LangSmith Evaluation: Primitives
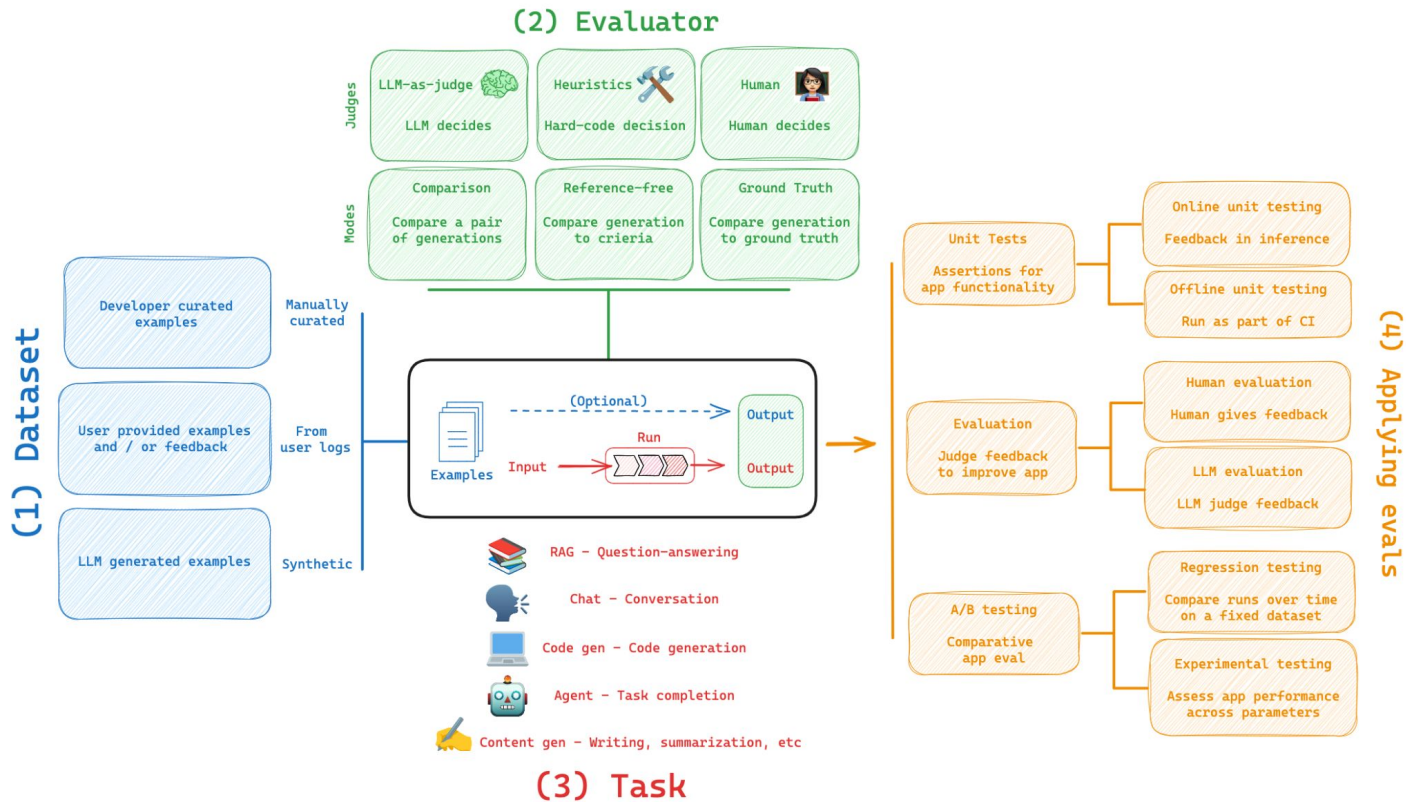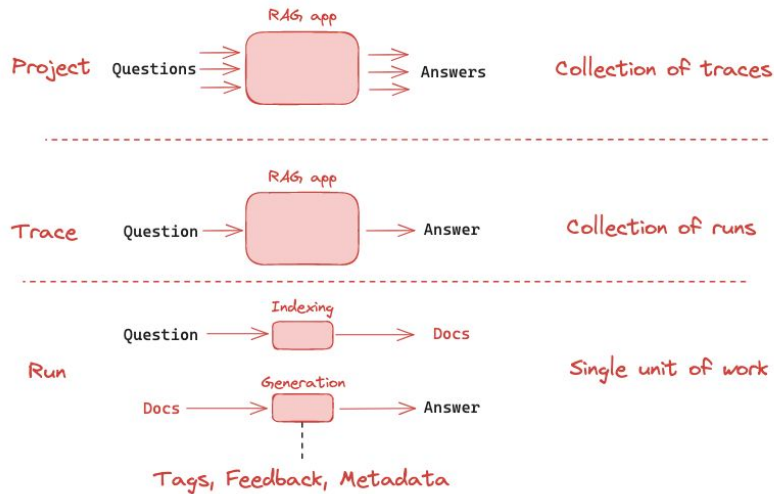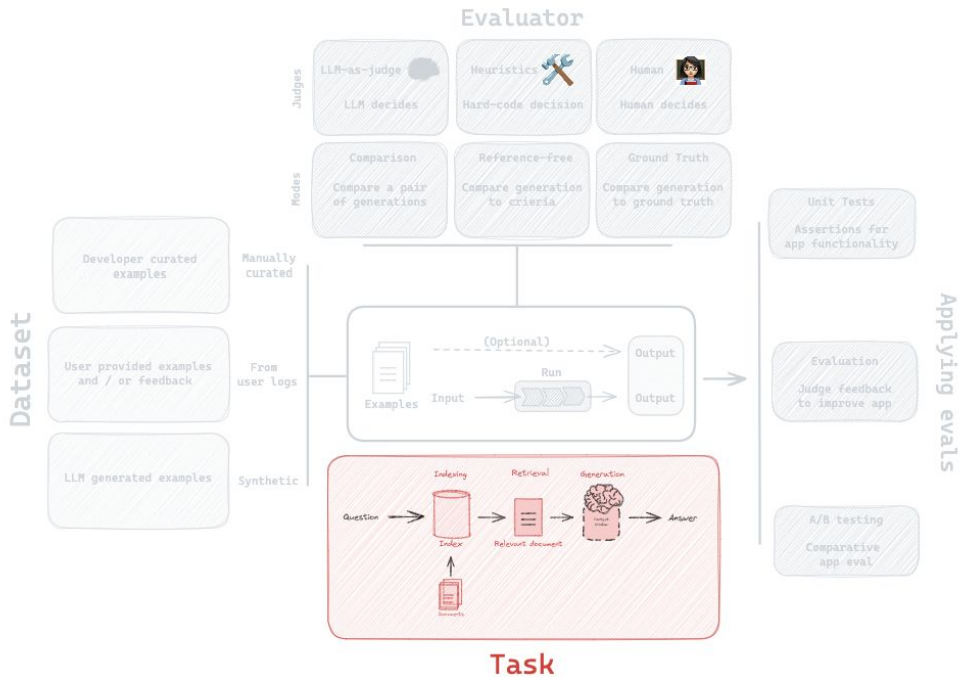
Lance Martin
Software Engineer, LangChain
@RLanceMartin

Building your own evals w/ LangSmith

(2) Evaluator

Judges

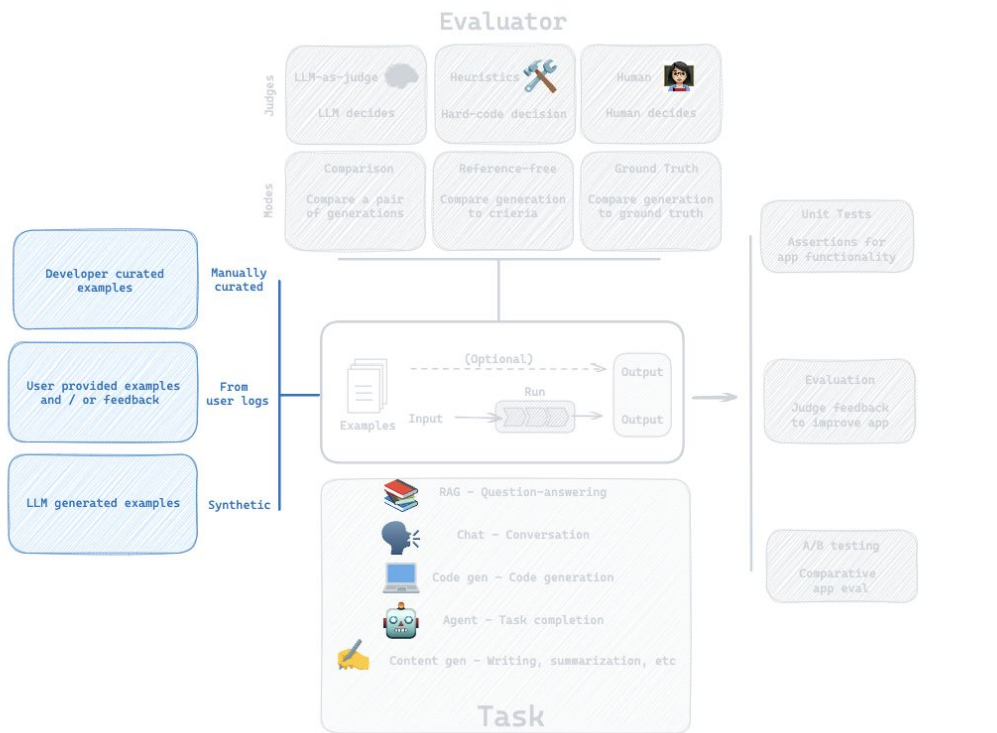| LLM-as-judge 🧠	Heuristics 🛠️	Human 👩🏽
LLM decides	Hard-code decision	Human decides

Modes

Comparison	Reference-free	Ground Truth
Compare a pair of generations	Compare generation to crieria	Compare generation to ground truth

(1) Dataset

Developer curated examples — Manually curated

User provided examples and / or feedback — From user logs

LLM generated examples — Synthetic

(Optional)

Examples   Input   Run   Output / Output

(3) Task

📚 RAG – Question-answering

🗣️ Chat – Conversation

💻 Code gen – Code generation

🤖 Agent – Task completion

🖌️ Content gen – Writing, summarization, etc

(4) Applying evals

Unit Tests — Assertions for app functionality
- Online unit testing — Feedback in inference
- Offline unit testing — Run as part of CI

Evaluation — Judge feedback to improve app
- Human evaluation — Human gives feedback
- LLM evaluation — LLM judge feedback

A/B testing — Comparative app eval
- Regression testing — Compare runs over time on a fixed dataset
- Experimental testing — Assess app performance across parameters

🦜🔗 LangChain

LLM app: Collection of runs (trace)

https://docs.smith.langchain.com/tracing/concepts
https://docs.smith.langchain.com/cookbook/tracing-examples/traceable#configuring-traces

Dataset: Collection of examples

Function to score LLM app on an example

https://docs.smith.langchain.com/tracing/concepts

Information flow

https://docs.smith.langchain.com/evaluation/faq/evaluator-implementations