
Ph.D. Research Proposal

Leveraging Large Language Model (LLM)-Based Artificial Intelligence for Building Scalable Software Applications

Submitted in Fulfillment of the Requirements for
Admission and Registration to the Doctoral Program
in Computer Science

Prepared by :

Mouhadjer Saif Eddine

Academic Year: 2025–2026

1 Introduction

1.1 Background

Over the past decade, software systems have evolved from monolithic architectures toward distributed, cloud-native ecosystems engineered for massive scale, high availability, and rapid iteration. In parallel, Large Language Models (LLMs) – transformer-based neural networks trained on vast corpora of natural language and source code – have demonstrated remarkable capabilities in code understanding, generation, and reasoning [1, 2].

The convergence of these two trends raises a central question that this research proposes to investigate: *can LLM-based artificial intelligence be systematically and reliably leveraged to design, build, and operate scalable software applications*, beyond simple code completion?

1.2 Motivation

Over the past few years, I have developed several personal projects, including mobile applications and web platforms. In these projects, I faced recurring challenges in building systems that could scale effectively under time and resource constraints. While tools like GitHub Copilot and Claude significantly accelerated routine code generation, I noticed that they consistently failed when dealing with complex architectural decisions.

The generated code often met functional requirements but neglected critical scalability aspects such as distributed design, resource management, and long-term maintainability. These experiences highlighted a clear limitation: scalability is not just about functional correctness, but depends heavily on sound architectural choices.

Although LLMs have advanced AI-assisted software engineering, current research remains largely focused on code productivity, leaving the architectural and scalability dimensions insufficiently explored. This gap, which I encountered directly in my own work, is the primary motivation for this research

1.3 General Approach

This proposal outlines a research program to (1) systematically review the state of the art regarding LLM usage in software architecture and scalable system design, (2) design a conceptual and operational framework integrating LLM agents across the software development lifecycle with explicit scalability objectives, (3) implement a prototype of this framework, and (4) empirically evaluate it against traditional development practices using representative case studies.

2 Preliminary Literature Review

Large Language Models have evolved significantly from traditional statistical and recurrent networks to transformer architectures [3], enabling models like GPT-3 [1] to demonstrate few-shot learning and Codex [2] to handle code generation from natural language. In software engineering, LLMs are increasingly used for code generation, testing, and limited design support, offering productivity gains while raising concerns about correctness, security, and over-reliance on generated code. Modern scalable systems commonly employ microservices, containerization (e.g., Kubernetes), and patterns such as Circuit Breaker, CQRS, and Event Sourcing. Recent work has also explored multi-agent LLM frameworks with human-in-the-loop validation to address current limitations in architectural reasoning. Nevertheless, there remains a significant gap: few studies integrate LLMs across the full development lifecycle with scalability as a core concern, supported by systematic empirical evaluation. This gap motivates the proposed research. Note that this literature review is preliminary and will be expanded through a systematic review in the first year of the program.

3 Problem Statement, Research Questions and Objectives

3.1 Problem Statement

Despite the growing adoption of LLMs in software development tools, there is currently a lack of systematic frameworks that:

1. Integrate LLMs into the full software development lifecycle for scalable applications, beyond code completion.
2. Evaluate the architectural soundness and scalability implications of LLM-generated designs using established, quantitative metrics.
3. Provide structured mechanisms for human oversight, validation, and continuous refinement of LLM-driven architectural and implementation decisions.

3.2 Research Questions

- **RQ1:** To what extent can LLMs assist in architectural decision-making for scalable, distributed applications?
- **RQ2:** How does LLM-generated code compare to human-written code in terms of scalability, performance, and maintainability?
- **RQ3:** What framework design and human-in-the-loop strategies maximize the reliability of LLM-assisted scalable application development?

- **RQ4:** What are the cost, latency, and resource trade-offs of integrating LLM agents into production-oriented development and operational pipelines?

3.3 Research Objectives

The main objectives of this research are to first conduct a systematic literature review of LLM applications in software engineering and scalable system design. Building on that, the study aims to design a conceptual framework that integrates LLM-based agents across the architecture, development, and operational stages of scalable applications, while incorporating clear human-in-the-loop checkpoints. This will be followed by the implementation of a working prototype that embodies the proposed framework. The research will then empirically evaluate this framework against traditional development methods using key quantitative metrics, including development efficiency, scalability, code quality, resource utilization, and cost. Finally, it seeks to derive practical design guidelines and best-practice recommendations to support practitioners and organizations in effectively adopting LLM-based agents for building scalable software systems.

3.4 Significance of the Study

This research is expected to contribute to academic literature by extending the understanding of AI-assisted software engineering with a specific focus on scalability – an aspect that has received comparatively limited attention relative to functional correctness. From a practical standpoint, the resulting framework and guidelines are intended to support software organizations in adopting LLM-based development tools while mitigating risks related to architectural quality, reliability, and cost.

4 Proposed Research Methodology

4.1 Planned Research Phases

1. **Phase 1 – Systematic Literature Review:** A structured review of existing work on LLMs in software engineering, scalable architecture design, and AI agent frameworks, following a documented search and selection protocol.
2. **Phase 2 – Framework Design:** Development of a conceptual framework integrating LLM-based agents into architectural planning, code generation, deployment configuration, and runtime monitoring/optimization, including explicit human-in-the-loop validation checkpoints.
3. **Phase 3 – Prototype Implementation:** Implementation of a working prototype of the proposed framework using contemporary LLM APIs, containerization, and orchestration tooling.

4. **Phase 4 – Empirical Evaluation:** Application of the prototype to a set of representative case-study applications (e.g., a RESTful backend, a real-time data pipeline, and a multi-tenant SaaS module), each developed both traditionally and using the proposed framework.
5. **Phase 5 – Analysis and Synthesis:** Comparative analysis of results, derivation of design guidelines, and dissemination through publications and the final dissertation.

4.2 Anticipated Evaluation Metrics

Table 1: Preliminary Evaluation Metrics

Category	Indicative Metrics
Development Efficiency	Time-to-prototype, lines of code, iteration count
Scalability	Throughput (RPS), latency percentiles, scaling efficiency
Code Quality	Cyclomatic complexity, static analysis defect density
Maintainability	Maintainability index
Resource Efficiency	CPU/memory utilization under load
Cost	LLM API token consumption and cost

4.3 Resources Required

Execution of this research will require access to: (1) LLM API services (e.g., Anthropic Claude, OpenAI GPT) with an estimated budget for experimentation; (2) computing infrastructure for containerized deployment and load testing (local cluster or cloud credits); and (3) standard software engineering tooling for static analysis, testing, and monitoring (open-source).

5 Expected Contributions

This research is expected to make several meaningful contributions to the field. It will provide a systematic mapping of existing work on using Large Language Models for scalable software architecture and development. More importantly, it aims to introduce a new conceptual framework that integrates LLM-based agents throughout the entire software development lifecycle, with strong emphasis on scalability and appropriate human oversight. The study will also include a working prototype to demonstrate the practical feasibility of this framework. Through empirical evaluation, it will compare LLM-assisted development with traditional approaches in terms of development efficiency, scalability, code quality, and overall cost. Finally, the research will offer practical design guidelines and best-practice recommendations to help developers and

organizations adopt LLM-based tools more effectively for building scalable applications.

6 Proposed Timeline

The doctoral research is planned over a three-year period, organized as follows.

Table 2: Indicative Research Timeline

Activity	Y1-S1	Y1-S2	Y2-S1	Y2-S2	Y3-S1	Y3-S2
Systematic Literature Review	X	X				
Framework Design		X	X			
Prototype Implementation			X	X		
Empirical Evaluation				X	X	
Analysis & Refinement					X	X
Dissertation Writing					X	X
Publications & Defense Preparation			X	X	X	X

Note: “Y1-S1” denotes Year 1, Semester 1, and so on. This timeline will be refined in coordination with the supervisor and doctoral committee following formal registration.

7 Conclusion

This proposal has outlined the motivation, problem statement, research questions, objectives, preliminary literature review, proposed methodology, expected contributions, and indicative timeline for a doctoral research program investigating the use of LLM-based artificial intelligence in building scalable software applications. The candidate respectfully requests approval from the department and doctoral committee to proceed with registration and the commencement of this research.

References

- [1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [2] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et al.*, “Evaluating large language models trained on code,” *arXiv preprint arXiv:2107.03374*, 2021.