Executive One-Pager: LiteLLM + Langfuse + Phoenix + Label Studio

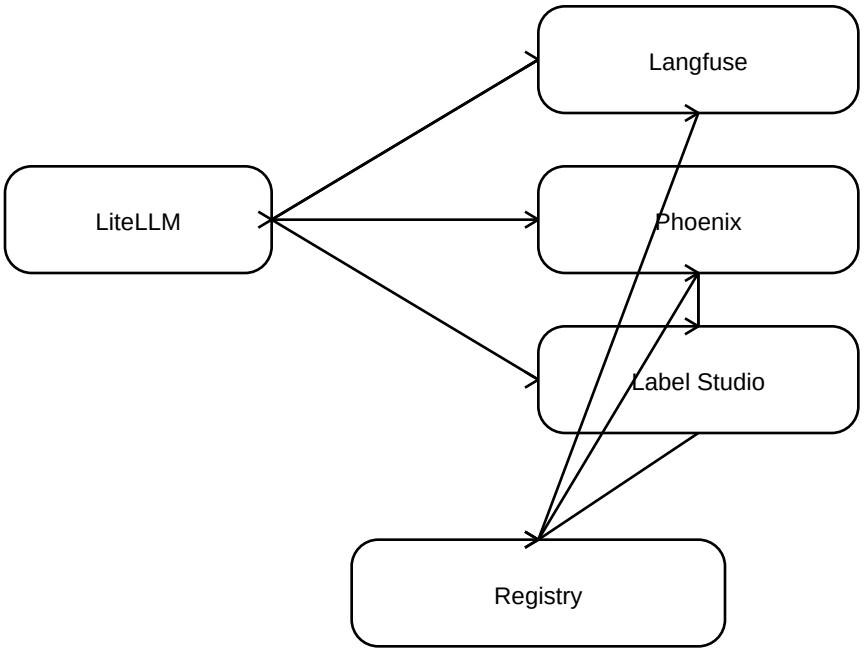
Executive Summary

This architecture provides a closed-loop system to build, monitor, and continuously improve LLM applications. LiteLLM serves inference; Langfuse provides real-time production observability and prompt versioning; Phoenix delivers evaluation and embedding/RAG diagnostics; and Label Studio enables human-in-the-loop annotations. Together, they enable fast iteration with governance-ready artifacts (optionally logged to a registry like MLflow).

Recommended Workflow

1) Observe (Production): LiteLLM routes all requests and sends traces to Langfuse for latency, cost, errors, and prompt version linkage.
2) Diagnose (Automatic Evals): Phoenix ingests logs/embeddings to score quality (e.g., relevance, hallucination) and detect embedding/RAG drift.
3) Human-in-the-Loop (HITL): Export problematic samples into Label Studio for structured human annotations (accuracy, tone, safety, relevance).
4) Improve: Feed annotations back into Phoenix as ground-truth; update prompts in Langfuse; (optionally) log datasets and results to a registry.
5) Deploy: Promote improved prompts/configs via Langfuse; continue monitoring with Langfuse while Phoenix tracks quality trends over time.

Data & Feedback Flow



Notes: Langfuse is the production console for latency/cost/tracing and prompt versioning. Phoenix provides automatic evaluation and embedding/RAG diagnostics. Label Studio is the HITL annotation layer. Optionally log datasets and

eval results to a registry (e.g., MLflow) for governance and lineage.