

concept-omcp: An Embeddable, Privacy-Preserving MCP Server for Semantic OMOP Vocabulary Search in Secure Data Environments

Niko Möller-Grell¹, Shihao Shenzhang¹, Ian Farr^{6,7}, Richard Dobson^{1,2,3,4,5}, Vishnu V Chandrabalan^{6,7}

1 Department of Biostatistics and Health Informatics, Institute of Psychiatry, Psychology and Neuroscience (IoPPN), King's College London, London, UK

2 Institute for Health Informatics, University College London, London, UK

3 NIHR Biomedical Research Centre, University College London Hospitals National Health Service Foundation Trust, London, UK

4 Health Data Research UK London, University College London, London, UK

5 NIHR Biomedical Research Centre, South London and Maudsley National Health Service Foundation Trust and King's College London, London, UK

6 Lancashire Teaching Hospitals Trust, Preston, UK

7 Lancaster University, Lancaster, UK

Background

Mapping free-text clinical descriptions to OMOP standard concepts is the first and most error-prone step of nearly every observational study, and it is the bottleneck when large language model (LLM) agents attempt to build phenotypes or cohorts autonomously. In prior work we introduced OMCP (1), a suite of Model Context Protocol (MCP) servers exposing OMOP CDM operations to LLM agents, and subsequently extended it to the OHDSI application layer with MCP servers wrapping the WebAPI vocabulary and cohort services (2). Those vocabulary tools delegate to a running WebAPI/ATLAS stack and rely on lexical (`LIKE`-style) concept lookup, which misses synonymy and clinically equivalent phrasings.

To give our FastOMOP agents (3) true *semantic* concept retrieval we had previously stood up a Milvus vector database holding ~4 million MedEmbed embeddings of OMOP concept names and synonyms. While accurate, this introduced a heavyweight, always-on service (~17 GB resident in float32) requiring its own infrastructure making it a poor fit for the Secure Data Environments (SDEs) and Trusted Research

Environments (TREs) where OMOP data actually lives, which are routinely network-isolated and resource-constrained.

We therefore built **concept-mcp**: a single, self-contained MCP server that provides semantic, relationship-aware OMOP concept search from a *prebuilt, quantized, in-memory artifact* with **no database server and no network dependency**. It is designed to be dropped into an SDE alongside the data, run on a laptop, or imported in-process by an agent.

Methods

Corpus and embeddings. From the OMOP/Athena vocabulary (v5, Oct 2025) we generate one embedding per term, each standard concept's primary name plus its English synonyms, using the open-source biomedical encoder `abhinand/MedEmbed-large-v0.1` (1024-d), L2-normalised. The current artifact covers **4,536,983 term vectors across 2,950,917 concepts**.

Quantized in-memory index. Vectors are stored in a `turbovec TurboQuantIndex` at **4-bit quantization**, reducing the index from ~17 GB (float32) to **2.18 GB on disk and in RAM**. Concept metadata and denormalised relationships (parent concept, participating relationship types) are held in a read-only **SQLite** database (1.51 GB). The total deployable artifact is < 4 GB and loads in seconds; the server never re-embeds or re-indexes at runtime. A build manifest pins the embedding model, dimension, `turbovec` version and build timestamp for reproducibility, and an `append` path adds new vocabulary domains without re-embedding the existing corpus.

Server and tools. A thin FastMCP layer (`stdio` for agent subprocesses, Streamable HTTP for shared deployment) exposes three tools: `search_concepts` (semantic search with optional domain filter and a lexical exact-match fallback), `get_concept_by_id` (direct

lookup with relationship metadata), and `find_concepts_by_relationship`. The same core (`ConceptStore`) can be imported directly, letting an agent bypass the MCP transport entirely; it is a signature-compatible drop-in for our previous Milvus search tool, so adoption is a one-line change.

Relation to prior OHDSI work: Two existing projects address adjacent problems, and we position `concept-mcp` explicitly against them:

Hecate (4) is a mature, hosted semantic-search service whose MCP component is a TypeScript client calling a remote Rust/Qdrant/PostgreSQL backend; it offers richer features we do not (e.g. PHOEBE relationships, bidirectional hierarchy expansion) but cannot run inside an isolated SDE. `OhdsiVocabVectorStore` (5) is a data-preparation pipeline, that builds embeddings via the Azure OpenAI cloud API and serves them from PostgreSQL/pgvector. `concept-mcp`'s distinct contribution is the **zero-infrastructure, fully-offline, locally-embedded** deployment: a portable artifact and an in-process/MCP interface engineered for privacy-preserving environments and direct integration into LLM-agent pipelines.

Results

The packaged server loads its 2.2 GB index plus 1.5 GB SQLite and answers queries within seconds of start-up, with no external service running. In an end-to-end MCP `stdio` smoke test, all three tools registered and returned correctly: `search_concepts("type 2 diabetes mellitus", domain=Condition)` returned *Type 2 diabetes mellitus* (concept 201826) at the top, followed by *Diabetes mellitus* (201820) and *Type 2 diabetes mellitus in obese* (4304377); `get_concept_by_id(201826)` returned the concept with its parent (201820). The server replaces our Milvus deployment at roughly **one-eighth of its memory footprint** while removing the standalone

database entirely, and is a signature-compatible substitute for the agent's existing search tool.

Conclusion

concept-omcp demonstrates that semantic, relationship-aware OMOP concept search can be delivered as a portable, privacy-preserving artifact rather than a hosted service, searchable offline, embeddable in an agent, and deployable inside a network-isolated SDE without a database server. It complements the broader OMCP/OHDSI tooling rather than replacing it, and is distinct from existing semantic-vocabulary efforts (Hecate; OhdsiVocabVectorStore) in being infrastructure-free, locally-embedded, and agent-native. Current limitations include reliance on a fixed prebuilt vocabulary snapshot and a relationship-filter tool whose recall is bounded by a candidate scan; planned work adds usage-frequency-weighted ranking (as in (5)) and richer hierarchy traversal (as in (4)). The server, build pipeline, and evaluation harness are released open-source under the FastOMOP project.

References

1. Shenzhang S, Möller-Grell N, Jiang Z, Dobson R, Chandrabalan V. OMCP: Model Context Protocol Servers for the OMOP Common Data Model. Global OHDSI 2025. https://www.ohdsi.org/wp-content/uploads/2025/10/304-2025_Global_OHDSI_OMCP-Report-Sihao-Shenzhang.pdf Gardner RM, Golubjatnikov OK, Laub RM, Jacobson JT, Evans RS. Computer-critiqued blood ordering using the HELP system. *Comput Biomed Res* 1990;23:514-28.
2. Möller-Grell N, Shenzhang S, Jiang ZJ, Dobson R, Chandrabalan VV. Extending FastOMOP to the OHDSI Application Layer: MCP Servers for Vocabulary, Cohort Definition, and Statistical Profiling. 2026.
3. Möller-Grell N, Shenzhang S, Jiang ZJ, Dobson RJB, Chandrabalan VV. FastOMOP: a foundational architecture for reliable agentic real-world evidence generation on OMOP CDM data [preprint]. arXiv:2604.24572. 2026. Available from: <https://arxiv.org/abs/2604.24572>
4. Hecate: OHDSI Vocabulary Semantic Search Engine (Pantheon-HDS). <https://github.com/OHDSI/Hecate>
5. Schuemie M. OhdsiVocabVectorStore. <https://github.com/schuemie/OhdsiVocabVectorStore>
6. OHDSI Common Data Model. <https://ohdsi.github.io/CommonDataModel/>