



Group 7

ECHODB

AN INTELLIGENT DATABASE CHAT AGENT

"ACCESS YOUR DATA WITH EASE"

BY PREET, JIVIN & RAM

PROBLEM STATEMENT

Valuable insights are locked behind technical barriers.

01

COMPLEXITY

SQL (Structured Query Language) is powerful but requires specialized knowledge.

02

BARRIER TO ENTRY

Non-technical stakeholders (managers, marketers) cannot easily access data.

03

BOTTLENECKS

Data analysts become bottlenecks for simple queries.

GOAL: “DEMOCRATIZE DATA ACCESS BY ALLOWING USERS TO QUERY DATABASES USING NATURAL LANGUAGE.”

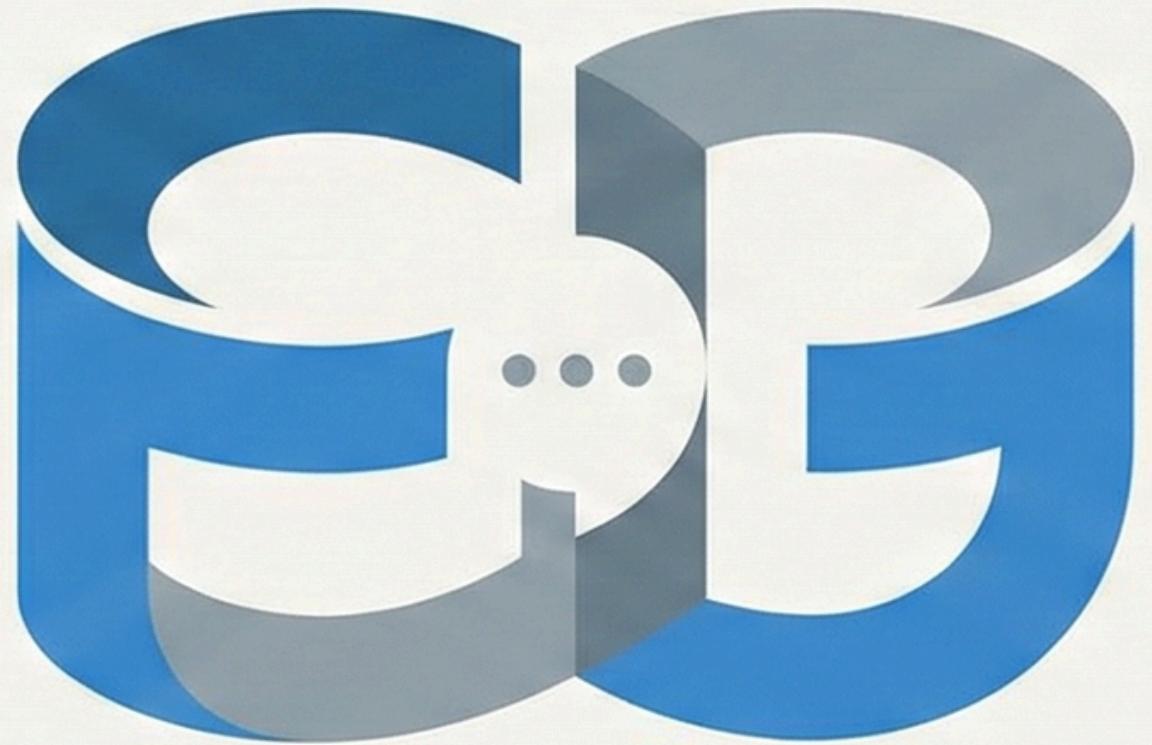
THE SOLUTION

WHAT IS ECHODB?

- An AI-powered agent that translates natural language into SQL queries.

HOW IT DIFFERS FROM “TEXT-TO-SQL”?

- It is a Model Context Protocol (MCP) Server that exposes database operations as "tools" to AI agents.
- **Active Agent:** It doesn't just generate code; it executes it.
- **Context Aware:** It actively reads the database schema to understand relationships before querying.
- **Platform Agnostic:** Works with any MCP client (e.g., Claude Desktop, IDEs). (we built our own)



KEY FEATURES

01

SECURITY FIRST

SYSTEM PROMPT FOR STRICTLY
READ-ONLY OPERATIONS.
REGEX VALIDATORS BLOCK
COMMANDS.

03

CONTEXT AWARE

CHECKPOINTER SHORT-TERM
MEMORY TO REMEMBER PREVIOUS
QUESTIONS AND CONTEXT WITHIN
A CONVERSATION THREAD.

02

ANY SQL DATABASE

SEAMLESSLY WORKS WITH
POSTGRESQL, MYSQL, AND
SQLITE.

04

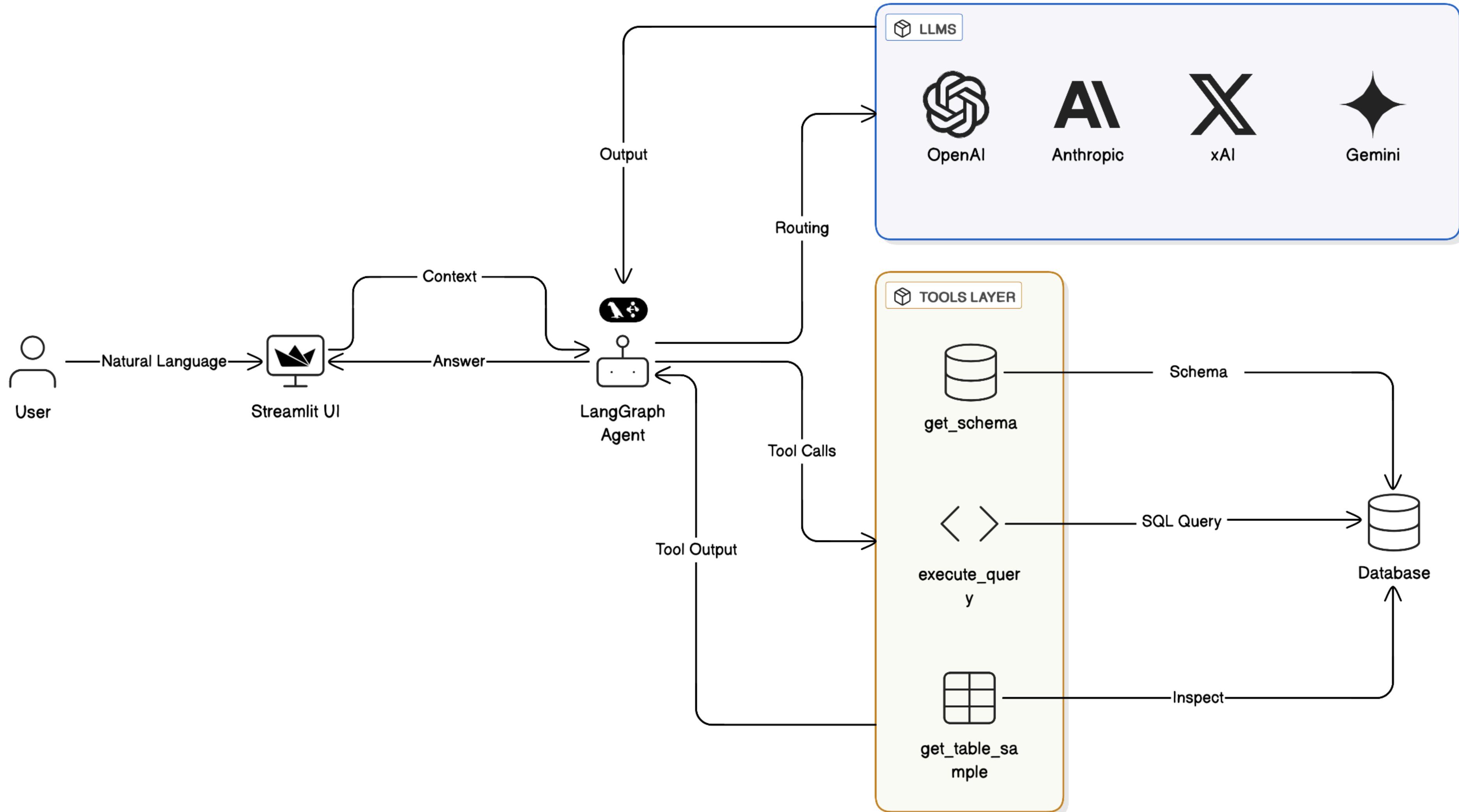
MULTIPLE MODAL PROVIDER

PROVIDING TOP-TIER MODELS VIA
OPENROUTER: **OPENAI GPT**,
ANTHROPIC CLAUDE, **GOOGLE**
GEMINI, AND **XAI GROK**.

SYSTEM ARCHITECTURE



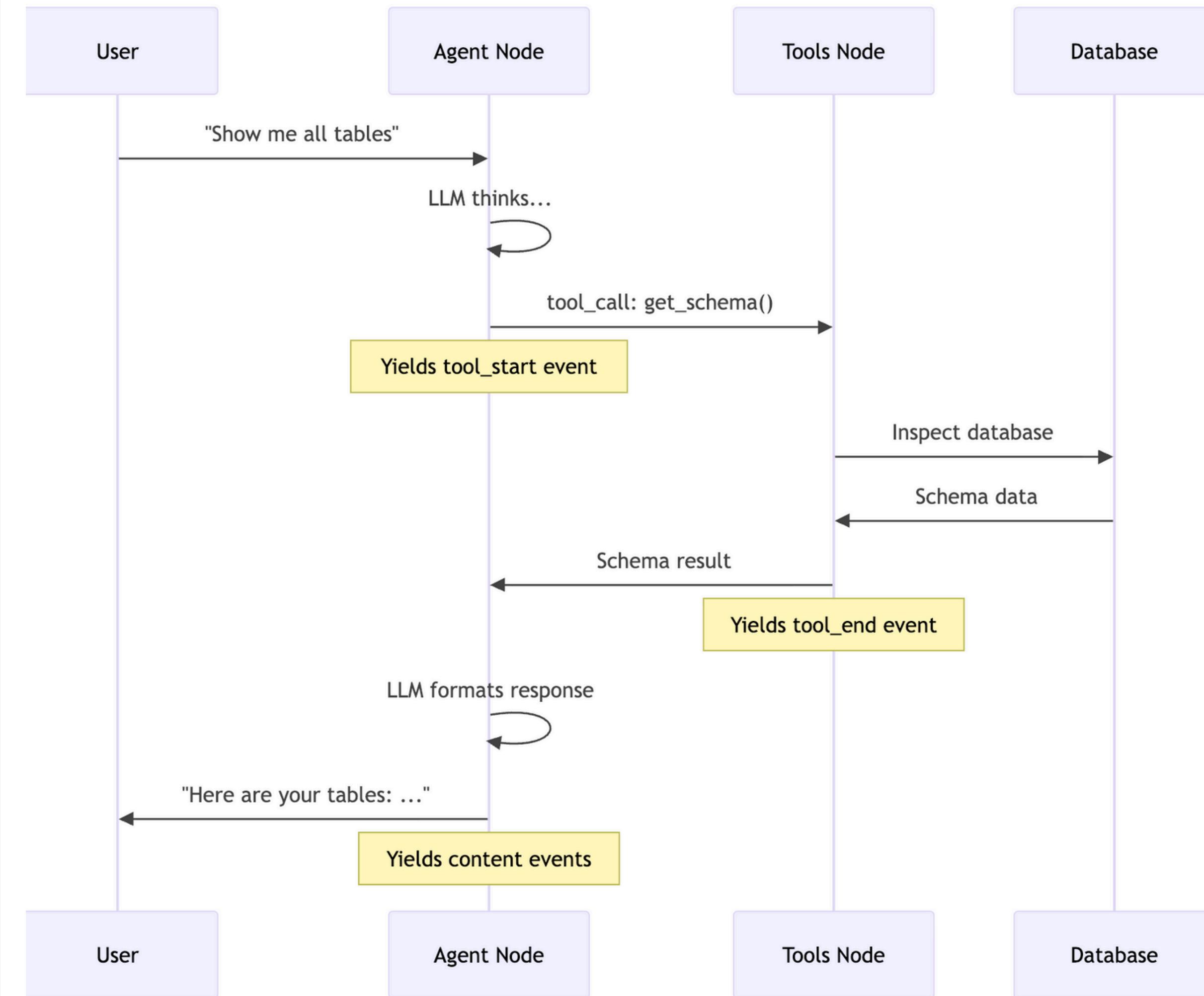
- 1. Client:** Acts as the user interface and the "Brain" (LLM) that plans the actions.
- 2. Communication Protocol:** MCP (Model Context Protocol) a standard JSON-RPC based protocol that allows the client to discover and call tools on the server.
- 3. EchoDB Server (Backend):**
 - Language: Python 3.11+
 - MCP SDK: Handles the protocol handshake and message routing.
 - ORM Layer: SQLAlchemy provides database-agnostic connectivity (Postgres, MySQL, SQLite).
 - Validation Layer: Custom logic to sanitize inputs and enforce read-only access.
- 4. Data Layer:** The actual database instance (local or cloud-hosted).



AGENT WORKFLOW



EchoDB



Configuration

Model Provider

Anthropic

Model Name

claude-4.5-sonnet

Database URI

postgresql://postgres:EHVa5l5

 Connected to DB!

Disconnect

Clear Chat

EchoDB - Database Chat Agent



Find me top 5 movies by sales from year 2017!!



I'll help you find the top 5 movies by sales from 2017! Let me start by understanding the database structure. Perfect! Now I can see the database structure. This is a movie rental database (like a DVD rental store). To find the top 5 movies by

✓ Tool: get_schema

Input:

▶ {}

Output:

```
{"payment_p2017_05": {"columns": [{"name": "payment_id", "type": "INTEGER", "primary_key": false, "nullable":
```

Ask a question about your database >

Configuration

Model Provider

Anthropic

Model Name

claude-4.5-sonnet

Database URI

postgresql://postgres:EHVa5l5

✓ Connected to DB!DisconnectClear Chat

Let me construct and execute this query: Perfect! Here are the **Top 5 Movies by Sales from 2017:**

Movie Title	Total Sales	Number of Rentals
SATURDAY LAMBS	\$319.58	42
TELEGRAPH VOYAGE	\$317.62	38
GOODFELLAS SALUTE	\$314.53	47
ZORRO ARK	\$311.56	44
HARRY IDAHO	\$308.53	47

Key Insights:

- **SATURDAY LAMBS** is the top-grossing movie with **\$319.58** in total sales from 42 rentals
- **GOODFELLAS SALUTE** and **HARRY IDAHO** had the most rentals (47 each) but slightly lower total sales, suggesting they may have had lower rental rates
- **TELEGRAPH VOYAGE** achieved the second-highest sales with fewer rentals (38), indicating it likely had a higher rental rate per transaction

The query combined all payment data from the 2017 partitioned tables (January through May) and traced each payment through the rental and inventory tables to identify which film was rented.

✓ Tool: get_schema

✓ Tool: execute_query

Ask a question about your database >

CHALLENGES

01

DIALECT NUANCES

Handling syntax differences between SQL dialects.

02

CONTEXT WINDOW LIMIT

Large database schemas or massive query results can overwhelm the LLM's context window

03

BOTTLENECKS

Data analysts become bottlenecks for simple queries.

FUTURE ROADMAP

01

- **Visual Analytics:** Add a `plot_data` tool using Python's `matplotlib`.
- Graph the revenue" -> Agent generates and executes plotting code.

02

- **Vector Search (RAG):** Combine SQL with semantic search for unstructured text columns.
- Adding Support for NoSQL databases.

03

- Implement a more robust "Guardrails" to validate generated SQL before execution.
- Add "Human-in-the-loop" approval for high-stakes queries.



Group 7

ANY QUESTIONS?

THANK YOU!

BY PREET, JIVIN & RAM