Lyra: A Local-First MCP Server for AI-Collaborative Desktop Research with Evidence Graph Construction

Katsuya Shibuki

2026-01-12

Summary

Research demands auditable evidence chains—the ability to trace every claim back to its source. Lyra is an open-source server implementing the Model Context Protocol (MCP, Anthropic 2024)—a standard interface for connecting AI assistants to external tools—that enables AI assistants to conduct desktop research using structured provenance, providing accurate and auditable evidence. The software exposes research capabilities—web search, content extraction, natural language inference, and evidence graph construction—as structured tools that MCP-compatible AI clients can invoke directly.

I designed Lyra to separate strategic reasoning—performed by the AI assistant in the MCP client—from mechanical execution: evidence discovery, classification, and scoring (Figure 1).

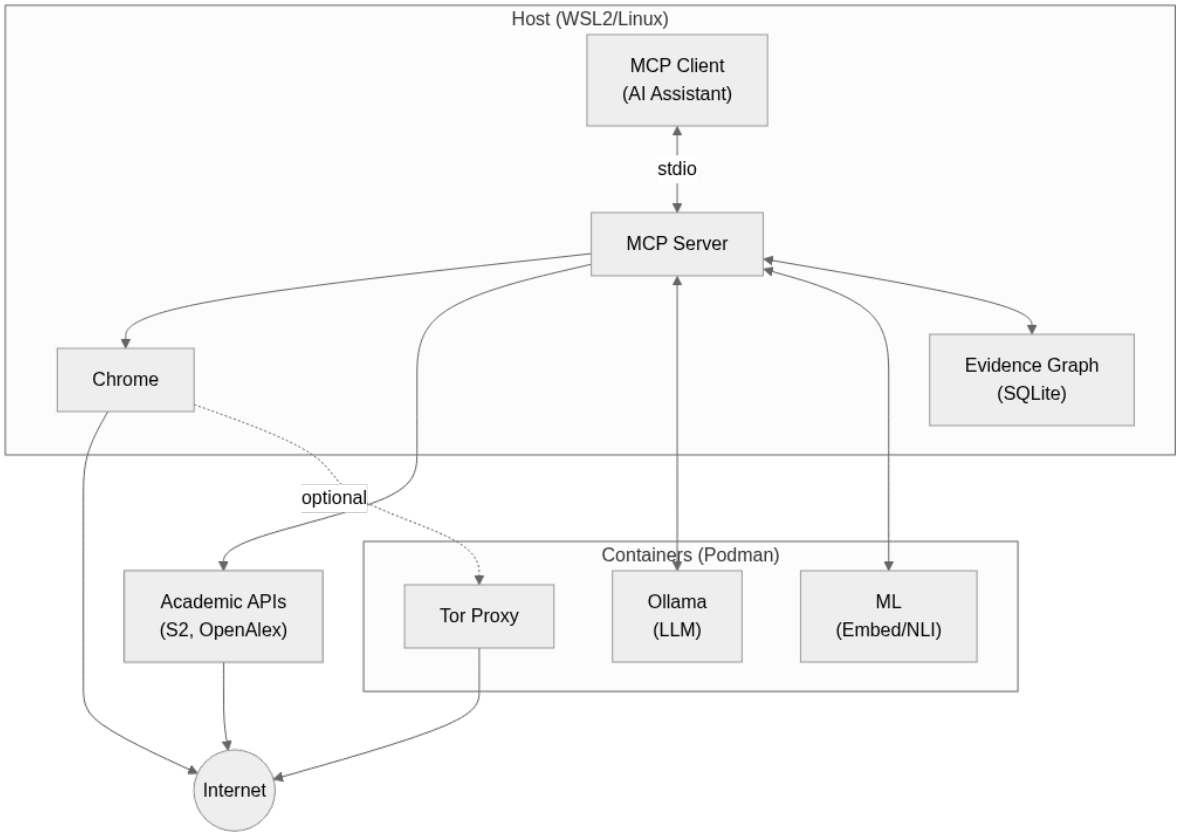

Figure 1: System architecture. The MCP server runs on the host; ML inference containers are network-isolated to prevent data exfiltration.

The AI assistant handles query design and synthesis, while Lyra executes search, extraction, and NLI-based stance detection. Lyra functions as a navigation tool: it discovers and organizes relevant sources, while detailed analysis of primary sources remains the researcher's responsibility (Figure 2).

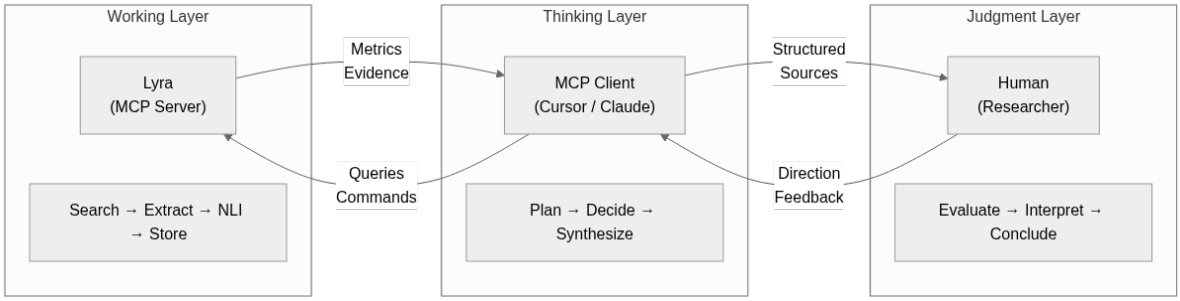

Figure 2: Three-layer collaboration model. The Thinking layer (human) provides domain expertise and final evaluation. The Reasoning layer (MCP client) handles query design and synthesis. The Working layer (Lyra) executes mechanical tasks: search, extraction, and NLI.

The software incorporates three machine learning components for local inference: a 3B-parameter language model (Qwen2.5, Qwen et al. 2025) for claim extraction, BGE-M3 embeddings (Chen et al. 2024) for semantic search, and a DeBERTa-based classifier (He et al. 2021) for stance detection. The system automatically detects GPU availability and applies appropriate container configurations; while CPU-only operation is supported, GPU acceleration is strongly recommended due to significant performance differences. Lyra constructs an **evidence graph** linking extracted claims to source fragments with structured provenance metadata (Figure 3).

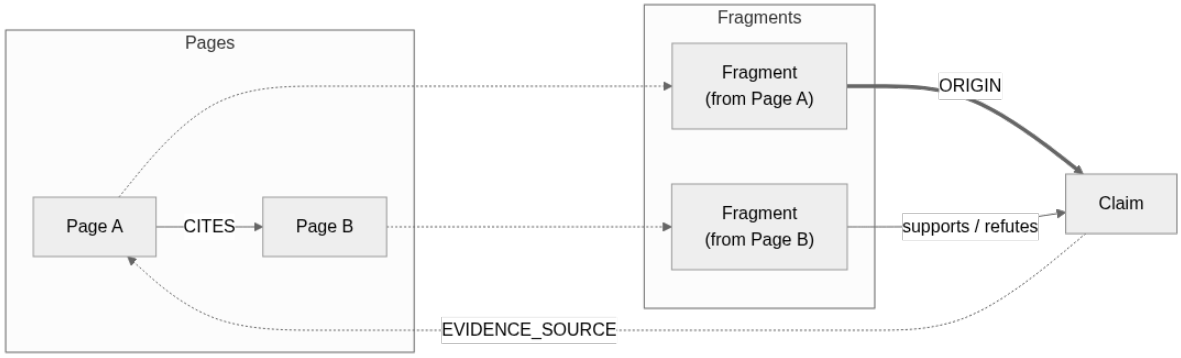

Figure 3: Evidence graph structure. Claims are extracted from fragments (ORIGIN edges track provenance). Cross-source verification via NLI creates SUPPORTS/REFUTES edges. The exploration score aggregates weighted evidence. CITES edges track academic citations.

Each claim accumulates an exploration score (`nli_claim_support_ratio`) derived from Natural Language Inference (NLI, Bowman et al. 2015) judgments—automated classification of

whether a text supports, refutes, or is neutral toward a claim. This score aggregates NLI-weighted evidence (supports vs. refutes) into a 0–1 ratio used for navigation and ranking, not as a statistically rigorous probability of truth.

Statement of Need

Lyra targets researchers and practitioners—particularly in healthcare, biomedical sciences, and other fields well-covered by academic databases—who need AI-assisted evidence gathering for desktop research. It provides auditable evidence chains: the ability to trace claims to their sources for verifying conclusions. Large language models, however, are inherently probabilistic; verifying that AI-generated citations accurately reflect source materials demands substantial manual effort. Existing tools address different aspects of this challenge: cloud-based assistants (Perplexity AI 2022; Elicit 2021) provide rapid retrieval with citation links; browser automation (Selenium 2013; Microsoft 2019) offers programmatic access; RAG frameworks (Chase 2022; Liu 2022) specialize in document retrieval. However, these tools typically produce disposable answers—results that do not persist or improve with use. Lyra takes a different approach: it builds a persistent evidence graph that accumulates across research sessions, enabling traceable conclusions that grow stronger with continued use and researcher feedback.

From a context engineering perspective—designing systems that supply AI models with accurate, relevant information—Lyra constructs a transparent evidence graph that provides AI clients with traceable information. Every claim links to source fragments, which link to page URLs, creating an auditable chain from assertion to origin. The graph explicitly represents both supporting and refuting evidence, with exploration scores quantifying the evidence balance. Researchers can trace any claim back to its source text and evaluate the reasoning path themselves.

The software follows a local-first design: machine learning inference (LLM, embeddings, NLI) runs on the researcher's hardware, and all research artifacts—the evidence graph, extracted claims, and source fragments—are stored locally in a SQLite database. For evidence discovery, Lyra retrieves content via browser-based web search and academic APIs (Allen Institute for AI 2025; Priem, Piwowar, and Orr 2022), with identifier extraction from SERP URLs enabling cross-source enrichment and DOI-based deduplication. Each research task defines a central hypothesis to verify; Lyra then finds evidence supporting or refuting this hypothesis. A human-in-the-loop mechanism enables researchers to correct NLI judgments; these corrections are accumulated for planned domain adaptation via Low-Rank Adaptation (LoRA, Hu et al. 2021) fine-tuning.

AI Usage Disclosure

I used Cursor and Claude Code during Lyra's development. AI assistance covered code generation, refactoring, test scaffolding, and documentation drafting. All AI-assisted outputs were reviewed, and validated by me. Core design decisions—including the evidence graph architecture, three-layer collaboration model, and MCP tool interface—were made by me, as documented in 17 Architecture Decision Records covering local-first principles, evidence graph structure, and security models.

Acknowledgements

Lyra builds upon several open-source projects: Ollama (Ollama 2023) for local language model runtime, Playwright (Microsoft 2019) for browser automation, Trafilatura (Barbaresi 2021) for web content extraction, and Hugging Face Transformers (Wolf et al. 2020) for NLI and embedding models. Academic metadata is provided by the Semantic Scholar (Allen Institute for AI 2025) and OpenAlex (Priem, Piwowar, and Orr 2022) APIs.

References

Allen Institute for AI. 2025. "Semantic Scholar Academic Graph API." 2025. https://www.semanticscholar.org/product/api.

Anthropic. 2024. "What Is the Model Context Protocol (MCP)?" 2024. https://modelcontextprotocol.io/docs/getting-started/intro.

Barbaresi, Adrien. 2021. "Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, edited by Heng Ji, Jong C. Park, and Rui Xia, 122–31. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-demo.15.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. "A Large Annotated Corpus for Learning Natural Language Inference." August 21, 2015. https://doi.org/10.48550/arXiv.1508.05326.

Chase, Harrison. 2022. "LangChain." https://github.com/langchain-ai/langchain.

Chen, Jianlv, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. "M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation." February 5, 2024. https://arxiv.org/abs/2402.03216v5.

Elicit. 2021. "Elicit: AI for Scientific Research." 2021. https://elicit.com/welcome.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. "DeBERTa: Decoding-enhanced BERT with Disentangled Attention." October 6, 2021. https://doi.org/10.48550/arXiv.2006.03654.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. "LoRA: Low-Rank Adaptation of Large Language Models." October 16, 2021. https://doi.org/10.48550/arXiv.2106.09685.

Liu, Jerry. 2022. "LlamaIndex." https://doi.org/10.5281/zenodo.1234.

Microsoft. 2019. "Playwright." https://github.com/microsoft/playwright.

Ollama. 2023. "Ollama." https://github.com/ollama/ollama.

Perplexity AI. 2022. "Perplexity." 2022. https://www.perplexity.ai/.

Priem, Jason, Heather Piwowar, and Richard Orr. 2022. "OpenAlex: A Fully-Open Index of Scholarly Works, Authors, Venues, Institutions, and Concepts." June 17, 2022. https://doi.org/10.48550/arXiv.2205.01833.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. 2025. "Qwen2.5 Technical Report." January 3, 2025. https://doi.org/10.48550/arXiv.2412.15115.

Selenium. 2013. "Selenium." https://github.com/SeleniumHQ/selenium.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, et al. 2020. "Transformers: State-of-the-Art Natural Language Processing." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, edited by Qun Liu and David Schlangen, 38–45. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6.