A pragmatic approach to building generative AI applications

lastmile AI

Agenda

1. About Me
2. Not every problem needs a Gen AI solution
3. Hype- & jargon-free patterns for building AI applications
4. The Last Mile problem of Evaluation
5. Customary plug of LastMile AI
6. Q&A

About Me

What we do at LastMile AI

Team mission:

Enable **software engineers,** not just ML research scientists, to ship generative AI applications **with confidence in production**

About the speaker:

- CEO of LastMile AI
- Building developer tools my entire career – VS Code and build systems at Microsoft, Jupyter notebook platform at Meta
- Helped build AI infra for Meta's ML engineers and data scientists.

Sarmad Qadri

lastmile AI

Meta Microsoft

There is a lot of hype around generative AI

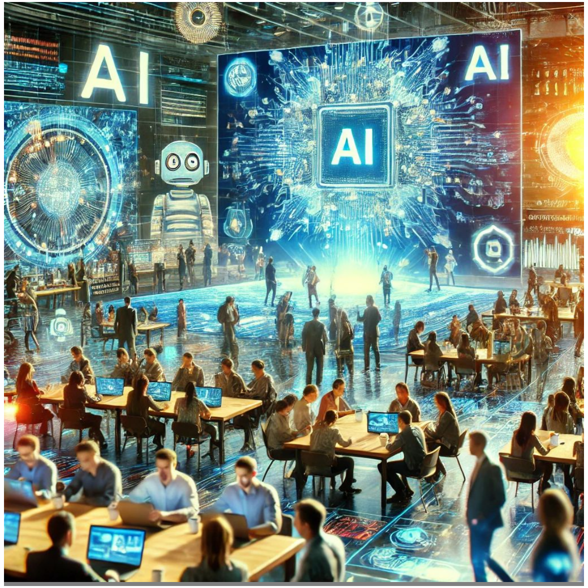
...but not every problem needs a Gen AI solution

AI innovation is accelerating:

- Models keep getting better and smaller
- Cost/token down 10x-100x

<u>But</u> we are at the risk of seeing every problem through the lens of Gen AI, whether it deserves to or not.

- **Not every problem is an AI problem**
  - Example: Chat interfaces everywhere
- **Not every AI problem is a Gen AI problem**
  - Example: RAG vs. IR or RecSys



Step-by-step guide to building AI applications That Work™

Hype- & jargon-free patterns that can help you scale

It takes days to build a prototype

Gen AI Prototype

- User Interface
- Prompts
- Hosted LLM

Messy development loop

But it takes months to make the prototype **production ready**

Production Ready Gen AI App

- User Interface
- Prompts
- Prompt Composition Logic
- Routing
- Evaluation/Testing Suite
- Encoding
- Context Retrieval
- Data Indexing
- Data Chunking
- Hosted LLM
- Local LLM

Step 1: Is this a hammer looking for a nail?

Will your problem benefit from an AI solution?

A. Clearly define the business problem you are trying to solve.

B. Identify the best implementation for your problem
   - *Example*: For math, use a calculator. For fixed patterns, a regexp will do.

C. If the simplest approach still requires a generative model, then proceed to Step 2.
   - *Example*: Intent recognition, natural language processing, information synthesis

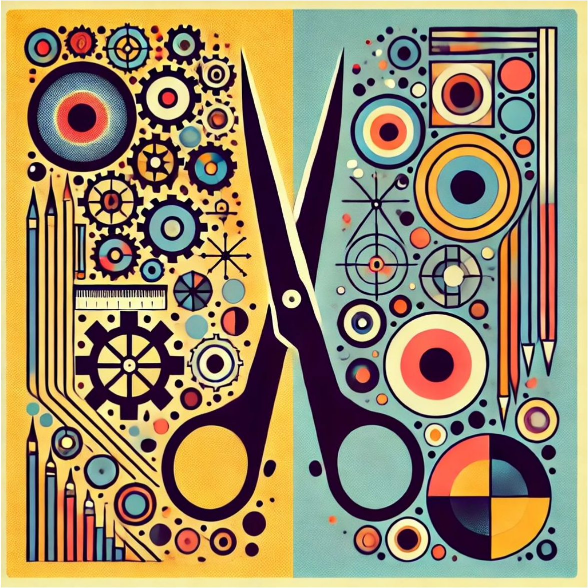
Step 2: Cut through the hype and jargon

There are a lot of distractions with the Shiny New Thing

Last year everyone was talking about vector databases, then RAG, then prompt optimization frameworks, and now we're talking about agentic workflows.

We don't need to reinvent everything:

- "LLM Observability" is just... observability.
- "Multi-agent workflows with memory" ~= Workflow orchestration with persistent state
- "Agent" ~= LLM with tool use (in vast majority of cases)
- "Prompt management" ~= just use source control



Step 3: Understand the limits of the current SOTA

Stay within the limits to ensure a robust application experience

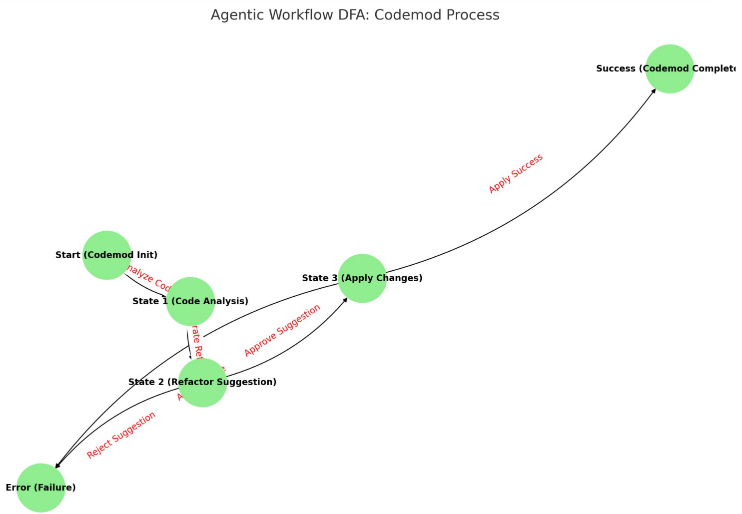
tl;dr:

- Really good for enhancing retrieval applications
- Not yet great for <u>unconstrained</u> agentic workflows

Tips:

A. *A*lmost every use case in enterprise boils down to information retrieval, extraction, synthesis.

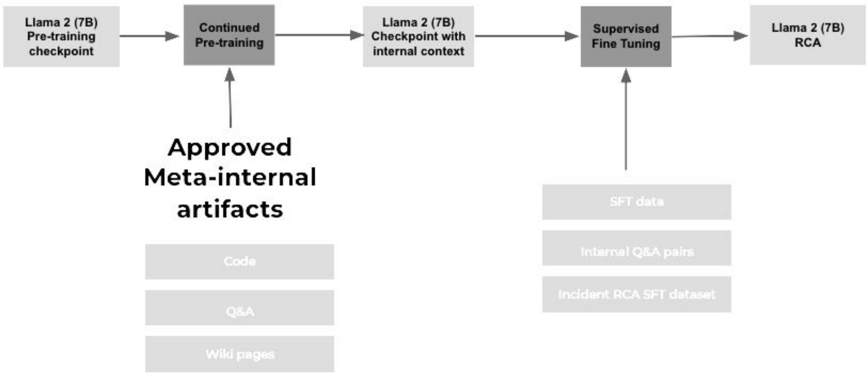
B. For agentic workflows, make it more deterministic by defining a state machine of interactions
   - *Example*: a codemod agent workflow can use its domain knowledge to define a state machine DFA



Step 4: Build the system!

Don’t forget about machine learning pre-ChatGPT

- Avoid unnecessary frameworks, except to prototype quickly.
  - As you iterate on the system, the abstractions often get in the way.
- For retrieval systems, don’t forget about IR research from pre-LLM days:
  - BERTopic for topic extraction
  - RBAC and data refresh/indexing
- Enhance experience with fine-tuned LLM’s and rerankers
  - Example: Incident response (Meta): 42% accuracy in incident RCA (root-cause analysis)



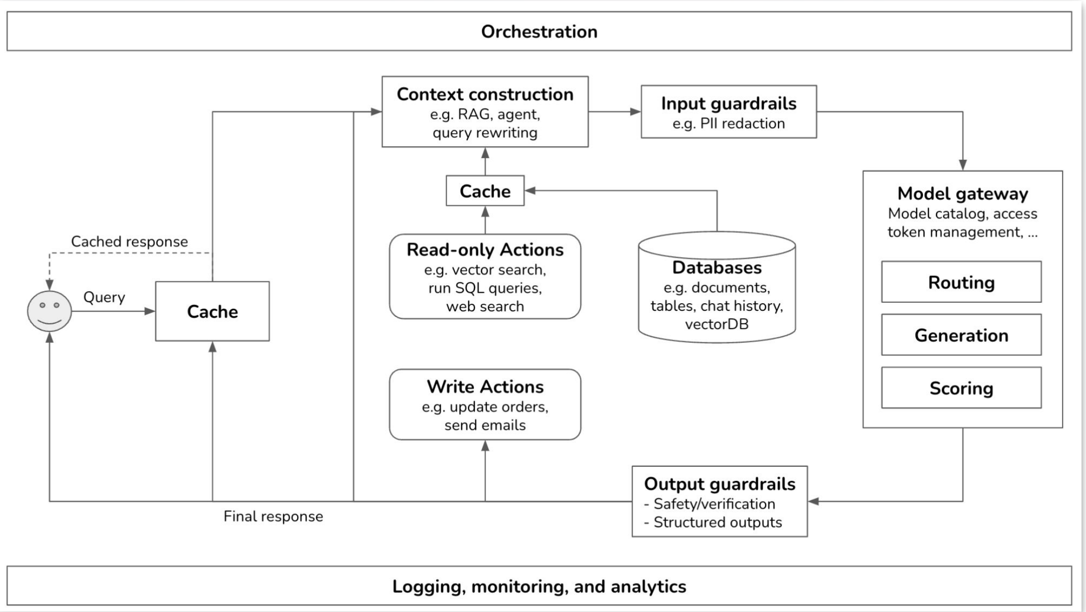
Step 5: Set up the harness *around* your application

Guardrails, monitoring & observability, etc.

Embrace the complexity, without overcomplicating things.

AI systems are distributed systems, and require a lot of the same primitives:

- Guardrails to constrain behavior **(more on this later)**
- Observability
- Feedback loop to improve the system (including data for fine-tuning)



Source: [Building a generative AI platform] (Chip Huyen)

Step 0: The Last Mile problem of Evaluation

Define how you are going to evaluate/measure performance

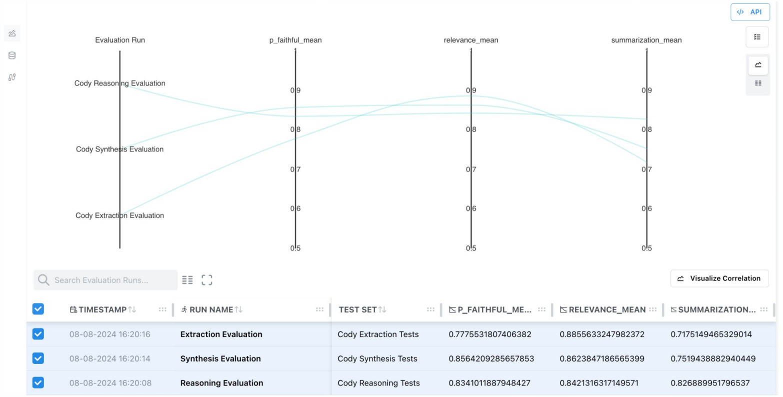
Evaluation: *how do I know my application is performing well?*

AI evaluation breaks the traditional SDLC:

- AI applications introduce non-determinism, which is different from standard integration tests/unit tests.

Current state-of-the-art for evaluating AI systems is LLM-as-a-judge. This approach is:

- Expensive
- Unreliable (LLMs weren't designed to be evaluators)
- Hard to customize for your specific application



LastMile SDLC – AutoEval

Custom evaluator models for **evaluation**, **testing** and **guardrails** of AI applications

AutoEval

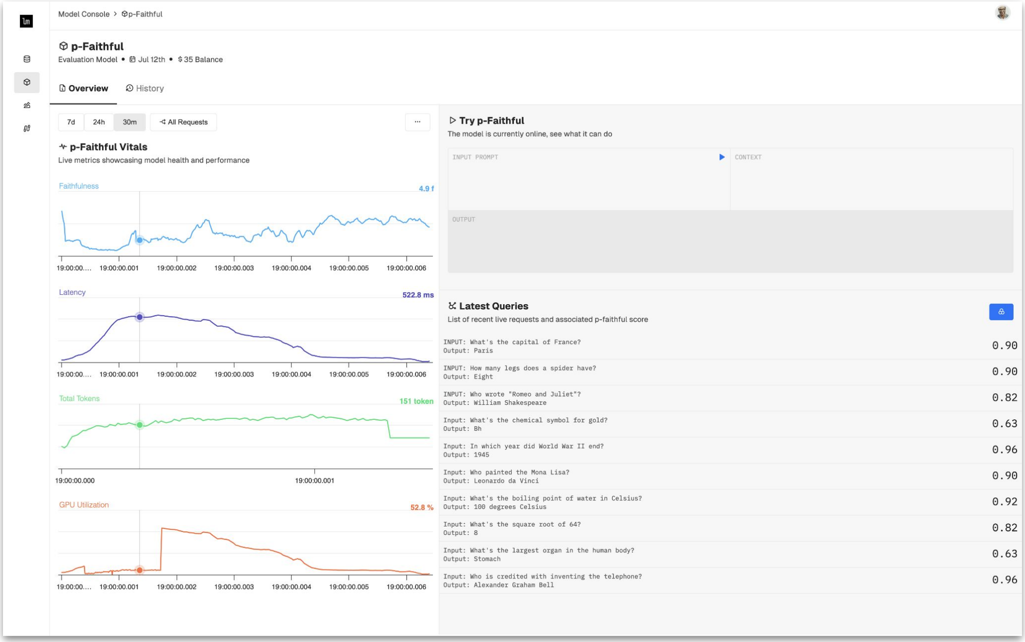
State-of-the-art evaluator models for evaluating and testing LLM & RAG applications. More performant and efficient than LLM-as-a-judge techniques.

	HaluEval	WikiEval
SotA Baseline	85%	85%
LM P(Faithful)	86%	98%

Supported Evaluators: *Faithfulness, Correctness, Toxicity, and Relevancy*

Fine-Tuning Evaluators: Cost-efficient enough to be fine-tuned to your business use case.

Guardrails: 100x faster than LLM-as-a-judge allows it to be used during online inference.

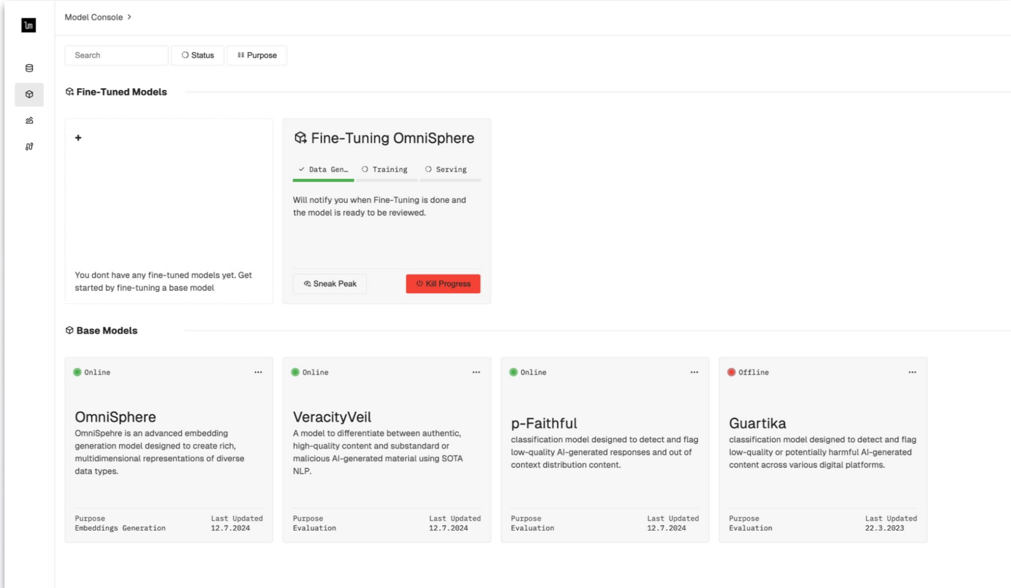


Fine-tuned to your application

AutoEval works well zero-shot, and can also be fine-tuned per application

Each model can be fine-tuned for your application, to get **customized metrics** and measure performance in the context of your task.

- Fine-tune with an API
- Manage your custom evaluators
- **Customize metrics specific to your application.**



Guardrails are just evaluators that run online

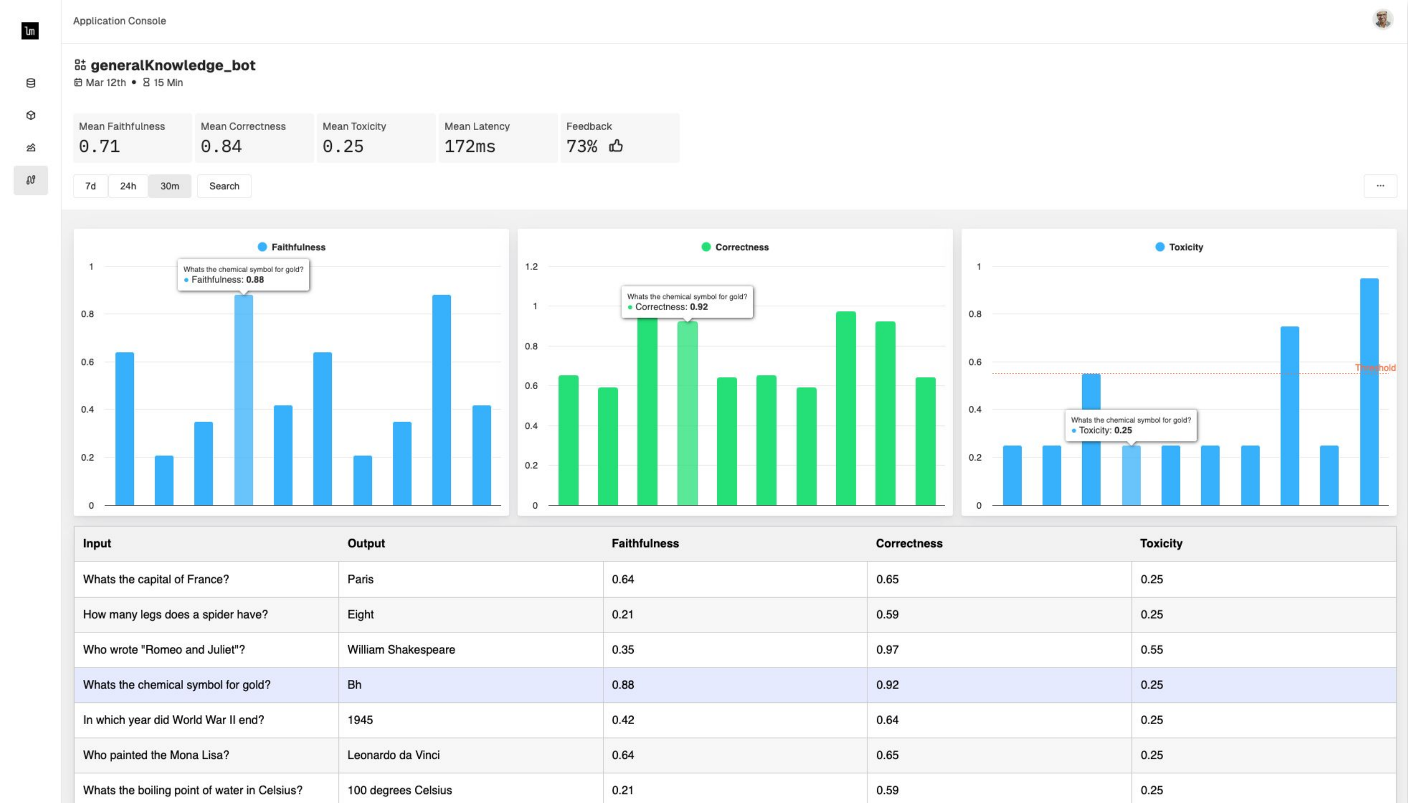
Evaluation and guardrails are 2 sides of the same coin

AutoEval models are fast (< 300ms on CPU), and cheap to operate (1/1000th the cost of GPT-4) → you can run them **on every response** as a guardrail.

- *Example*: use AutoEval to measure faithfulness, and if a response is deemed to have hallucinated, route to a backup response instead.
- You can also train custom guardrail models for specific safeguard policies using the same base model

LastMile AutoEval

Application performance dashboard



Thank you!

Get in touch with me about LastMile AutoEval

AutoEval Signup

email: sarmad@lastmileai.dev

any questions?