DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Thesis Proposal

Trung Nguyen

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Thesis Proposal

Design and Evaluation of Multi-Agentic AI Systems for Contract Lifecycle Management in B2B SaaS Platforms

Author:	Trung Nguyen
Supervisor:	Prof. Dr. [Supervisor Name]
Advisor:	[Advisor Name]
Submission Date:	September 9, 2025

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, September 9, 2025 Trung Nguyen

Motivation

Contract Lifecycle Management (CLM) [Wik25] is critical in enterprise environments, serving as operational and financial anchors for complex projects involving multiple stakeholders and regulatory requirements. Current CLM tools rely on rule-based automation or single-agent LLMs, presenting limitations in scalability, explainability, and domain-specific optimization.

Multi-agent systems (MAS) offer promising potential for addressing these limitations through specialized task decomposition, inter-agent collaboration, and enhanced explainability mechanisms. However, there is a significant research gap in applying multi-agent systems to enterprise contract management workflows, particularly regarding compliance-aware orchestration and explainability requirements.

Current CLM systems face critical challenges: scalability limitations in complex workflows, insufficient explainability leading to low enterprise trust, and inadequate handling of compliance and regulatory constraints across industries and jurisdictions.

Research gaps include: no established frameworks for multi-agent orchestration in enterprise CLM workflows, limited empirical evaluation comparing single-agent versus multi-agent approaches, insufficient attention to compliance-aware AI system design, and lack of standardized evaluation methodologies for trust and explainability assessment.

Core Research Problem: How can multi-agent systems be designed and evaluated to improve efficiency, accuracy, and trust in contract lifecycle processes?

Research Questions

Primary Research Question: How can a multi-agent framework be designed and implemented to improve contract analysis efficiency and accuracy in enterprise CLM systems?

Secondary Research Questions:

1. What agent specialization patterns are most effective for contract analysis tasks (drafting, review, compliance checking)?
2. How does a multi-agent approach compare to single-agent systems in contract analysis accuracy and processing time?
3. What explainability features are essential for enterprise users to trust multi-agent contract analysis results?

Research Objectives: (1) Design a focused multi-agent framework for contract analysis based on industry foundation papers, (2) Develop a working prototype with 2-3 specialized agents, (3)

Conduct comparative evaluation using real contract data, (4) Establish practical implementation guidelines for enterprise CLM integration.

Methodology

Foundation Framework Design Based on industry foundation papers (e.g., Wooldridge's MAS principles, enterprise AI frameworks), design a focused multi-agent architecture for contract analysis. Define agent roles: Contract Analyzer Agent (extraction, classification), Compliance Checker Agent (regulatory validation), and Orchestrator Agent (task coordination).

Prototype Implementation Develop a working prototype implementing the designed framework. Core components: (1) Agent communication protocols using message passing, (2) Contract processing pipeline with NLP integration, (3) Basic explainability features (decision reasoning, confidence scores), (4) Simple web interface for testing.

Evaluation and Validation Conduct comparative evaluation using real contract datasets. Metrics: (1) Processing accuracy (precision, recall for contract analysis tasks), (2) Performance comparison (multi-agent vs single-agent processing time), (3) User trust assessment through explainability features, (4) System reliability and error handling.

Implementation Approach: Rapid prototyping using Python with existing NLP libraries (spaCy, transformers), agent framework (asyncio for concurrent processing), and evaluation using standard contract analysis benchmarks.

Starting Literature

Multi-Agent Systems and LLM Applications: [Han+25] identify key challenges and open problems in LLM-based multi-agent systems, while [Shu+24] present design and evaluation frameworks for enterprise GenAI multi-agent collaboration.

Agent Architecture and Collaboration: [Bel+25] explore the potential of small language models for agentic AI, [Wan+24] demonstrate how mixture-of-agents approaches enhance LLM capabilities, and [Wu+23] introduce AutoGen as a framework for enabling next-generation LLM applications via multi-agent conversation.

Enterprise Applications and Contract Management: Recent work focuses on practical implementation of multi-agent systems in enterprise contexts, with particular emphasis on collaboration patterns, agent specialization, and real-world deployment challenges in business workflows.

Schedule

Phase 1: Foundation Framework Design (October - November 2025) Study industry foundation papers on multi-agent systems and enterprise AI frameworks. Design focused multi-agent architecture for contract analysis with 2-3 specialized agents. Define agent roles, communication protocols, and system requirements.

Phase 2: Prototype Implementation (December 2025 - January 2026) Develop working prototype using Python with NLP libraries (spaCy, transformers). Implement agent communication protocols, contract processing pipeline, basic explainability features, and simple web interface for testing.

Phase 3: Evaluation and Validation (February - March 2026) Conduct comparative evaluation using real contract datasets. Test multi-agent vs single-agent performance, assess explainability features, and validate system reliability. Complete data analysis and results documentation.

Milestones:

- November 2025: Framework design complete, agent roles defined
- January 2026: Working prototype with core functionality
- March 2026: Evaluation complete, thesis writing finished

Expected Contributions

Scientific Contributions:

- Focused multi-agent framework for contract analysis with 2-3 specialized agents, demonstrating practical implementation patterns and coordination strategies
- Comparative evaluation results showing multi-agent vs single-agent performance in contract analysis tasks, with concrete metrics for accuracy and processing time
- Practical explainability features for enterprise contract analysis, validated through user testing and trust assessment

Practical Contributions:

- Working prototype implementation with documented architecture, agent design patterns, and integration approach for enterprise CLM systems
- Evaluation framework and metrics for assessing multi-agent contract analysis systems, with practical guidelines for implementation
- Open-source codebase demonstrating rapid prototyping approach using Python and existing NLP libraries for enterprise AI applications

Impact Areas: Contract analysis automation, Multi-Agent Systems in enterprise software, Legal Technology prototyping, and practical AI implementation patterns.

Bibliography

[Bel+25] P. Belcak, G. Heinrich, S. Diao, Y. Fu, X. Dong, S. Muralidharan, Y. C. Lin, and P. Molchanov. *Small Language Models are the Future of Agentic AI*. 2025. arXiv: 2506.02153 [cs.AI].

[Han+25] S. Han, Q. Zhang, Y. Yao, W. Jin, and Z. Xu. *LLM Multi-Agent Systems: Challenges and Open Problems*. 2025. arXiv: 2402.03578 [cs.MA].

[Shu+24] R. Shu, N. Das, M. Yuan, M. Sunkara, and Y. Zhang. *Towards Effective GenAI Multi-Agent Collaboration: Design and Evaluation for Enterprise Applications*. 2024. arXiv: 2412.05449 [cs.CL].

[Wan+24] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou. *Mixture-of-Agents Enhances Large Language Model Capabilities*. 2024. arXiv: 2406.04692 [cs.CL].

[Wik25] Wikipedia contributors. *Contract lifecycle management*. Last edited on 5 September 2025, at 06:44 (UTC). 2025.

[Wu+23] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang. *AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation*. 2023. arXiv: 2308.08155 [cs.AI].