Agentic AI - Design and Evaluation of an MCP-Based Architectural Foundation for Enterprise Data Interactions

Master Thesis – Anton Dolgov – 20. May 2026

Supervisors: Prof. Dr. Martin Breunig, Dr.-Ing. Paul Vincent Kuper – Geodetic Institute
Kilian Lehn – dmTECH

Thesis Overview

- **AI in Business: 90%** of organizations use AI, but only **7%** achieve full enterprise integration (McKinsey 2025)
- **Thesis Context:** Conducted at **dmTECH** to automate data retrieval for their **store layout management system**
- **Challenges:** Manual **GraphQL** query construction is time-consuming and requires specialized domain knowledge
- **Approach:** Use **MCP (Model Context Protocol)** to enable **LLMs** to autonomously interact with data

Methods

- **LLM Model Selection**
  - 3 reasoning models (GPT-5.1, Claude-Opus-4.6, Gemini-2.5-Pro)
  - 3 lightweight models (GPT-5-Mini, Gemini-2.5-Flash, Claude-Haiku-4.5)
- **Evaluation Metrics**
  - **Performance:** Token usage, tool calls, resolution time
  - **Quality:** Field & data precision and recall, exact match
  - **Human evaluation:** Error analysis and categorization

- **MCP Architecture**

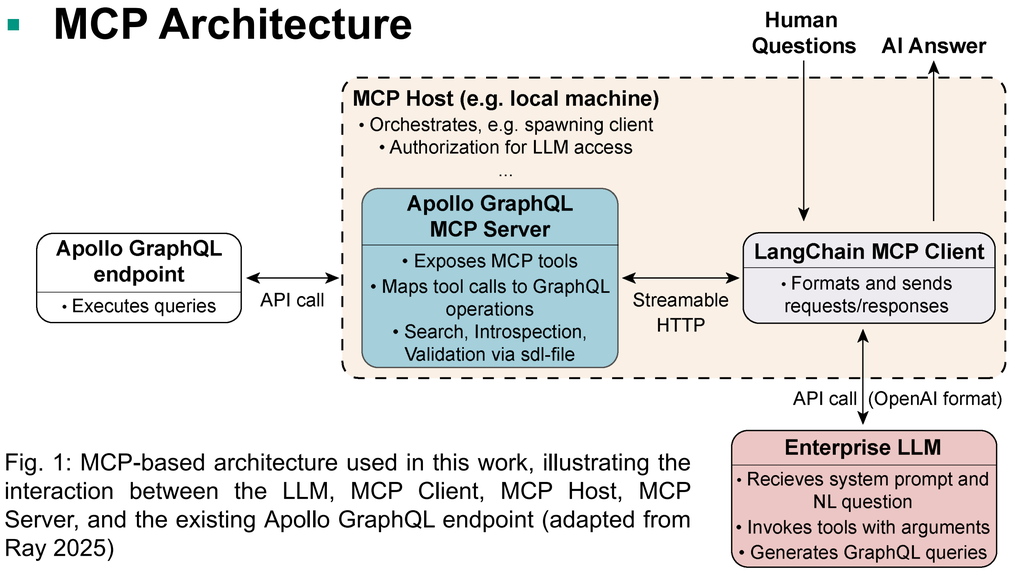

Fig. 1: MCP-based architecture used in this work, illustrating the interaction between the LLM, MCP Client, MCP Host, MCP Server, and the existing Apollo GraphQL endpoint (adapted from Ray 2025)

Results

- **Test Dataset:** 14 natural language questions derived from IT-tickets with gold standard query
  - 3 difficulty levels from basic lookups (simple questions) to deep nested connections (medium & complex questions)

Tab. 1: Evaluation results for the test dataset by difficulty level (prec: precision, rec: recall)

	Simple (5 Q.)	Medium (6 Q.)	Complex (3 Q.)
Average Performance	4 – 6 tool calls 20k – 50k tokens 15 – 75 s	10 – 30 tool calls 100k – 600k tokens 25 – 250 s	10 – 30 tool calls 50k – 700k tokens 20 – 250 s
Quality (successful queries)	Field rec/prec: 0.7 – 1.0 Data rec/prec: 0.4 – 1.0	Field prec: 0.28 – 1.0 Field rec: 0.13 – 1.0 Data prec: 0.025 – 1.0 Data rec: 0.014 – 1.0	Field prec: 0.70 – 0.92 Field rec: 0.43 – 0.86 Data prec: 0.0035 – 0.83 Data rec: 0.045 – 0.86
Common Issues	Extra fields unnecessary filters, slicing decisions, field hallucinations	Validation, timeout & query complexity errors, wrong entry points, poor slicing/pagination, over/under-fetching, field hallucinations	

- Reasoning models outperform lightweight ones
- Simple questions: High success rate → **solved**
- Medium & complex questions: Success depends on question clarity and model capacity
- **With improved prompt and system adjustments**
  - **Clear questions:** More stable, lower token usage, faster execution
  - **Ambiguous questions:** Improved quality, but at higher performance cost

Conclusion

- LLMs can reliably translate natural language to GraphQL queries in an autonomous **agentic AI workflow**
- Complex questions require careful **prompt engineering**
- Remaining challenges**: Entry point selection**, over- and underfetching, recognize unproductive workflows

Outlook

- (new) questions with different **ambiguity levels**
- **Token caching** to counter token accumulation
- Integration with ticket system to **automate workflows**
- Testing of different prompting strategies

References

McKinsey & Company (Nov. 2025). The state of AI in 2025: Agents, innovation, and transformation. Retrieved 2026-02-01 from https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai#/.
Ray, Partha Pratim (Apr. 18, 2025). A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions. Preprint. DOI: 10.36227/techrxiv.174495492.22752319/v1.
Yu, Tao et al. (Feb. 2, 2019). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. Preprint. DOI: 10.48550/arXiv.1809.08887. arXiv:1809.08887[cs].
Apollo Graph, Inc. (Jan. 2026a). Apollo MCP server Deepwiki. Retrieved 2026-02-10 from https://deepwiki.com/apollographql/apollo-mcp-server.
Jiang, Juyong et al. (Feb. 28, 2026). "A survey on large language models for code generation". In: ACM Transactions on Software Engineering and Methodology 35.2, pp. 1–72. ISSN: 1049-331X, 1557-7392. DOI: 10.1145/3747588
LangChain (2026). LangChain documentation. Retrieved 2026-01-20 from https://reference.langchain.com

Contact: ugsdo@student.kit.edu, kuper@kit.edu