FAISS vs. ChromaDB: A Technical Dialog

Junior Developer: Hi, I've been looking at our RAG agent's code. We're using FAISS, but I've also seen ChromaDB mentioned in a lot of articles. They both seem to do vector search, but what's the actual difference? When would we choose one over the other?

Senior Engineer: That's a great question, as it's a key architectural decision. The biggest difference is that FAISS is a library, while ChromaDB is a database.

Junior Developer: What do you mean by "just a library"?

Senior Engineer: FAISS (Facebook AI Similarity Search) is an incredibly powerful and high-performance toolkit for similarity search. It runs directly in your application's memory. When you use it, you build a vector index from scratch, load it into RAM, and perform searches. When your application shuts down, that index is gone unless you manually save it to a file. It's lightweight, blazing fast, and gives you a lot of low-level control.

Junior Developer: So, it's temporary and lives with the application?

Senior Engineer: Exactly. That's why it was perfect for our project. We process a single PDF per session, create an index for it, and then we don't need to keep it around. Now, ChromaDB is a full-fledged vector database. It runs as a separate server process, just like a traditional database like PostgreSQL.

Junior Developer: Ah, so it's persistent?

Senior Engineer: Right. With ChromaDB, you can add, update, and delete vectors, and that data is stored permanently on disk. It handles all the server management, data persistence, and even has features for filtering by metadata. It's designed to be a long-term, managed knowledge base that your application can connect to.

Junior Developer: So, if we were building an application that needed to remember every document ever uploaded and search across all of them at once, we would use ChromaDB?

Senior Engineer: Precisely. You'd use Chroma for a persistent, multi-user knowledge base. But for our current use case—a fast, single-session, in-memory search—FAISS is the simpler and more efficient choice. It's about picking the right tool for the job: a high-speed engine versus a complete database system.