

الجمهورية الشعبية الديمقراطية الجزائرية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

المدرسة العليا للإعلام الآلي 08 ماي 1945 بسيدي بلعباس
Higher School of Computer Science
-08 May 1945- Sidi Bel Abbès



ENGINEERING THESIS

To obtain the diploma of **Engineer**
Field: **Computer Science**
Specialty: **Artificial Intelligence and Data Science (IASD)**

Theme

**Reliable Integration of Large Language Models into
Multi-Tenant Business Systems using Bounded-Authority
Architectures**

Presented by:
BENAHMED Firdaws Sotra

Submission Date: **23/09/2026**
In front of the jury composed of:

Dr. SAIDI Mohammed Elfatih
Dr. BOUSMAHA Rabab
Dr. ELHANNANI Souad

President
Supervisor
Examiner

Academic Year: 2025/2026

ABSTRACT

Every business platform faces the same paradox at adoption time: it is useless until the customer’s data is inside it, yet the customer’s data exists only in the messy artefacts of their previous way of working — multilingual spreadsheets, printed PDF catalogues, photographed price lists. This thesis presents UniFlow, a multi-tenant business management platform for the retail and aviation industries, extended with four cooperating AI surfaces that together cover the life cycle of a tenant’s data: an *Onboarding Engine* that creates the business, an *AI Auditor* that critiques it, a *Simulation Lab* that brings it to life, and an *MCP server* through which the customer’s own assistant serves it.

The unifying engineering contribution is a single design principle applied three times: *a large language model is used exactly where language judgment is required, and deterministic code is used everywhere correctness must be guaranteed*. In the Onboarding Engine, the model interprets what a column means while Python validates, deduplicates, generates identifiers, and inserts — no unvalidated model output ever reaches the database, and a mandatory human review screen closes the loop. In the Auditor, thirteen deterministic checks establish a *guaranteed floor* of facts that the model may explain and extend but can never remove or downgrade, so the report’s factual content survives even a total model outage. In the Simulation Lab, the model is called once to author two hundred customer personas, and a seeded, fully reproducible Python engine then runs a month of their behaviour through the platform’s real endpoints, producing emergent purchasing patterns that no rule in the code prescribes. The platform also exposes each tenant’s public catalogue and guest booking flow to external AI assistants through a Model Context Protocol (MCP) server whose fourteen tools are scoped to a brand by an explicit company name, requiring no credential and no per-brand configuration.

Each feature is offered on two interchangeable LLM backends — a hosted frontier model and a local model served by Ollama — and the measured gap between them quantifies what a small business gives up by keeping its commercial data on its own hardware. Results show that the layered architectures deliver the reliability of rule-based systems with the flexibility of language models, and that the boundary between the two is where the engineering value lies.

Keywords — Large language models, multi-tenant SaaS, data onboarding, information extraction, data quality auditing, human-in-the-loop, agent-based simulation, synthetic data, local inference, Model Context Protocol, FastAPI, structured output

الملخص

تواجه كل منصة لإدارة الأعمال المفارقة نفسها عند اعتمادها: فهي عديمة الفائدة ما لم تكن بيانات العميل داخلها، في حين لا توجد هذه البيانات إلا في مخلفات طريقة عمله السابقة --- جداول بيانات متعددة اللغات، وكالوجات PDF مطبوعة، وقوائم أسعار مصوّرة. تقدم هذه المذكرة منصة **UniFlow**، وهي منصة متعددة المستأجرين لإدارة الأعمال في قطاعي تجارة التجزئة والطيران، مزوّدة بأربع واجهات ذكاء اصطناعي متعاونة تغطي معاً دورة حياة بيانات المستأجر: محرك إدماج يُنشئ العمل التجاري، ومدقق ذكي ينتقده، ومختبر محاكاة يبعث فيه الحياة، وخادم **MCP** يخدمه من خلاله مساعد العميل نفسه.

تتمثل المساهمة الهندسية الموحدة في مبدأ تصميمي واحد يُطبّق ثلاث مرات: يُستخدم النموذج اللغوي الكبير تحديداً حيث يلزم حكم لغوي، ويُستخدم الكود الحتمي في كل موضع يجب فيه ضمان الصواب. ففي محرك الإدماج يفسر النموذج معنى العمود بينما تتولى Python التحقق وإزالة التكرار وتوليد المعرّفات والإدراج --- فلا تصل أي مخرجات غير محققة من النموذج إلى قاعدة البيانات، وتغلق شاشة مراجعة بشرية إلزامية الحلقة. وفي المدقق، تُرسي ثلاث عشرة عملية تحقق حتمية أرضية مضمونة من الحقائق يمكن للنموذج أن يشرحها ويوسعها لكنه لا يستطيع إزالتها أو تخفيض خطورتها، بحيث يبقى المحتوى الواقعي للتقرير قائماً حتى في حال تعطل النموذج كلياً. وفي مختبر المحاكاة، يُستدعى النموذج مرة واحدة لكافة متني شخصية عميل، ثم يشغل محرك Python حتمي وقابل لإعادة الإنتاج شهراً كاملاً من سلوكها عبر نقاط النهاية الحقيقية للمنصة، منتجاً أنماط شراء ناشئة لا تفرضها أي قاعدة في الكود. كما تعرض المنصة الكالوج العام لكل مستأجر ومسار حجز الضيوف لمساعدات الذكاء الاصطناعي الخارجية عبر خادم يعتمد بروتوكول سياق النموذج، (**MCP**) تُقيّد أدواته الأربع عشرة بعلامة تجارية واحدة عبر اسم شركة صريح، دون الحاجة إلى أي بيانات اعتماد أو إعداد خاص بكل علامة.

تُتاح كل ميزة على محركين قابلين للتبادل للنماذج اللغوية --- نموذج رائد مستضاف ونموذج محلي يُشغّل عبر **Ollama** --- ويقيس الفارق بينهما ما تتنازل عنه المؤسسة الصغيرة مقابل الاحتفاظ ببياناتها التجارية على عتادها الخاص. وتُظهر النتائج أن البنى التطبيقية تجمع بين موثوقية الأنظمة القائمة على القواعد ومرونة النماذج اللغوية، وأن الحد الفاصل بينهما هو موضع القيمة الهندسية.

الكلمات المفتاحية --- النماذج اللغوية الكبيرة، البرمجيات نكدة متعددة المستأجرين، إدماج البيانات، استخراج المعلومات، تدقيق جودة البيانات، الإنسان في الحلقة، المحاكاة القائمة على الوكلاء، البيانات الاصطناعية، الاستدلال المحلي، بروتوكول سياق النموذج، **FastAPI** المخرجات المهيكلية

RÉSUMÉ

Toute plateforme de gestion se heurte au même paradoxe au moment de l'adoption : elle est inutile tant que les données du client n'y sont pas, alors que ces données n'existent que sous la forme d'artefacts hérités — tableurs multilingues, catalogues PDF imprimés, listes de prix photographiées. Ce mémoire présente UniFlow, une plateforme multi-locataire de gestion destinée aux secteurs du commerce de détail et de l'aviation, étendue par quatre surfaces d'IA coopérantes qui couvrent ensemble le cycle de vie des données d'un locataire : un *moteur d'intégration* (Onboarding Engine) qui crée l'entreprise, un *auditeur IA* qui la critique, un *laboratoire de simulation* qui lui donne vie, et un *serveur MCP* par lequel l'assistant du client lui-même la sert.

La contribution d'ingénierie unificatrice est un principe de conception unique appliqué trois fois : *le modèle de langage est utilisé exactement là où un jugement linguistique est nécessaire, et du code déterministe partout où la correction doit être garantie*. Dans le moteur d'intégration, le modèle interprète la signification des colonnes tandis que Python valide, dédoublonne, génère les identifiants et insère — aucune sortie non validée du modèle n'atteint la base de données, et un écran de révision humaine obligatoire ferme la boucle. Dans l'auditeur, treize vérifications déterministes établissent un *plancher garanti* de faits que le modèle peut expliquer et enrichir mais jamais supprimer ni minimiser, de sorte que le contenu factuel du rapport survit même à une indisponibilité totale du modèle. Dans le laboratoire de simulation, le modèle est appelé une seule fois pour rédiger deux cents personas de clients, puis un moteur Python déterministe et reproductible exécute un mois de leur comportement à travers les points d'accès réels de la plateforme, produisant des comportements d'achat émergents qu'aucune règle du code ne prescrit. S'y ajoute un *serveur MCP* tourné vers l'extérieur qui place l'entreprise dans l'assistant du client lui-même : quatorze outils couvrant les deux secteurs, par lesquels l'application ChatGPT peut consulter le catalogue, vérifier stocks et vols et finaliser des réservations invité, restreints à une seule marque par un nom d'entreprise explicite, sans identifiant ni configuration par marque.

Chaque fonctionnalité est proposée sur deux moteurs LLM interchangeables — un modèle hébergé de pointe et un modèle local servi par Ollama — et l'écart mesuré entre les deux quantifie ce qu'une petite entreprise sacrifie en gardant ses données commerciales sur son propre matériel. Les résultats montrent que les architectures en couches offrent la fiabilité des systèmes à base de règles avec la souplesse des modèles de langage, et que la frontière entre les deux est précisément là où réside la valeur d'ingénierie.

Mots clés — Grands modèles de langage, SaaS multi-locataire, intégration de données, extraction d'information, audit de qualité des données, humain dans la boucle, simulation à base d'agents, données synthétiques, inférence locale, FastAPI, sortie structurée