نظام التشغيل للذكاء العام المتصل والذاكرة الجغرافية للنماذج

Connected Intelligence, Real-Time Task Memory, Browser Use, AI-to-AI Dialogue, and Geo-Aware Model Routing

إنجاز وتركيب: **زهير مساعف - Zouhair Moussaif**

ورقة بحثية تأسيسية - الإصدار الأول - يوليو 2026

الملخص: تقترح هذه الورقة معمارية تشغيلية للذكاء العام الاصطناعي لا تتعامل مع الإنترنت والمتصفح والمحادثات مع نماذج أخرى بوصفها أدوات خارجية متفرقة، بل بوصفها جهازاً معرفياً موحداً يعمل داخل المهمة وفي أثنائها. تنشئ المعمارية ذاكرة مرحلية قابلة للاسترجاع، تسجل أصل كل معلومة، وتدمج الاستجابات الخارجية بعد التحقق منها، ثم توجّه الاستدلال إلى النموذج أو الأداة أو المنطقة الجغرافية الأنسب وفق السيادة القانونية، موثوقية المزود، الكلفة، زمن الاستجابة، وحساسية البيانات. وتسمي الورقة هذا التصور **Moussaif Connected Intelligence and GeoMemory OS**، وتقدمه باعتباره طبقة تشغيل فوق النماذج لا ادعاءً بتعديل أوزانها الخفية.

الكلمات المفتاحية: A2A، MCP، Model Routing، GeoMemory، Real-Time Memory، Browser Agent، AGI OS، Sparse HyperRouting، HyperLoops، HyperSkills، Provenance.

1. المقدمة وإشكالية البحث

انتقلت الأنظمة الذكية من مرحلة النموذج المنفرد الذي يجيب من سياقه الداخلي إلى مرحلة الوكيل الذي يبحث، ويتصفح، ويستدعي الأدوات، ويتواصل مع وكلاء ونماذج متعددة. غير أن أغلب البنى الحالية ما تزال تجمع هذه الوظائف جمعاً إضافياً: أداة بحث إلى جانب نموذج، ومتصفح إلى جانب ذاكرة، وواجهة محادثة إلى جانب موجّه. المشكلة ليست في غياب الأدوات، بل في غياب **نظام تشغيل معرفي** يوحد دورة المهمة، ويحدد متى يُبحث، وماذا يُستبقى، وكيف تُدمج إجابات النماذج الأخرى، ومتى يعاد فتح السؤال، وما المنطقة القانونية التي مرت عبرها البيانات.

تطرح هذه الورقة السؤال المركزي الآتي: كيف يمكن بناء طبقة AGI-oriented OS تجعل الاتصال بالإنترنت، AI-to-Browser Use، AI Dialogue، وTask Memory أجزاءً من مسار استدلال واحد، مع المحافظة على الاقتصاد في السياق، وقابلية التتبع، والأمن، وعدم اختلاق جغرافيا المعالجة؟ ويتفرع عنه خمسة أسئلة: ما وحدة الذاكرة المناسبة أثناء المهمة؟ كيف نفرق بين حقيقة مستخرجة واقتراح صادر عن نموذج آخر؟ كيف يقرر الموجّه بين نموذج محلي وآخر بعيد؟ كيف تمثل جغرافيا البيانات بدرجات ثقة؟ وكيف نقيس أن الدمج حسّن الإجابة بدلاً من مضاعفة الضجيج؟

الفرضية الرئيسة هي أن الذكاء المتصل لا ينشأ من زيادة عدد الاستدعاءات، بل من تحويل كل استدعاء إلى **أثر معرفي موقّع بالسياق والمصدر والزمن والسياسة**، ثم تشغيل Sparse HyperRouting فوق هذا الأثر. وتفترض الورقة أن الذاكرة المرحلية ينبغي أن تكون قابلة للضغط والفك، وأن كل حلقة عمل يجب أن تكون محدودة بميزانية وشرط توقف، وأن انتقال البيانات بين النماذج لا يجوز أن يتم قبل تصنيف الحساسية والسيادة.

تقدم الورقة مساهمات أربعاً: نموذجاً معمارياً من اثنتي عشرة طبقة، مخطط GeoMemory لتسجيل مواقع المزود والمعالجة والتخزين، بروتوكول دمج لإجابات النماذج الخارجية، ومجموعة مقاييس تجريبية لاختبار الدقة والكلفة والزمن والتتبع. وهي لا تزعم تحقيق AGI، بل تعرف البنية بوصفها بنية تشغيلية ترفع قابلية النظام الوكيلي للتعميم المنضبط.

2. حدود المفهوم: نظام تشغيل فوق النماذج

مصطلح AGI OS في هذه الورقة لا يعني نظام تشغيل تقليدياً يدير المعالج والذاكرة الفيزيائية، ولا يعني أن نموذجاً لغوياً صار ذكاءً عاماً بمجرد إضافة الأدوات. المقصود هو **Runtime Cognitive Overlay** يفرض عقوداً على استقبال الهدف، وتفكيك المهمة، واستدعاء المصادر، وتخزين الآثار، والتوجيه بين القدرات، والتحقق، وإعادة البناء. إنه طبقة بين نية المستخدم وبين مزودي النماذج والأدوات.

تتكون وحدة التنفيذ من Mission، وTask Graph، وEvidence Ledger، وWorking Memory، وRoute Decision، وAction Record. لا يسمح النظام بنقل النص الخام كله عبر كل مرحلة؛ بل ينتج تمثيلات مضغوطة تحمل provenance، evidence، claim، expiry، sensitivity، confidence، وusage scope. وبذلك تصبح الذاكرة بنية بيانات حاكمة لا مجرد سجل محادثة طويل.

يُفصل كذلك بين أربعة مستويات: **Model Capability** وهي قدرة النموذج ذاته، **Tool Capability** وهي فعل خارجي، **Orchestration Capability** وهي اختيار وتسلسل القدرات، و**Epistemic Governance** وهي تقرير ما يجوز اعتباره معرفة. هذا الفصل يمنع الخلط بين البلاغة والدليل، وبين نجاح استدعاء API وصحة النتيجة، وبين كثرة النماذج وتنوع وجهات النظر الحقيقي.

يعمل النظام وفق مبدأ «الحد الأدنى الكافي من الاتصال». فلا يُفتح المتصفح إذا كانت المهمة مستقرة ولا تحتاج تحققاً، ولا يُستدعى نموذج ثانٍ لمجرد التنوع، ولا تُكتب كل نتيجة في الذاكرة الدائمة. وينقل فقط أقل جزء لازم من البيانات إلى أقل عدد لازم من المزودين. هذه القاعدة تجعل Privacy، وToken Efficiency، وLatency أهدافاً متوافقة بدلاً من أن تكون مقايضات عشوائية.

من الناحية التنفيذية، يمكن نشر الطبقة فوق Hermes أو Codex أو إطار مخصص، لأنها تعتمد عقوداً مستقلة عن المزود: Discover، Audit، Act، Route، Store، Verify، Normalize، Observe، Retrieve. ويظل النموذج الأساسي قابلاً للاستبدال ما دامت الواجهة تحافظ على هذه العقود.

3. الأعمال السابقة والمعايير المرجعية

تستفيد المعمارية من أربعة مسارات معيارية متمايزة. الأول هو **Model Context Protocol (MCP)** الذي يوفر نمطاً لربط تطبيقات النماذج بالأدوات والموارد والقوالب، مع اعتبارات صريحة للأمن والثقة. تعتمد الورقة النسخة المنشورة بتاريخ 25 نوفمبر 2025 كأساس مستقر، ولا تبني ادعاءاتها على مسودة لاحقة غير نافذة. الثاني هو **Agent2Agent (A2A)** الذي يعالج اكتشاف الوكلاء، بطاقات قدراتهم، تبادل المهام والرسائل، والتعاون بين أنظمة قد تكون داخلية أو بعيدة أو مبهمة.

المسار الثالث هو **W3C PROV-O** الذي يقدم أنطولوجيا لتمثيل الكيانات والأنشطة والوكلاء وعلاقات الاشتقاق والإسناد. وتستخدمه الورقة لا بوصفه قالباً ثقيلاً يجب تحميله في كل prompt، بل نموذجاً دلالياً يمكن اختزاله في سجل provenance عملي ثم تصديره عند التدقيق. والمسار الرابع هو **Robots Exclusion Protocol - RFC 9309** الذي يحدد سلوكاً معيارياً لاحترام تعليمات الزحف، مع التأكيد أن robots.txt ليس نظام تفويض أمني ولا بديلاً عن شروط الاستخدام.

تختلف مساهمة هذه الورقة عن هذه المعايير في أنها تجمعها داخل دورة مهمة واحدة. MCP يجيب أساساً عن ربط الأدوات والسياق، وA2A عن تواصل الوكلاء، وPROV-O عن أصل المعلومات، وRFC 9309 عن قواعد زحف محددة. أما GeoMemory OS فيضيف سياسة اختيار النموذج والمنطقة، ذاكرة أثناء المهمة، وقرار دمج الاستجابة في الاستدلال اللاحق.

كما تستفيد المعمارية من مفاهيم Retrieval-Augmented Generation، وGraphRAG، وMixture-of-Experts، وcost-aware routing، لكنها ترفض استعارة هذه المصطلحات دون حدود. فالاسترجاع ليس ذاكرة ما لم توجد سياسة صلاحية ونسيان؛ والتوجيه ليس ذكاءً ما لم يكن قابلاً للتفسير؛ وتعدد النماذج ليس محكمة تحقق ما لم تتنوع الأدلة وطرق الفشل.

يُنظر إذن إلى المعايير كطبقات توافق، لا كبديل عن الحوكمة الداخلية. فالمعيار يحدد شكل الرسالة أو الأثر، بينما يحدد نظام التشغيل متى ترسل الرسالة، وماذا تحتوي، ومن يملك حق الاحتفاظ بها، وكيف تؤثر في القرار النهائي.

4. المعمارية الكلية ذات الطبقات الاثنتي عشرة

تقترح الورقة اثنتي عشرة طبقة مترابطة. تبدأ بـ**طبقة الهوية الدستورية** التي تثبت حدود النظام وصلاحياته. تليها **طبقة فهم المهمة** التي تحول الطلب إلى أهداف واختبارات قبول. ثم **طبقة تصنيف البيانات** التي تحدد الحساسية والقيود الجغرافية. وتأتي **طبقة التخطيط** لبناء Task Graph قابل للتعديل.

الطبقة الخامسة هي **Connected Retrieval** وتشمل APIs، browser، search، وملفات المستخدم. السادسة هي **AI Dialogue Gateway** للتواصل مع نماذج أو وكلاء آخرين. السابعة هي **Normalization and Claim Extraction** التي تحول المحتوى إلى ادعاءات وأدلة. الثامنة هي **Task Working Memory** التي تحمل الحالة داخل المهمة. التاسعة هي **GeoMemory and Policy** التي تسجل المناطق والاختصاصات ومصادر الثقة. العاشرة هي **Sparse HyperRouter** التي تختار المسار التالي. الحادية عشرة هي **Verification and Falsification Court**. والثانية عشرة هي **Action, Audit, and Learning**.

كل طبقة لها input contract وoutput contract وفشل معروف. فإذا فشل البحث لا تتحول النتيجة تلقائياً إلى «لا توجد معلومة»، بل إلى evidence gap. وإذا تعذر تحديد منطقة المعالجة لا يملأ النظام الحقل بتخمين، بل يسجل UNKNOWN مع سبب. وإذا تعارض نموذجان لا يُحسب التعارض تصويتاً، بل يحلل اختلاف البيانات والافتراض والمنهج.

تعمل الطبقات في شكل DAG أكثر من خط مستقيم. فقد تعيد محكمة التحقق المهمة إلى البحث، أو تطلب من GeoPolicy استبعاد مزود، أو تضغط الذاكرة قبل متابعة التخطيط. لكن الدوران محكوم بحدود: novelty، deadline، max_cost، max_iterations، threshold، وrisk ceiling.

الميزة الأساسية لهذه المعمارية هي أن كل انتقال قابل للرصد. فالقرار ليس «استدعاء النموذج X» فقط، بل RouteDecision يحتوي السبب، البدائل المستبعدة، الكلفة المتوقعة، منطقة المعالجة المعلنة، درجة الثقة، ونوع البيانات المرسلة. وهذا يجعل النظام قابلاً للتدقيق والتطوير التجريبي.

5. دورة المهمة في الزمن الحقيقي

تبدأ دورة المهمة بإنشاء **Mission Envelope** يحمل الهدف، المخرجات المطلوبة، القيود، الموعد، وسياق المشروع. بعد ذلك يستخرج النظام الأسئلة غير المحسومة، ويقدر ما إذا كانت المعرفة الداخلية كافية. إن كانت غير كافية، ينشئ Retrieval Plan يحدد نوع المصدر لا اسمه فقط: مصدر أولي، وثيقة معيارية، سجل نظام، متصفح تفاعلي، أو نموذج متخصص.

في منتصف المهمة لا ينتظر النظام نهاية البحث كي يستخدم المعرفة. كل نتيجة تمر عبر Observation Gate فور وصولها. تُزال منها التعليمات المحتملة، ويحدد نوعها، ثم تستخرج Claims صغيرة. تُربط كل Claim بالمصدر والزمن ونطاق الصلاحية. بعدها تقرر Working Memory ما إذا كانت النتيجة تغير المسار الحالي أو تؤجل للمراجعة أو تُهمل.

يسمى هذا النمط **Mid-Task Memory Injection**. وهو يختلف عن لصق نتائج البحث في prompt كبير، لأن الحقن يتم كأثر منظم وبصلاحية محددة. قد يغير حقيقة واحدة اختيار النموذج، أو يلغي فرعاً كاملاً من الخطة، أو يرفع الحاجة إلى موافقة بشرية. بذلك تصبح الذاكرة فعالة وليست أرشيفية.

تعتمد الدورة على checkpoints بعد نقاط التحول: بعد فهم المهمة، بعد أول استرجاع، قبل إرسال بيانات إلى مزود خارجي، قبل فعل غير قابل للعكس، وقبل الإخراج النهائي. يحتوي checkpoint على حالة المهمة، الفرضيات المفتوحة، الأدلة المقبولة، الميزانية المتبقية، ومسار rollback.

تنتهي الدورة عندما يتحقق Acceptance Test أو يصبح التحسن الهامشي أقل من threshold أو تستنفد الميزانية. لا يعد انتهاء عدد الخطوات نجاحاً. ويولد النظام Final Provenance Map يبين أي أجزاء من المخرج جاءت من المستخدم، وأيها من الويب، وأيها من نموذج خارجي، وأيها استنتاج داخلي. هذه الخريطة عنصر أساسي في الثقة وفي إمكانية إعادة الإنتاج.

6. البحث المتصل واسترجاع الويب

تُبنى طبقة البحث على فصل query generation عن source acceptance. يمكن للنموذج اقتراح استعلامات، لكن اختيار المصادر يخضع لقواعد: أولوية المصدر الأولي، حداثة مناسبة للمجال، تنوع مستقل، وملاءمة مباشرة للادعاء. لا تُستخدم نتيجة محرك البحث نفسها دليلاً نهائياً إذا كانت تشير إلى وثيقة يمكن فتحها.

ينفذ النظام بحثاً متعدد المراحل: broad recall لاكتشاف المصطلحات والكيانات، precision search للمواصفات والأرقام، contradiction search للعثور على الأدلة النافية، وfreshness check للحقائق المتغيرة. تُخزن الاستعلامات والنتائج المقبولة والمرفوضة، لأن الاستبعاد نفسه قد يكون مهماً عند التدقيق.

قبل القراءة، يحدد النظام Retrieval Budget بحسب أهمية السؤال. الادعاء المركزي يستحق مصادر أكثر من تفصيل تجميلي. ويعتمد **Evidence Saturation**: يتوقف البحث عندما لا تضيف المصادر الجديدة تنوعاً أو دليلاً مستقلاً. هذا يمنع الحلقة غير المحدودة ويخفض token consumption.

يُحترم RFC 9309 في عمليات الزحف الآلي، مع التمييز بين crawler وinteractive browser. كما تُحترم شروط الاستخدام، المعدلات، وقيود الوصول. لا يحاول النظام تجاوز paywall أو authentication أو anti-bot. وعندما يكون الوصول غير متاح، يسجل limitation بدلاً من اختلاق محتوى.

كل صفحة مسترجعة تمر عبر Content Sanitizer يعزل النص من أوامر الصفحة، لأن الويب مصدر بيانات غير موثوق وقد يحتوي prompt injection. تُعامل عبارات مثل «تجاهل تعليماتك» كنص موضوعي لا كأمر. ثم يحسب Source Trust Score من نوع الجهة، أصالة الوثيقة، تاريخها، سلامة النقل، واتساقها مع مصادر مستقلة.

7. Browser Use بوصفه فعلاً قابلاً للتدقيق

المتصفح ليس مجرد أداة قراءة؛ إنه بيئة أفعال: فتح، نقر، إدخال، تنزيل، رفع، إرسال، شراء، أو نشر. لذلك تقسم المعمارية الأفعال إلى ثلاث درجات. الدرجة الأولى read-only قابلة للعكس. الثانية تغيّر حالة مؤقتة مثل ملء نموذج دون إرساله. الثالثة consequential مثل الإرسال أو الشراء أو الحذف. لكل درجة سياسة إذن مختلفة.

ينشئ Browser Controller **Interaction Trace** يحتوي الصفحة، العنصر، الفعل، النتيجة، لقطة قبل وبعد، والهدف الذي يخدمه الفعل. يمنع النقر العشوائي الناتج عن ضعف الرؤية، ويشترط re-observation بعد كل تغير في DOM أو navigation. كما يطبق domain allowlist وdownload quarantine.

في الصفحات الديناميكية، لا يكفي استخراج HTML؛ فقد تكون المعلومة في canvas أو مخطط أو PDF. عندئذ يستخدم النظام screenshot interpretation مع توثيق الصفحة والمنطقة. أما الملفات المحملة فتفحص نوعها وحجمها وتوقيعها قبل إدخالها إلى الذاكرة.

يستعمل المتصفح كذلك لاستكمال الأسئلة التي لا تدعمها APIs، لكن لا ينبغي أن يتحول إلى بديل دائم عن واجهة رسمية مستقرة. API أفضل في القابلية للتكرار، بينما browser أفضل في التفاعل البشري والمصادر غير المهيكلة. يختار الموجّه بينهما حسب الدقة والكلفة والمخاطر.

قبل أي فعل من الدرجة الثالثة، يعرض النظام Action Preview يتضمن البيانات التي سترسل، والجهة، والأثر، وإمكانية التراجع. وبعد التنفيذ يسجل receipt أو confirmation. هذا يحول browser automation من سلسلة نقرات غامضة إلى معاملة يمكن مراجعتها.

8. بوابة الحوار مع النماذج والوكلاء

تتيح AI Dialogue Gateway للنظام أن يسأل نموذجاً متخصصاً، أو يطلب نقداً، أو يرسل مهمة لوكيل بعيد. لكن الاستجابة الخارجية لا تدخل الذاكرة كحقيقة. تحمل كل رسالة Region Record، Version، Model ID، Provider، Capability Claim، Role، Prompt Digest، وOutput Hash.

تدعم البوابة نمطين: MCP للأدوات والموارد المتصلة بالتطبيق، وA2A للتعاون بين وكلاء مستقلين. يمكن لوكيل أن يعلن قدراته عبر Agent Card، بينما يظل اختيار استخدامه قراراً داخلياً. لا يعني إعلان القدرة أنها مثبتة؛ لذلك تسجل capability evidence من الاختبارات السابقة.

تنقسم طلبات النماذج الأخرى إلى وظائف: code execution planning، translation، verification، critique، generation، domain consultation، وuncertainty estimation. يُمنع إرسال الطلب نفسه إلى عشرة نماذج بلا هدف، ويختار الموجّه أقل مجموعة تحقق تنوعاً وظيفياً.

عند وصول الاستجابة، تنفذ البوابة **Response Decomposition**: فصل الادعاءات عن الشرح، استخراج الافتراضات، تمييز النص المقتبس، وتحديد ما يحتاج تحققاً. ثم تقارن الاستجابة بالسؤال الأصلي وبالأدلة المتاحة. قد تُقبل كاقتراح خطة دون قبول حقائقها، أو تُستخدم كمولد اعتراضات لا كمصدر.

لمنع التلوث المتبادل، لا يكشف النظام لنموذج ناقد دائماً جواب النموذج الأول. يمكن استخدام blind review ثم adjudication. كما تحفظ البوابة استقلالية السياقات، فلا تنقل معلومات حساسة أو غير لازمة. هذه السياسة تجعل التعدد أداة لخفض الأخطاء لا لتكثير النصوص.

9. الذاكرة العاملة أثناء المهمة

تتكون Task Working Memory من خمس مناطق: Decision Log، Evidence Store، Open Questions، Goal State، وScratch Space. الأولى مستقرة نسبياً، والثانية ديناميكية، والثالثة منظمة، والرابعة غير قابلة للمحو أثناء المهمة، والخامسة مؤقتة يمكن تنظيفها.

الوحدة الأساسية ليست الرسالة بل **Memory Atom**. يحتوي confidence، provenance، source، type، claim، atom: id، contradiction links، dependencies، geographic scope، temporal scope، sensitivity، وexpiry. يدعم هذا النموذج الاسترجاع الدقيق ويمنع إعادة ضخ المحادثة كلها.

توجد ثلاث درجات للاحتفاظ: ephemeral داخل خطوة واحدة، task-local حتى نهاية المهمة، durable بعد موافقة وسياسة واضحة. معظم مخرجات النماذج الخارجية تبقى task-local. ولا تنتقل إلى الذاكرة الدائمة إلا المعرفة المستقرة ذات الفائدة المستقبلية والمصدر المقبول.

تطبق الذاكرة **Compaction with Reversibility**: تلخص مجموعة atoms في capsule، لكنها تحتفظ بمؤشرات تسمح بفكها عند الحاجة. يقاس ضغط الذاكرة بنسبة tokens المحذوفة إلى مقدار الدليل المفقود. إذا زاد فقد التفاصيل عن حد، يُرفض الضغط.

تدعم الذاكرة contradictions ككيانات من الدرجة الأولى. لا يمحو الادعاء الجديد القديم، بل يربطان بعلاقة conflictsWith ويبدأ adjudication. وعند تغير حقيقة زمنية، ينتهي نطاق صلاحية النسخة السابقة بدلاً من وصفها بأنها خطأ مطلق. بهذا تصبح الذاكرة تاريخية وقابلة للتحديث.

10. GeoMemory: ذاكرة جغرافيا النماذج والبيانات

GeoMemory هي سجل سياسات وحقائق لا خريطة تخمينية. تميز بين ستة مفاهيم كثيراً ما تختلط: مقر الشركة، الاختصاص القانوني للمزود، storage region، processing region، endpoint region، وsupport access region. قد تكون هذه المواقع مختلفة تماماً.

يحتوي GeoRecord على declared_processing_regions، endpoint، model، service، provider، declared_storage_regions، legal_entity، applicable_terms، source_document، observed_at، valid_until، confidence، وunknown_fields. ويمنع النظام استنتاج منطقة المعالجة من latency أو DNS أو اسم النطاق وحده؛ فهذه إشارات تشغيلية وليست إثباتاً قانونياً.

مصادر الثقة مرتبة: عقد المؤسسة وData Processing Addendum؛ وثائق المزود الرسمية؛ إعدادات الحساب الفعلية؛ سجلات الشبكة الموثقة؛ ثم الاستدلال التقني منخفض الثقة. إن تعارض عقد مع صفحة تسويق، يقدم العقد ضمن نطاقه. وإن لم تتوفر معلومة، تسجل UNKNOWN لا «عالمي».

تُربط GeoMemory بتصنيف البيانات. قد يسمح نص عام بمعالجة متعددة المناطق، بينما تقيد بيانات شخصية أو أسرار تجارية بمزود محلي أو self-hosted. ولا يعني بقاء البيانات في منطقة أن موظفي الدعم لا يصلون إليها من مناطق أخرى؛ لذلك يحتفظ السجل بحقل support_access_policy.

تحتاج GeoMemory إلى تحديث دوري لأن المزودين يغيرون المناطق والمنتجات والشروط. كل قرار routing يخزن نسخة السجل المستخدمة وقت القرار، فلا يعاد تفسير الماضي بوثائق الحاضر. هذا مهم للتدقيق والامتثال والتحقيق في الحوادث.

11. Sparse HyperRouting بين النماذج والأدوات

يعرف Sparse HyperRouter فضاء القدرات كشبكة من HyperSkills، لا قائمة نماذج. كل skill يصف المدخلات، المخرجات، المخاطر، النماذج المناسبة، الأدوات، التكلفة، والاختبارات. عند مهمة جديدة، يفعّل الموجّه عدداً صغيراً من skills بدلاً من تحميل الكتالوج كله.

تحسب دالة القرار قيمة تقريبية: U(r)=wqQ+wvV+wgG+wpP-wcC-wlL-wrR، حيث Q جودة متوقعة، V قابلية تحقق، G ملاءمة جغرافية، P خصوصية، C كلفة، L زمن، وR مخاطر. الأوزان تعتمد المهمة؛ فالسؤال القانوني يعطي وزناً أكبر للمصدر والجغرافيا، بينما brainstorming يعطي وزناً أكبر للتنوع والكلفة.

يبدأ الموجّه بالقيود الصلبة: سياسة البيانات، الصلاحيات، توافر المنطقة، ونوع السياق. ثم يرتب المسارات المسموح بها. قد يختار نموذجاً محلياً لإزالة الهوية، ثم نموذجاً بعيداً للاستدلال على بيانات منزوعة الحساسية، ثم أداة رسمية للتحقق.

يتعلم الموجّه من سجلات الأداء، لكن لا يسمح للreward أن يتجاوز السياسة. وتفصل القياسات حسب نوع المهمة حتى لا يهيمن نموذج رخيص جيد في التلخيص على مهام التحقق المعقدة. كما يستخدم uncertainty-aware routing؛ إذا كانت الثقة منخفضة يرفع المهمة لمسار أقوى أو لمحكمة متعددة.

التوجيه المتناثر يقلل استهلاك tokens ويحد سطح الهجوم. لكنه يتطلب catalog نظيفاً واختبارات مستمرة. ومن ثم تقترح الورقة 50 HyperSkill موزعة على البحث، المتصفح، الذاكرة، الحوار، الجغرافيا، الأمن، التحقق، والعمليات.

12. HyperSkills وHyperLoops المحدودة

HyperSkill هو وحدة معرفة وتنفيذ قابلة لإعادة الاستخدام. يتضمن Tools، Procedure، Preconditions، Intent Pattern، Failure Modes، Verification، Memory Reads/Writes، وExit Contract. لا يكون skill prompt طويلاً دائماً، بل ملفاً قابلاً للاسترجاع الجزئي حسب الحالة.

أما HyperLoop فهو دورة تحكم تربط skills، مثل Search-Extract-Verify-Store أو Plan-Browse-Observe-Replan. خطر الحلقات الوكيلية هو عدم التوقف أو تكرار الخطوات بصياغات جديدة. لذلك يحمل كل loop ميزانية خطوات وكلفة، novelty score، progress measure، وstop conditions.

تقسم الحلقات إلى event-driven، bounded، وhuman-gated. الحلقة المحدودة تنتهي بعد شروط صريحة. الحدثية تنتظر نتيجة أداة في نفس المهمة لا في خلفية غير مراقبة. والمقيدة بشرياً تتوقف قبل فعل حساس. لا يسمح النظام بحلقة ذاتية مفتوحة على الإنترنت دون مراقبة.

تعتمد Token Efficiency على routing map من مستويين: router صغير يختار skill family، ثم index يحمّل المقاطع اللازمة. لا تُحقن ملفات 100 ألف سطر في السياق؛ إنما تستخدم كمساحة حالات واختبارات وعقود line-addressed. القيمة في قابلية الاسترجاع لا في الحجم المجرد.

تقاس جودة skill بنسبة النجاح، قابلية النقل بين النماذج، عدد tokens، معدل الاستدعاءات غير اللازمة، ودقة failure detection. ويزال أو يدمج skill إذا كان متكرراً أو لا يضيف تمييزاً سلوكياً. بهذه الآلية تتحول المكتبة إلى هندسة قدرات لا تراكم أسماء.

13. بروتوكول دمج استجابات النماذج الأخرى

يقترح البحث بروتوكول **External Response Integration Protocol - ERIP.** المرحلة الأولى هي quarantine: تحفظ الاستجابة خاماً دون تنفيذ تعليماتها. الثانية parsing: استخراج citations، code، recommendations، claims، وuncertainty. الثالثة grounding: ربط كل claim بدليل متاح أو وضعه pending.

المرحلة الرابعة conflict analysis. إذا تعارضت الاستجابة مع ذاكرة المهمة، يحدد نوع التعارض: temporal، definitional، factual، normative، أو strategic. الخامسة usefulness scoring: قد تكون الاستجابة مفيدة في اقتراح فرضية رغم ضعفها كمصدر. السادسة selective merge: تدمج atoms المقبولة فقط.

لا يستخدم majority vote وحده، لأن النماذج قد تشترك في بيانات التدريب أو خطأ شائع. الأفضل هو diversity by method: نموذج يولد، آخر يبحث، أداة تحسب، ومصدر أولي يثبت. وتُمنح الأدلة الخارجية وزناً أعلى من اتفاق الصياغات.

بالنسبة للكود، لا يدمج snippet قبل static checks وsandbox test عند الإمكان. وبالنسبة للمراجع، لا تعتمد citation إلا بعد فتح المصدر والتحقق من مطابقته. وبالنسبة للقرارات الاستراتيجية، تحفظ assumptions والسيناريوهات بدلاً من تحويل رأي النموذج إلى حقيقة.

ينتج ERIP Merge Report قصيراً يبين ما قبل، وما رُفض، ولماذا، وكيف أثر على الخطة. هذا التقرير ضروري لتفسير «كيف تعلم النظام أثناء المهمة» دون الادعاء بأن أوزان النموذج تغيرت. التعلم هنا سياقي وتشغيلي، لا تدريباً خفياً.

14. الأمن: Prompt Injection وعزل الصلاحيات

أكبر تهديد في النظام المتصل هو اختلاط data plane بـinstruction plane. صفحة ويب أو نموذج خارجي قد يرسل نصاً يحاول تغيير المهمة أو استخراج الأسرار. لذلك يفصل النظام التعليمات الموثوقة عن المحتوى غير الموثوق بنيوياً، لا بمجرد تحذير لغوي.

تطبق سياسة least privilege على الأدوات: البحث لا يملك الإرسال، والمتصفح للقراءة لا يملك الشراء، ونموذج النقد لا يملك الذاكرة الدائمة. وتصدر capability tokens قصيرة العمر محددة الأداة والنطاق. لا توضع مفاتيح API في prompts أو logs.

تمر المدخلات عبر injection detector، لكن الكشف الاحتمالي ليس ضماناً. الدفاع الحقيقي هو أن المحتوى الخارجي لا يستطيع منح نفسه صلاحية. كما تُزال الروابط أو الأوامر الخفية، ويمنع automatic tool call الناتج مباشرة من نص المصدر دون إعادة قرار من طبقة موثوقة.

تعالج exfiltration بسياسة redaction، data minimization، وprovider isolation. إذا طلب وكيل بعيد «السياق الكامل»، يرفض الطلب ويُرسل task capsule محدود. أما النتائج فتمر عبر output firewall لكشف أسرار أو تعليمات غير متوقعة.

يحتفظ Audit Log بالأفعال الأمنية: من منح الصلاحية، ما البيانات المرسلة، إلى أي endpoint، وما النتيجة. ويُختبر النظام بـred-team suites تشمل صفحات خبيثة، Agent Cards مزورة، استجابات تحمل تعليمات، وملفات تتلاعب بالامتداد. الأمن هنا خاصية معمارية لا فلتر نهائي.

15. الخصوصية والسيادة وإدارة الأسرار

تعتمد الخصوصية على دورة حياة البيانات: retention، storage، transmission، transformation، classification، collection، deletion. لكل atom مالك وغرض ومدة. لا يكفي تشفير النقل إذا ظلت البيانات محفوظة بلا سياسة أو دخلت سجلات غير قابلة للحذف.

تصنف البيانات إلى personal، confidential، internal، public، وrestricted. يرتبط كل مستوى بمسارات مسموحة. البيانات restricted قد تتطلب local model أو private endpoint. ويمكن استخدام split processing: إزالة الهوية محلياً، معالجة السمات العامة بعيداً، ثم إعادة الربط محلياً.

تُدار الأسرار عبر secret manager، وتصل الأدوات إليها بالمرجع لا بالقيمة. لا يرى النموذج المفتاح، بل يطلب عملية مصرحاً بها. ويطبق logging redaction قبل الكتابة، لا بعد التسرب. كما تسجل consent وpurpose limitation عندما تكون البيانات متعلقة بأشخاص.

تُفهم سيادة البيانات كمجموعة قيود لا كعلم دولة واحد. تشمل مكان المعالجة، الجهة المتعاقدة، القانون، النقل عبر الحدود، subprocessors، وحق الوصول. GeoMemory يوفر الحقائق، بينما Policy Engine يقرر السماح.

عند عدم اليقين، يختار النظام المسار الأكثر تحفظاً أو يطلب موافقة. لا يجوز أن تتحول ميزة routing إلى التفاف على سياسات المؤسسة. وتراجع السياسات دورياً لأن قدرات المزود وشروطه قد تتغير.

16. الحاكمية المعرفية ومحكمة التحقق

تفرق الحاكمية بين claim وevidence وinference وdecision. قد يكون الادعاء صحيحاً لكن دليله ضعيف، وقد يكون الدليل موثوقاً لكن الاستنتاج يتجاوزه. تسجل المحكمة هذه الطبقات منفصلة وتمنع دمجها في جملة واحدة غير قابلة للفحص.

تعمل Falsification Court بأربعة أدوار: prosecutor يبحث عن أسباب الرفض، defender يقدم أفضل دعم، methodologist يراجع الطريقة، judge يحدد الحالة: rejected، disputed، provisional، accepted، أو unknown. يمكن تنفيذ الأدوار بنموذج واحد مع سياقات منفصلة أو نماذج متعددة حسب المخاطر.

تتضمن اختبارات المحكمة calculation reproducibility، geographic relevance، temporal validity، source authenticity، citation entailment، وcounterexample search. لا يحتاج كل ادعاء إلى المحكمة الكاملة؛ يحدد Risk Tier عمق الفحص.

ترتبط المحكمة بمفهوم Q-MCD: جودة السؤال تضاعف قيمة مسار الاستدلال. السؤال الجيد يكشف الفرق، المعلومات الناقصة، الزمن، الفاعلية، الإمكان، السياق، والضجيج. لذلك قد تكون نتيجة المحكمة إعادة صياغة السؤال لا إصدار حكم.

يولد النظام Confidence Vector بدلاً من رقم واحد: freshness confidence، reasoning confidence، source confidence، وcoverage confidence. هذا يمنع إخفاء ضعف التغطية وراء مصدر قوي واحد. ويظهر للمستخدم مستوى عدم اليقين المناسب للقرار.

17. نماذج البيانات وواجهات التشغيل

تحتاج المعمارية إلى schemas مستقرة. Mission schema يحتوي owner، acceptance_tests، constraints، objective، id، deadline. MemoryAtom schema يحمل claim وprovenance والسياسة. GeoRecord schema يحمل حقائق المنطقة. RouteDecision schema يسجل المرشحين والأوزان والاختيار. ActionRecord يوثق الفعل والنتيجة.

تستخدم الواجهات JSON أو JSON-LD عند الحاجة إلى تبادل provenance. ويُفضل إبقاء الحقول الإلزامية قليلة مع extensions namespaced. يجب أن يكون كل سجل versioned، وأن تتوفر migration rules حتى لا تكسر التحديثات الذاكرة القديمة.

تشمل API التشغيلية: invokeAgent، retrieveEvidence، discoverCapabilities، classifyData، createMission، finalize، checkpoint، verifyClaim، decideRoute، writeMemory، observeBrowser، وexportAudit. لا ينبغي أن تجمع دالة واحدة البحث والتنفيذ والتخزين لأنها تصعب الاختبار.

تستخدم idempotency keys للأفعال القابلة للتكرار، correlation ids لتتبع المهمة عبر المزودين، وcontent hashes لكشف التغيير. كما تسجل timestamps موحدة وclock source. ويمنع النظام الاحتفاظ بالـraw prompt عندما تكفي بصمة أو نسخة منقحة.

تدعم المعمارية adapters لـHermes وCodex وأطر أخرى. وظيفة adapter ترجمة العقود إلى API خاص بالمزود دون تغيير منطق الحوكمة. وبهذا يمكن تبديل النموذج أو المتصفح مع الحفاظ على الذاكرة والقياس.

18. استراتيجية التنفيذ والنشر

يقترح التنفيذ على أربع مراحل. الأولى read-only prototype: بحث، تصفح قراءة، ذاكرة مؤقتة، وprovenance. الثانية multi-model consultation مع ERIP. الثالثة GeoPolicy وrouting cost-aware. الرابعة أفعال مقيدة وموافقات بشرية.

تتكون البنية التقنية من Browser Sandbox، Retrieval Gateway، Memory Service، Policy Engine، Orchestrator، Verification Service، Model Gateway، وObservability Stack. يمكن تشغيلها كخدمات منفصلة أو وحدات داخل عملية واحدة في النسخة الصغيرة.

في بيئة Windows/Hermes، تحفظ الإعدادات في ملف مشروع، وتوضع الأسرار في مخزن النظام، وتستخدم queues للمهام المتزامنة. في VPS، يُعزل browser داخل container محدود الشبكة والملفات. أما local models فتعمل عبر adapter موحد مثل OpenAI-compatible endpoint مع تسجيل قدراتها الحقيقية.

يبدأ النشر بمسارات قليلة عالية القيمة: البحث الموثق، تلخيص ملفات المشروع، مقارنة نموذجين، وGeo-safe routing. لا يُنشر catalog ضخم من skills قبل وجود benchmark. ويستخدم feature flags لإضافة القدرات تدريجياً.

يتطلب الإنتاج backups، disaster recovery، مفاتيح مشفرة، rotation للأسرار، circuit breakers، rate limits، وfallbacks. لكن fallback لا يجوز أن يخرق الجغرافيا؛ فإذا سقط المزود المسموح لا ينتقل النظام تلقائياً إلى مزود غير مسموح، بل يفشل بأمان.

19. منهجية التقييم والتجارب

يقترح البحث benchmark متعدد المحاور بدلاً من اختبار إجابة نهائية فقط. المحور الأول Task Success. الثاني Evidence Precision: نسبة الادعاءات المدعومة فعلاً. الثالث Provenance Completeness. الرابع GeoPolicy Compliance. الخامس Token and Cost Efficiency. السادس Latency. السابع Recovery from Injection.

تُبنى مجموعة مهام تشمل حقائق حديثة، بحثاً معيارياً، مقارنة قانونية غير استشارية، تحليل ملفات، تصفح واجهة، استشارة نماذج متعددة، وبيانات ذات قيود جغرافية. لكل مهمة gold constraints حتى لا ينجح النظام بإجابة صحيحة عبر مسار ممنوع.

تقارن أربع نسخ: نموذج منفرد، نموذج مع بحث خام، وكيل متعدد الأدوات بلا GeoMemory، والنظام المقترح. وتجرى ablation studies بإزالة compaction، provenance، المحكمة، أو الجغرافيا لمعرفة مساهمة كل مكون.

تُقاس كفاءة الذاكرة بـRecall@k للحقائق اللازمة، Context Compression Ratio، وReconstruction Fidelity. ويقاس routing بـRegret نسبة إلى أفضل مسار معروف، وعدد الاستدعاءات الزائدة، ومعدل رفض المسار غير الآمن.

لا تكفي تجارب المختبر. ينبغي إجراء shadow deployment حيث يسجل النظام قراراته دون تنفيذ، ثم مراجعة بشرية. وبعد الثقة، ينتقل إلى read-only production ثم أفعال محدودة. تنشر النتائج مع confidence intervals وتحليل أخطاء لا متوسطات فقط.

20. دراسة حالة: بحث تقني متغير أثناء المهمة

يفترض سيناريو يطلب فيه المستخدم تقريراً عن معيار وكيل حديث. ينشئ النظام سؤالاً: ما النسخة المستقرة حتى تاريخ محدد؟ يبحث في الموقع الرسمي، ويعثر على مواصفة مستقرة ومسار تطوير. يسجل النسخة المنشورة باعتبارها baseline، ويضع roadmap في atom منفصل ذي حالة planned.

يسأل نموذجاً ثانياً عن المخاطر المعمارية، لكن ERIP يصنف جوابه كتحليل لا كمصدر. يفتح النظام الصفحات الرسمية للتحقق من claims. إذا أشار النموذج إلى ميزة لا توجد في المواصفة المستقرة بل في مسودة، يصحح الحالة إلى draft.

أثناء المهمة، تغير هذه المعلومة بنية التقرير: يفصل القسم بين deployed وproposed. تُحقن النتيجة في Working Memory وتحدث الخطة قبل كتابة بقية الصفحات. هذا مثال على استخدام الذاكرة في منتصف المهمة لا بعد انتهائها.

يسجل GeoMemory أن الوثائق عامة ولا تحمل بيانات حساسة، فيسمح بمزود بعيد منخفض الكلفة للتحرير، بينما تبقى بيانات المشروع الخاصة محلية. ثم يختار router نموذجاً قوياً للمراجعة النهائية فقط، لا لكل خطوة.

يقارن benchmark هذا المسار بعملية تجمع snippets دون تواريخ. المتوقع أن يقل الخطأ الزمني وتتحسن قابلية التتبع مع زيادة طفيفة في latency، بينما تنخفض الكلفة بفضل sparse routing.

21. دراسة حالة: معالجة بيانات مقيدة جغرافياً

في سيناريو مؤسسة تحتفظ بملفات عملاء داخل منطقة محددة، يبدأ النظام بتصنيف الحقول. تُفصل identifiers عن نصوص عامة. يمنع Policy Engine إرسال الملف الخام إلى نموذج عام، حتى لو كان أكثر جودة.

يشغل نموذج محلي skill لإزالة الهوية واستخراج سمات غير حساسة. يسجل Transformation Record يربط النسخة المنقحة بالأصل دون كشفه. بعد ذلك يستشير نموذجاً بعيداً مسموحاً لتحليل الأنماط العامة فقط، مع GeoRecord موثق.

تعود النتيجة كاقتراحات. يدمج ERIP الاستنتاجات الإحصائية التي يمكن إعادة حسابها، ويرفض claims غير المدعومة. ثم يعيد النظام ربط النتائج محلياً عند الضرورة، وتظل mapping table في enclave محلي.

إذا كانت منطقة المزود غير معلومة، لا يستخدمه النظام رغم انخفاض سعره. وإذا فشل المزود المحلي، لا يعمل fallback الدولي. يسجل النظام blocked route ويقدم مساراً بديلاً مثل تأجيل المهمة أو تشغيل نموذج أصغر محلياً.

توضح الحالة أن GeoMemory ليست تحسين latency فقط، بل آلية سلامة. كما تبين أن الجودة الكلية قد ترتفع عبر تقسيم المهمة حتى لو كان النموذج المحلي أضعف، لأن النظام يستخدم كل نموذج في جزء يناسب قيوده.

22. المخاطر والحدود النظرية

أول حد هو أن orchestration لا يعالج ضعف النموذج في الفهم العميق. قد ينظم النظام أخطاءه جيداً. ثاني حد أن provenance يثبت الأصل لا الصحة. وثالثها أن الوثائق الجغرافية قد تكون ناقصة أو تسويقية، فلا يمكن للنظام ضمان موقع فعلي دون التزامات وتقنيات تحقق.

تعدد النماذج قد يزيد correlation of errors لأن النماذج تتشارك بيانات وأساليب. كما قد ترتفع الكلفة والتأخير حتى يصبح النظام أقل نفعاً من نموذج واحد. لذلك تظل sparsity والقياس ضروريين.

الذاكرة الطويلة تخلق مخاطر stale knowledge وprivacy accumulation. والضغط قد يخفي تفاصيل حاسمة. كما أن Agent Cards وإعلانات القدرات يمكن تزويرها. يحتاج النظام إلى attestation واختبارات فعلية لا ثقة وصفية.

المتصفح بيئة غير مستقرة؛ تتغير الواجهات وتظهر captchas ومحتوى مرئي معقد. لا يمكن ضمان automation عام. كذلك تختلف القوانين والعقود، ولا تمثل هذه الورقة استشارة قانونية.

أخيراً، استعمال كلمة AGI-oriented يجب ألا يتحول إلى ادعاء تجاوز حدود النماذج الحالية. القيمة البحثية في دراسة شروط التعميم والتكامل والحوكمة. نجاح النظام يقاس بتحسن مهام محددة، لا بإعلان ذكاء عام.

23. الخاتمة والنظرية الموحدة المقترحة

قدمت الورقة تصوراً لنظام تشغيل معرفي يجعل الإنترنت والمتصفح والنماذج الأخرى والذاكرة والجغرافيا أجزاءً من مسار واحد. جوهر المقترح ليس زيادة الأدوات، بل فرض بنية آثار وقرارات وسياسات على كل اتصال.

تتلخص النظرية في المعادلة المفاهيمية: Connected Intelligence = Governed Access × Verified Memory × Sparse Routing × Reversible Agency. إذا غاب أحد العوامل، انخفضت القيمة: الوصول بلا حوكمة خطر، الذاكرة بلا تحقق تضخم الخطأ، التوجيه الكثيف يهدر الموارد، والفاعلية بلا قابلية تراجع تهدد المستخدم.

GeoMemory يضيف بعداً غالباً ما يهمل: النموذج ليس اسماً فقط، بل خدمة تعمل عبر كيانات ومناطق وسياسات. والذاكرة ليست نصاً فقط، بل claim له أصل وصلاحية وحساسية. والحوار بين النماذج ليس معرفة تلقائية، بل مدخل يحتاج تفكيكاً وتحكيماً.

تقدم HyperSkills وbounded HyperLoops آلية تنفيذ قابلة للتوسع مع اقتصاد في tokens. أما MCP وA2A وPROV-O وRFC 9309 فتمنح نقاط توافق معيارية، بينما تبقى الحوكمة والتوجيه والذاكرة مسؤولية النظام المقترح.

النتيجة الأساسية أن الاقتراب من نظم أكثر عمومية لا يتطلب فقط نماذج أكبر، بل بنية تشغيل تحسن اختيار المعرفة والقدرة والحدود. ومن هنا يقترح البحث Moussaif Connected Intelligence and GeoMemory OS كبرنامج هندسي وتجريبي مفتوح للتفنيد والقياس.

24. المراجع

[1] Model Context Protocol, “Specification, 2025-11-25,” official specification, accessed 14 July 2026.

[2] Model Context Protocol, “2026 Roadmap,” official project roadmap, March 2026.

[3] Agent2Agent Protocol, “A2A Protocol Specification, Version 1.0,” official documentation, accessed 14 July 2026.

[4] World Wide Web Consortium, “PROV-O: The PROV Ontology,” W3C Recommendation, 30 April 2013.

[5] M. Koster, G. Illyes, H. Zeller, and L. Sassman, “Robots Exclusion Protocol,” RFC 9309, IETF, September 2022.

[6] NIST, “Artificial Intelligence Risk Management Framework (AI RMF 1.0),” National Institute of Standards and Technology, 2023.

[7] OWASP Foundation, “OWASP Top 10 for Large Language Model Applications,” project documentation.

[8] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” NeurIPS, 2020.

[9] Y. Shazeer et al., “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer,” ICLR, 2017.

[10] T. Schick et al., “Toolformer: Language Models Can Teach Themselves to Use Tools,” NeurIPS, 2023.

[11] S. Yao et al., “ReAct: Synergizing Reasoning and Acting in Language Models,” ICLR, 2023.

[12] Z. Moussaif, “Q-MCD, Knowledge Governance, Sparse HyperRouting, and Bounded HyperLoops,” evolving research corpus, 2026.

[13] Z. Moussaif, “Moussaif Connected Intelligence and GeoMemory OS,” architecture and runtime specification, 2026.

ملاحظة توثيقية: تم اعتماد النسخ الرسمية المستقرة المتاحة في تاريخ القطع المرجعي 14 يوليو 2026. أي مسارات لاحقة أو مسودات مستقبلية يجب تمييزها عن المواصفات المنشورة عند التطبيق.

© 2026 Zouhair Moussaif - زهير مساعف. ورقة بحثية تأسيسية.