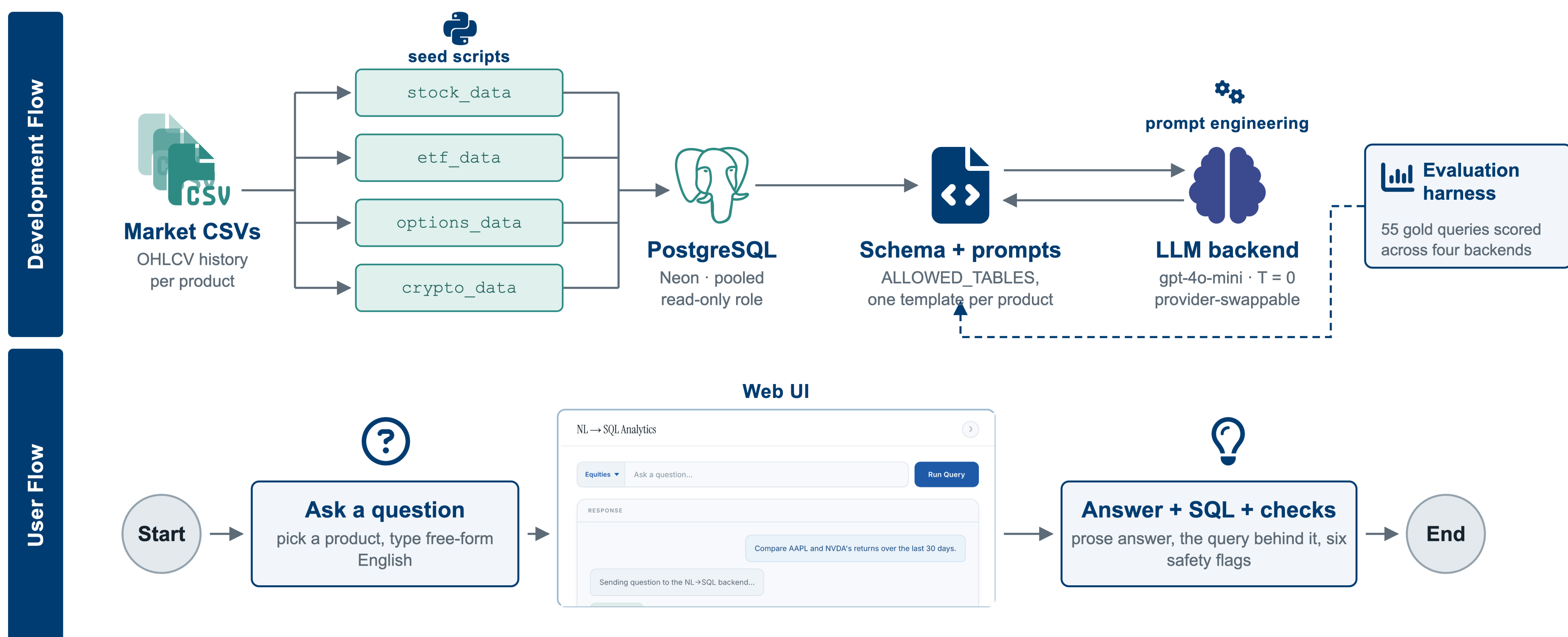


Introduction

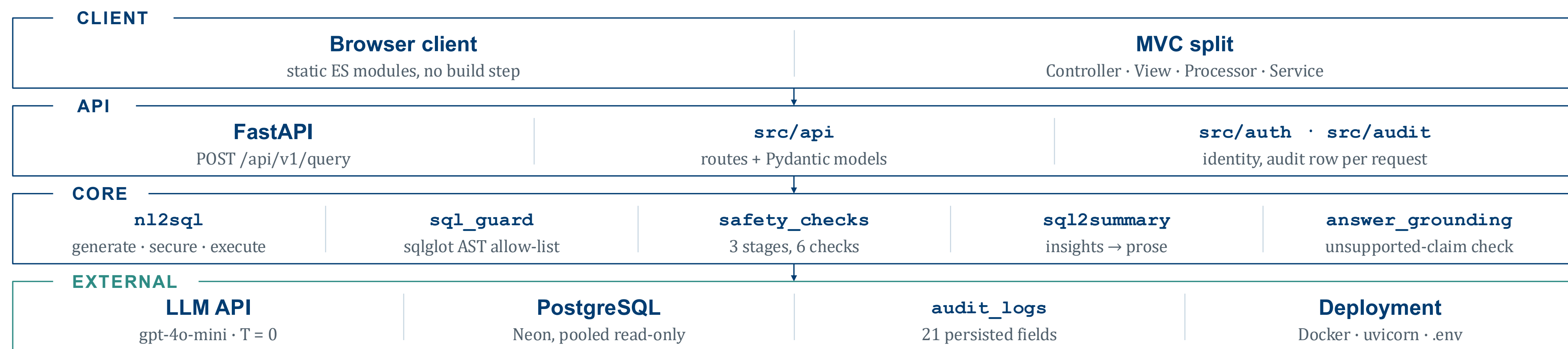
Natural-language interfaces make financial data more accessible, but syntactically valid SQL is not necessarily correct SQL. A model can misinterpret an ambiguous request, query the wrong business definition, or answer a question it should have refused. The most dangerous failure is therefore not a query that crashes, but one that runs and returns a plausible yet incorrect result. Our guarded NL2SQL pipeline addresses this risk through clarification, schema routing, SQL validation, read-only execution, answer grounding, and systematic evaluation.

Workflow



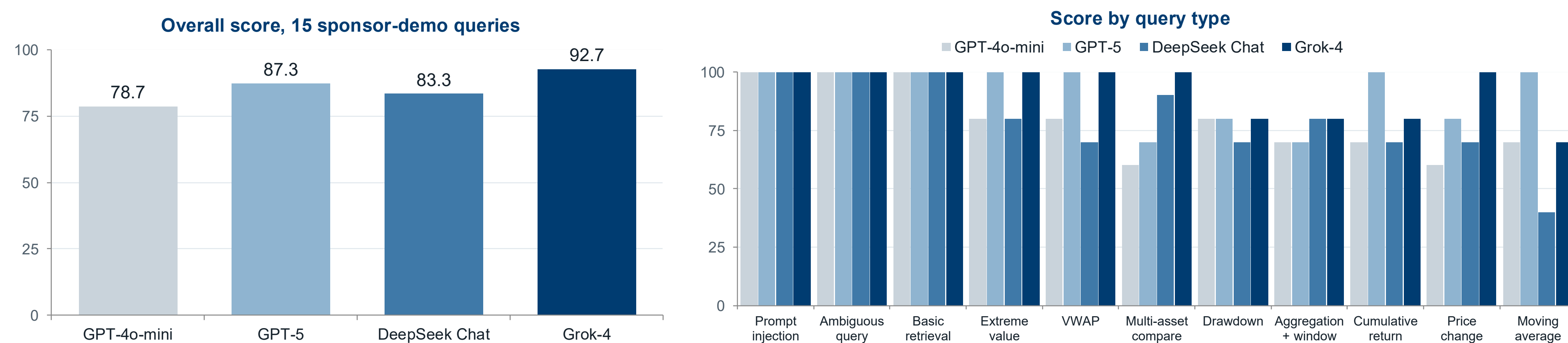
Platform Architecture

A static browser client over a FastAPI service; all query logic lives in src/core, reachable only through one endpoint.



Evaluation and Performance

Generated and gold queries are both executed and their normalised rows compared. Each response starts at 100 and loses 50 if injected SQL executes, 40 if it answers where it should have clarified, 30 if the rows differ from gold, and 30 if the summary claims a trend the rows cannot support.



Pipeline Logic

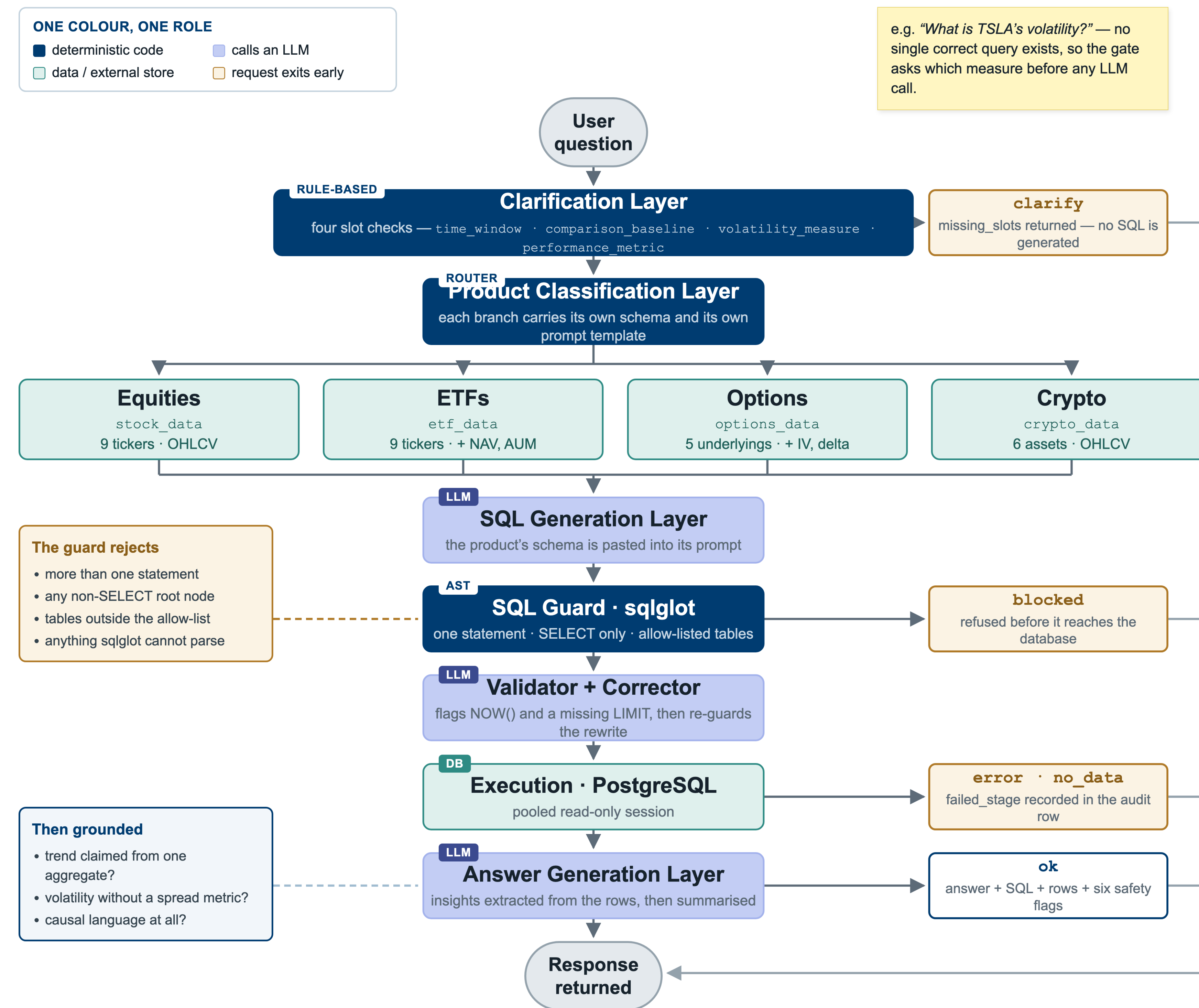
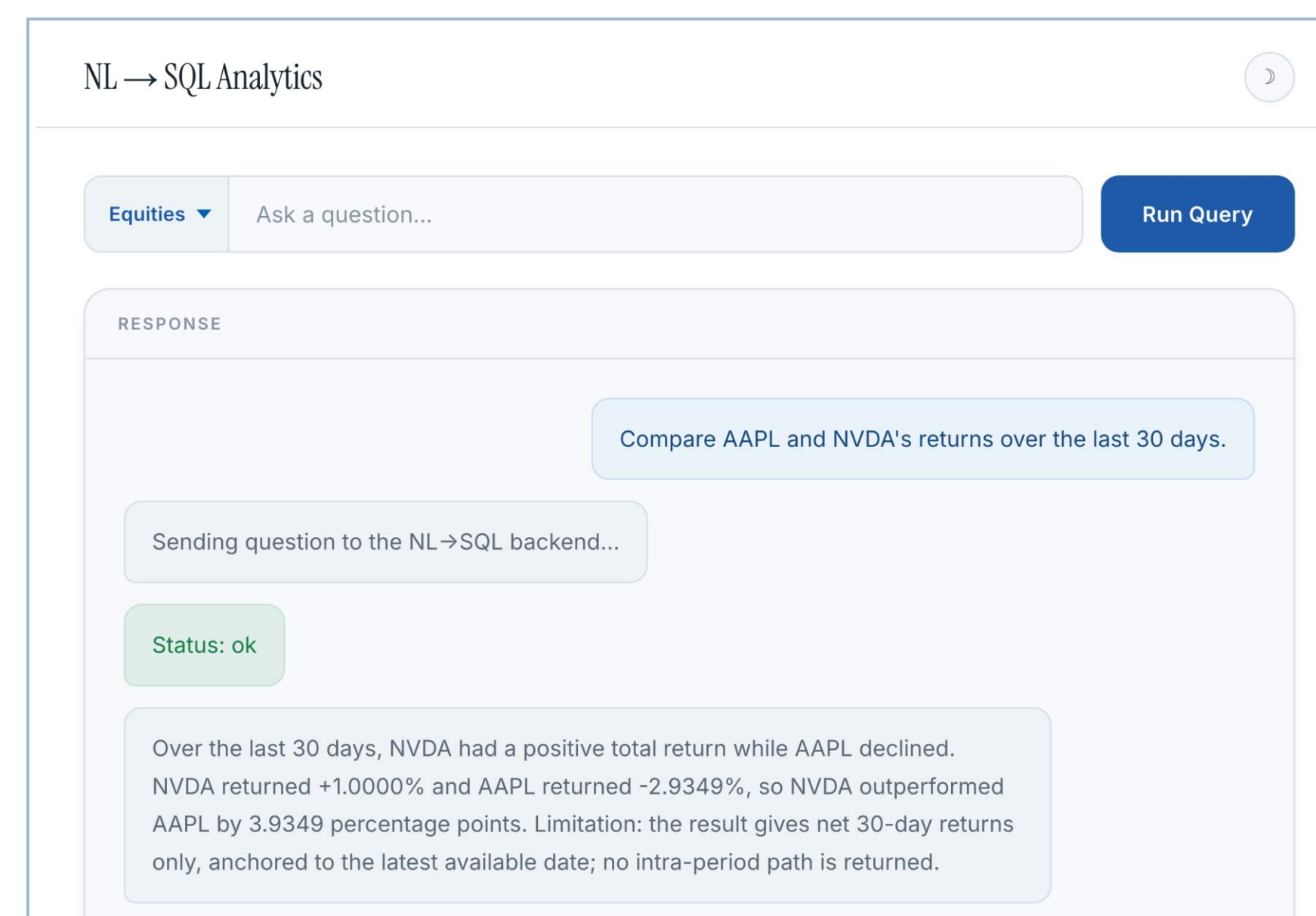


Figure 1. The router fans the question into one of four product schemas; the branches converge on a single guarded execution chain.

Live Interface



Q "Compare AAPL and NVDA's returns over the last 30 days."
A NVDA +1.00%, AAPL -2.93% — NVDA outperformed by 3.93 pp.

This example compares the 30-day returns of AAPL and NVDA. Alongside the final response, the interface exposes the generated SQL, raw database rows, and six safety checks. Analysts can therefore verify how each number was produced instead of treating the model's answer as a black box.

Conclusion

Our evaluation shows that guardrails prevented unsafe SQL execution across all tested models, but semantic consistency remained a challenge. Model scores and outputs varied depending on how consistently each model followed the expected query logic and business definitions. Future work will focus on improving output stability across models.