنافس. خريطة الاستخدام في هذا الكتاب: الفصول 7، 17، 22، 24، 25.

[R22] **Bai et al. (2022). Constitutional AI: Harmlessness from AI Feedback. https://arxiv.org/abs/2212.08073** - مبادئ مكتوبة ونقد ومراجعة؛ ليست بديلاً عن حواجز نظامية. خريطة الاستخدام في هذا الكتاب: الفصول 3، 30، 31، 38.

[R23] **Ouyang et al. (2022). Training Language Models to Follow Instructions with Human Feedback.** https://arxiv.org/abs/2203.02155 - تعلم اتباع التعليمات من إشراف وتفضيلات بشرية. خريطة الاستخدام في هذا الكتاب: الفصول 3.

[R24] **Christiano et al. (2017). Deep Reinforcement Learning from Human Preferences. https://arxiv.org/abs/1706.03741** - تعلم هدف تقريبي من مقارنات بشرية. خريطة الاستخدام في هذا الكتاب: الفصول 3، 26، 30، 31، 38.

[R25] **Debenedetti et al. (2024). AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents.** https://arxiv.org/abs/2406.13352 - بيئة لتقييم حقن التعليمات ودفاعات الوكلاء. خريطة الاستخدام في هذا الكتاب: الفصول 21، 23، 29، 32، 34، 40، 41.

[R26] **Asai et al. (2023/2024). Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection.** https://arxiv.org/abs/2310.11511 - استرجاع ونقد متعلمان مع رموز انعكاس. خريطة الاستخدام في هذا الكتاب: الفصول 8، 15، 30.

[R27] **NIST (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0).** https://www.nist.gov/itl/ai-risk-management-framework - إطار رسمي لإدارة مخاطر الذكاء الاصطناعي. خريطة الاستخدام في هذا الكتاب: الفصول 3، 9، 23، 29، 31، 32، 33، 34، 38، 39، 41.

المراجع العلمية والتقنية - 3/3

Primary papers, benchmarks, and official risk frameworks

تغطي هذه الصفحة Benchmarks والتوجيه والذاكرة الوكيلة والمسوح الحديثة ومشهد المخاطر. المصادر الحديثة سريعة التقادم؛ لذلك تحمل كل دعوى تشغيلية تاريخ تحقق وFreshness policy.

قواعد الاستعمال

- اقرأ Abstract والمنهج والنتيجة والقيود؛ لا تستشهد بالعنوان وحده.
- ميز نتيجة الورقة عن تفسير هذا الكتاب وعن النظرية المساعفية المقترحة.
- تحقق من الإصدار والتاريخ والBenchmark والموارد قبل المقارنة.
- اربط الادعاء بالمقطع الداعم، وسجل ما لا يغطيه المصدر.

[R28] Manheim and Garrabrant (2018). Categorizing Variants of Goodhart's Law. https://arxiv.org/abs/1803.04585 - تصنيف لفشل المقاييس عندما تصبح أهدافًا أو بدائل مضللة. خريطة الاستخدام في هذا الكتاب: الفصول 6، 11.

[R29] Liu et al. (2023). AgentBench: Evaluating LLMs as Agents. https://arxiv.org/abs/2308.03688 - حزمة بيئات لتقييم السلوك الوكيلي. خريطة الاستخدام في هذا الكتاب: الفصول 4، 10، 20، 26، 35، 36، 42.

[R30] Mialon et al. (2023). GAIA: A Benchmark for General AI Assistants. https://arxiv.org/abs/2311.12983 - أسئلة متعددة الخطوات والأدوات لمساعدات عامة. خريطة الاستخدام في هذا الكتاب: الفصول 5، 10، 25، 26، 35، 36، 42.

[R31] NIST (2024). Artificial Intelligence Risk Management Framework: Generative AI Profile, AI 600-1. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf - ملف مخاطر وإجراءات لأنظمة الذكاء التوليدي. خريطة الاستخدام في هذا الكتاب: الفصول 3، 8، 15، 31، 32، 33، 38، 39.

[R32] Jimenez et al. (2023/2024). SWE-bench: Can Language Models Resolve Real-World GitHub Issues? https://arxiv.org/abs/2310.06770 - مهام إصلاح برمجيات مستخرجة من مستودعات حقيقية. خريطة الاستخدام في هذا الكتاب: الفصول 35.

[R33] Zhou et al. (2023). WebArena: A Realistic Web Environment for Building Autonomous Agents. https://arxiv.org/abs/2307.13854 - بيئة مواقع واقعية لتقييم الأفعال متعددة الخطوات. خريطة الاستخدام في هذا الكتاب: الفصول 35.

[R34] Xie et al. (2024). OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. https://arxiv.org/abs/2404.07972 - تقييم وكلاء حاسوب متعدد الوسائط في نظم تشغيل حقيقية. خريطة الاستخدام في هذا الكتاب: الفصول 35.

[R35] Ong et al. (2024). RouteLLM: Learning to Route LLMs with Preference Data. https://arxiv.org/abs/2406.18665 - تعلم توجيه الاستدعاءات بين نماذج وفق الجودة والكلفة. خريطة الاستخدام في هذا الكتاب: الفصول 2، 7، 13، 18، 19، 27، 28، 29، 40.

[R36] Kapoor et al. (2026). The 2025 AI Agent Index: Technical and Safety Features of Deployed Agentic Systems. https://arxiv.org/abs/2602.17753 - فهرس وصفي لأنظمة وكيلة؛ لا يمثل اعتمادًا أو إثبات AGI. خريطة الاستخدام في هذا الكتاب: الفصول 1، 8، 27، 30، 34، 35، 39، 42.

[R37] The Attack and Defense Landscape of Agentic AI (2026). https://arxiv.org/abs/2603.11088 - تصنيف حديث لهجمات الأنظمة الوكيلة ودفاعاتها. خريطة الاستخدام في هذا الكتاب: الفصول 15، 21، 23، 32، 33، 41.

[R38] Xu et al. (2025). A-Mem: Agentic Memory for LLM Agents. https://arxiv.org/abs/2502.12110 - ذاكرة ديناميكية وروابط وتطور للملاحظات في تقييمات منشورة. خريطة الاستخدام في هذا الكتاب: الفصول 12، 14، 16، 17، 36، 37.

[R39] Agentic AI: A Comprehensive Survey of Architectures, Applications, and Future Directions (2025). https://arxiv.org/abs/2510.25445 - مسح حديث؛ يستعمل كخريطة لا كمرجع أولي لكل نتيجة. خريطة الاستخدام في هذا الكتاب: الفصول 1، 14، 42.

[R40] Raza et al. (2025). TRiSM for Agentic AI: Trust, Risk, and Security Management in LLM-based Multi-Agent Systems. https://arxiv.org/abs/2506.04133 - مراجعة للحوكمة والتفسير والخصوصية والأمن في الأنظمة متعددة الوكلاء. خريطة الاستخدام في هذا الكتاب: الفصول 9، 15، 31، 32، 33، 34، 38، 42.

خاتمة تنفيذية وManifest معرفي

Moussaif Agentic AGI OS Research Book v1.0

اكتملت في هذا الإصدار الخريطة التأسيسية: نظرية أم، 42 نظرية فرعية، 72 أداة، نواة نشر تراكمي، بروتوكول قياس، وسجل مراجع. القيمة التالية ليست إضافة أسماء بلا نهاية، بل تنفيذ Prototype صغير يختبر Q-MCD وHyperRouting ومحكمة الهدم على مهام قابلة لإعادة الإنتاج.

Manifest المحتوى

Pages	250 fixed A4 pages
Language	Arabic RTL + English technical terms
Typeface	Amiri / Amiri Bold
Subtheories	42 = 7 chapters × 5 lenses
Tools	72 operational cards
References	40 sources
Companion	Full-length Markdown twin
Author	Zouhair Moussaif / زهير مساعف

الخطوة البحثية التالية

- تحويل Contracts إلى JSON Schemas.
- بناء 30 مهمة Baseline قابلة للتكرار.
- تنفيذ Q-MCD وMCD كـdeterministic scorers أوليين.
- تشغيل Ablation ومراجعة أوزان المعادلات.
- نشر النتائج الإيجابية والسلبية مع Traces منقحة.
- ترقية الأدوات الناجحة إلى HyperSkills versioned.

الكلمة الأخيرة

الذكاء الفتّاح ليس تضخم القدرة وحده، إنه فتح السؤال والآلية والدليل والقيمة مع بقاء الحدود مرئية وقابلة للحكم.

إنجاز وتأليف وتركيب: زهير مساعف - Zouhair Moussaif. الإصدار: الإصدار البحثي التأسيسي 13 - v1.0 يوليو 2026.