

Building Next-Generation Machine Learning Applications

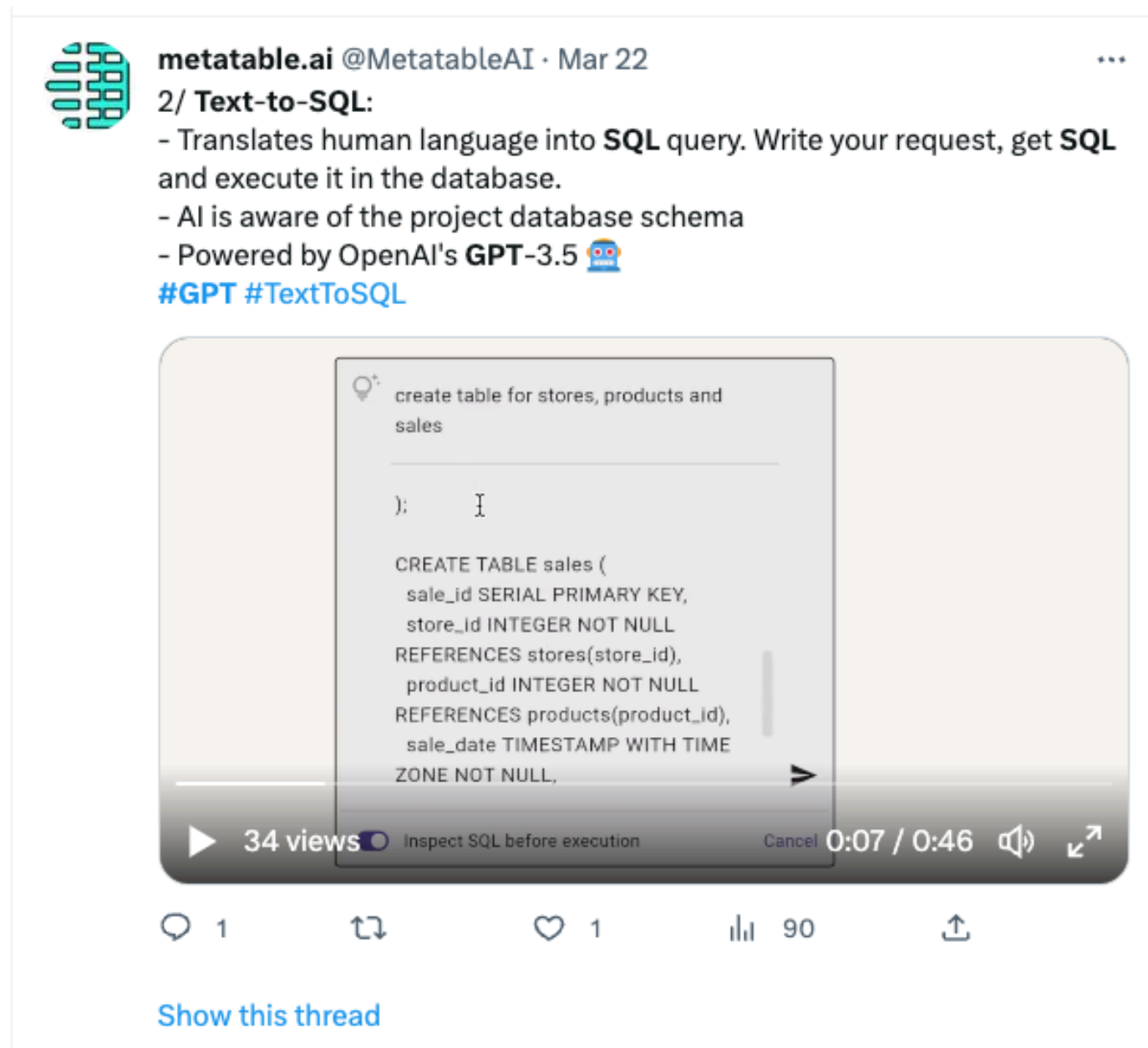
Shreya Shankar
April 2023

E P I C
D A T A

We're in a new era

10B parameters for everyone!

- It's now feasible to use ML models in software *without* lots of data and ML expertise
- The demos are unbelievable



metatable.ai @MetatableAI · Mar 22

2/ Text-to-SQL:
- Translates human language into SQL query. Write your request, get SQL and execute it in the database.
- AI is aware of the project database schema
- Powered by OpenAI's GPT-3.5 🤖
#GPT #TextToSQL

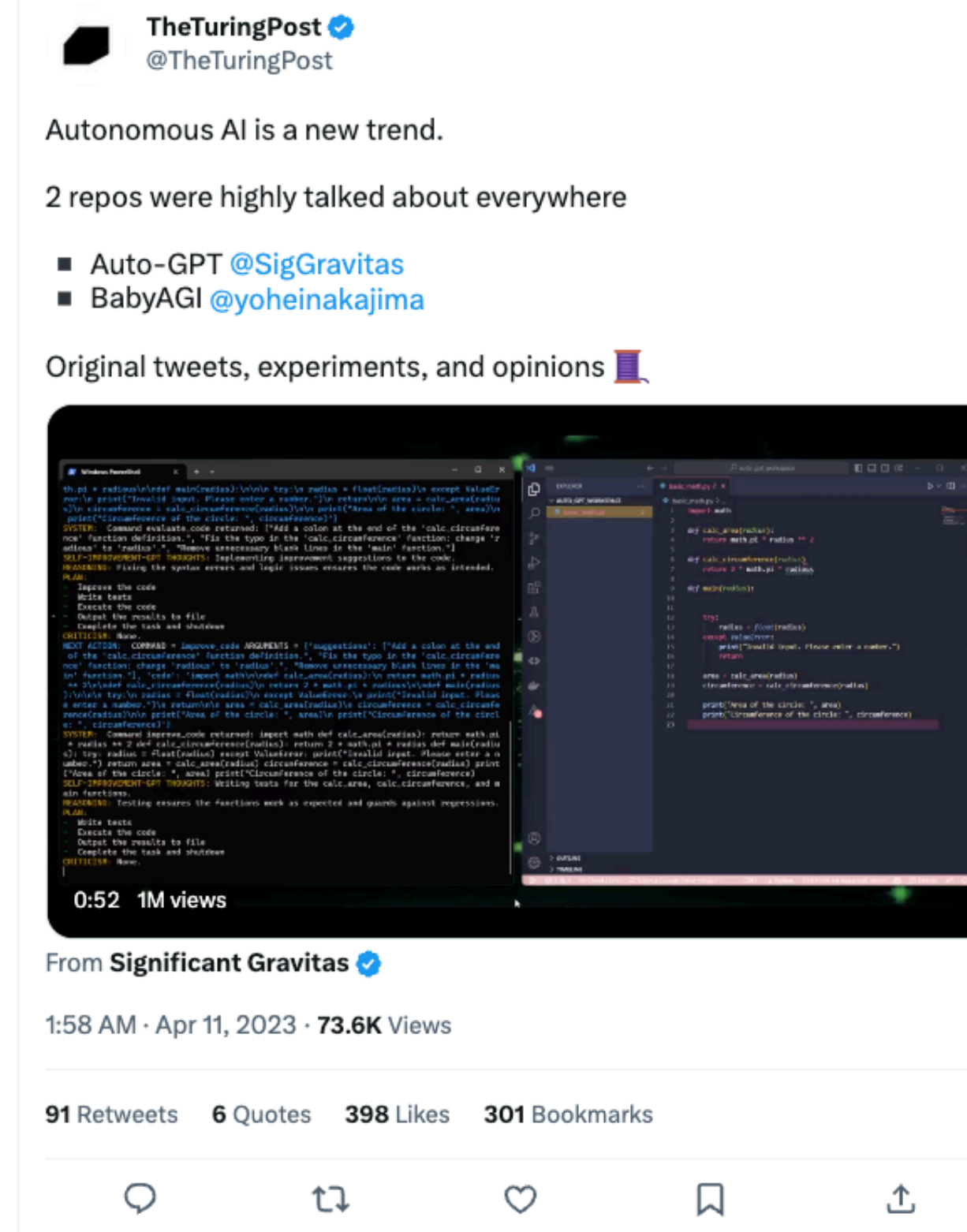
```
create table for stores, products and sales

);
```

```
CREATE TABLE sales (
  sale_id SERIAL PRIMARY KEY,
  store_id INTEGER NOT NULL
  REFERENCES stores(store_id),
  product_id INTEGER NOT NULL
  REFERENCES products(product_id),
  sale_date TIMESTAMP WITH TIME
  ZONE NOT NULL.
```

34 views Inspect SQL before execution Cancel 0:07 / 0:46

Show this thread

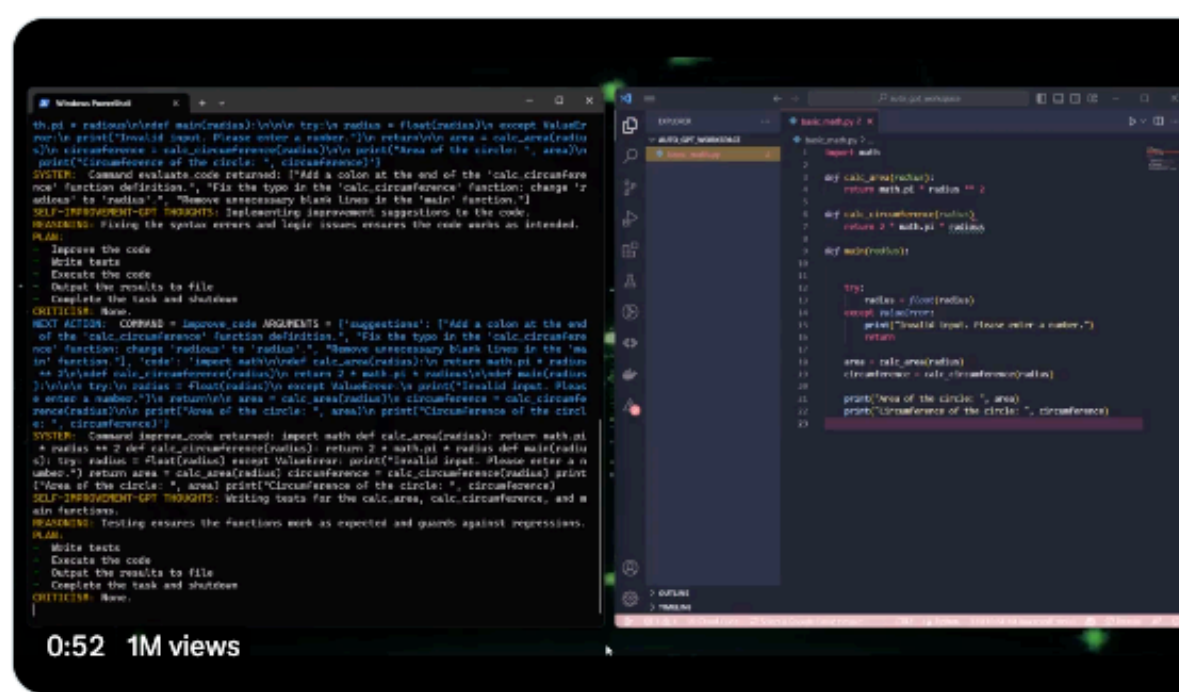


TheTuringPost @TheTuringPost

Autonomous AI is a new trend.
2 repos were highly talked about everywhere

- Auto-GPT @SigGravitas
- BabyAGI @yoheinakajima

Original tweets, experiments, and opinions 📖



0:52 1M views

From Significant Gravitas

1:58 AM · Apr 11, 2023 · 73.6K Views

91 Retweets 6 Quotes 398 Likes 301 Bookmarks



So many LLMOps tools out there

It's kinda overwhelming

- LLM frameworks
- Vector databases
- Prompt templates
- Deployment tools

The collage features several logos and promotional banners for LLMOps tools:

- LlamaIndex**: A blue banner with the text "LlamaIndex latest".
- LangChain**: A logo with a parrot icon and the text "LangChain 0.0.139".
- DUST**: A logo with the text "DUST".
- Helicone**: A logo with the text "Helicone Docs".
- Long-term Memory for AI**: A banner with the text "Long-term Memory for AI" and "The Pinecone vector database makes it easy to build high-performance vector search applications. Developer-friendly, fully managed, and". It also includes "Backed by Y Combinator" and "We're open source >".
- Observability for Generative AI**: A banner with the text "Observability for Generative AI".
- Humanloop**: A logo with the text "Humanloop" and a "Sign in" button.
- spellbook**: A logo with the text "scale spellbook".
- chroma**: A logo with the text "chroma" and "the AI-native open-source embedding database".
- Make GPT-4 faster, cheaper, more effective**: A banner with the text "Make GPT-4 faster, cheaper, more effective" and "Find prompts users love and fine-tune models for higher performance at lower cost".

LLM reality check

Demos aren't easily translating to production

- From survey on LLMs in production
- *We once used setfit for a production task but found that performance was really tricky to debug. **The proof of concept was fine. Production was a complete mess.** To this day I don't know why. Underlying hardware?*
- *We had a nice time selecting models, proving it would all work...then came some **dreadful times** making our vector lookup system work under load for long durations **in a proper prod environment** (where we couldn't touch it to do things like restart it or reload the index).*



Yohei @yoheinakajima · Apr 1

Taking a pause from adding complexity to the agent as it's harder to debug.

Took it an opposite direction and stripped out Langchain for now. It can't engage with APIs now, but can still perpetually build and execute tasks (just writing and "thinking").

```
what materials are most durable and long-lasting.
I will research the cost of production and determine what materials are affordable to use while still producing quality mittens. I will also research the manufacturing process and determine which production method is most efficient and cost effective in producing the mittens.

Finally, I will research the market for kitten mittens and determine the best pricing strategy for the mittens. This will include researching the potential customer base and understanding the competitive landscape in order to price the mittens accordingly.

****TASK LIST****
1: Develop a pricing strategy for kitten mittens
2: Establish distribution channels for kitten mittens
3: Develop a customer loyalty program for kitten mittens
4: Develop a customer service strategy for kitten mittens
5: Research legal requirements for selling kitten mittens
6: Create a marketing plan for a startup that sells kitten mittens
7: Research the most popular colors and styles for kitten mittens
8: Evaluate the potential customer base for kitten mittens

****NEXT TASK****
8: Evaluate the potential customer base for kitten mittens

****TASK RESULT****
My pricing strategy for kitten mittens will be based on a cost-plus approach. I will factor in the cost of materials, labor, and other expenses related to production, as well as a reasonable profit margin. Additionally, I will factor in market trends and competitor pricing to ensure that our pricing is competitive and attractive to customers.

I will also offer a variety of different price points for our kitten mittens based on the quality of the product, as well as any additional features or customization options offered. For example, higher quality mittens with more features or customization options will be priced at a premium, while basic mittens will be priced more affordably.

Finally, I will offer discounts and promotions to encourage customers to purchase our kitten mittens. This could include promotional codes for discounts, special offers for bulk purchases, or other creative offers.

0:16 | 41.5K views
```

36 41 354 154.3K



Teaching Robots about Insurance
@YourBuddyConner

Everyone eventually discovers you gotta rip out langchain to build a production LLM system it's inevitable

8:25 AM · Apr 2, 2023 · 1,596 Views

1 Quote 9 Likes



LLMs amplify existing MLOps challenges

Operationalizing Machine Learning: An Interview Study (Shankar and Garcia et al.)



Development environments are not production environments!



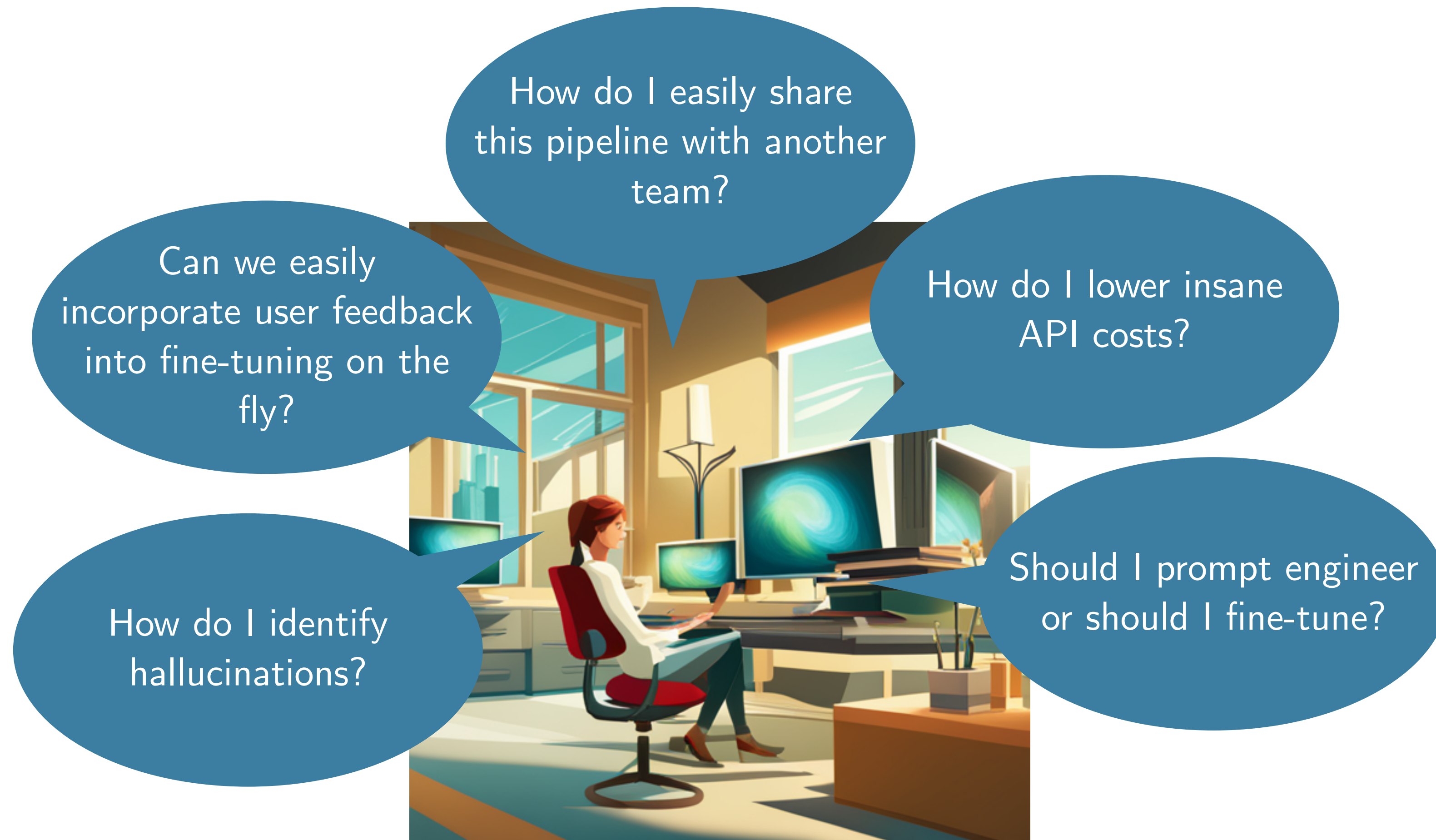
Too many services glued together



Hard to know when things go wrong without manually inspecting all outputs

Frequently asked questions

When making ML capabilities more accessible to non-ML people

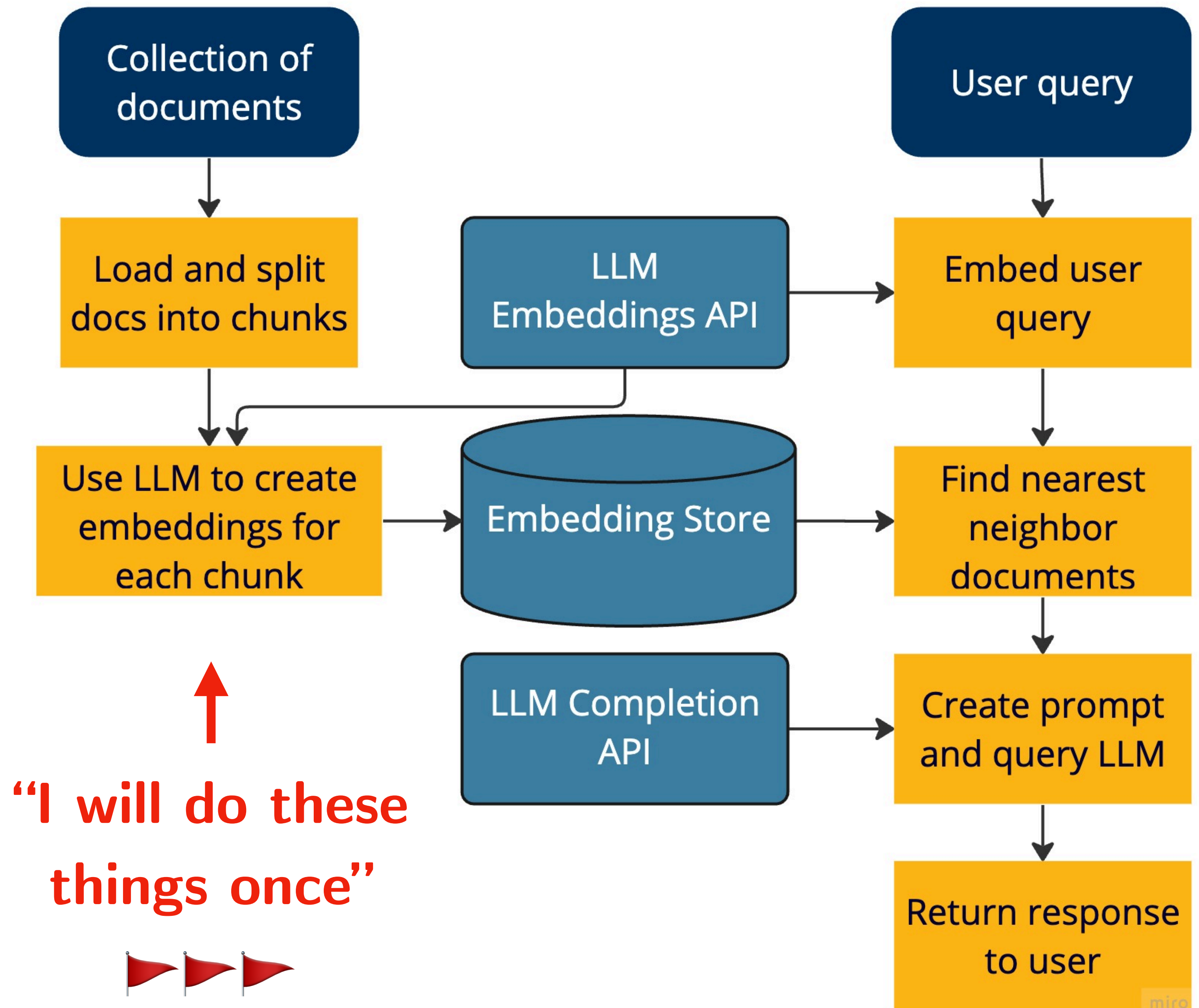


**What Does Building an ML
Application Look Like Now?**

Making a demo

Question-answering on docs

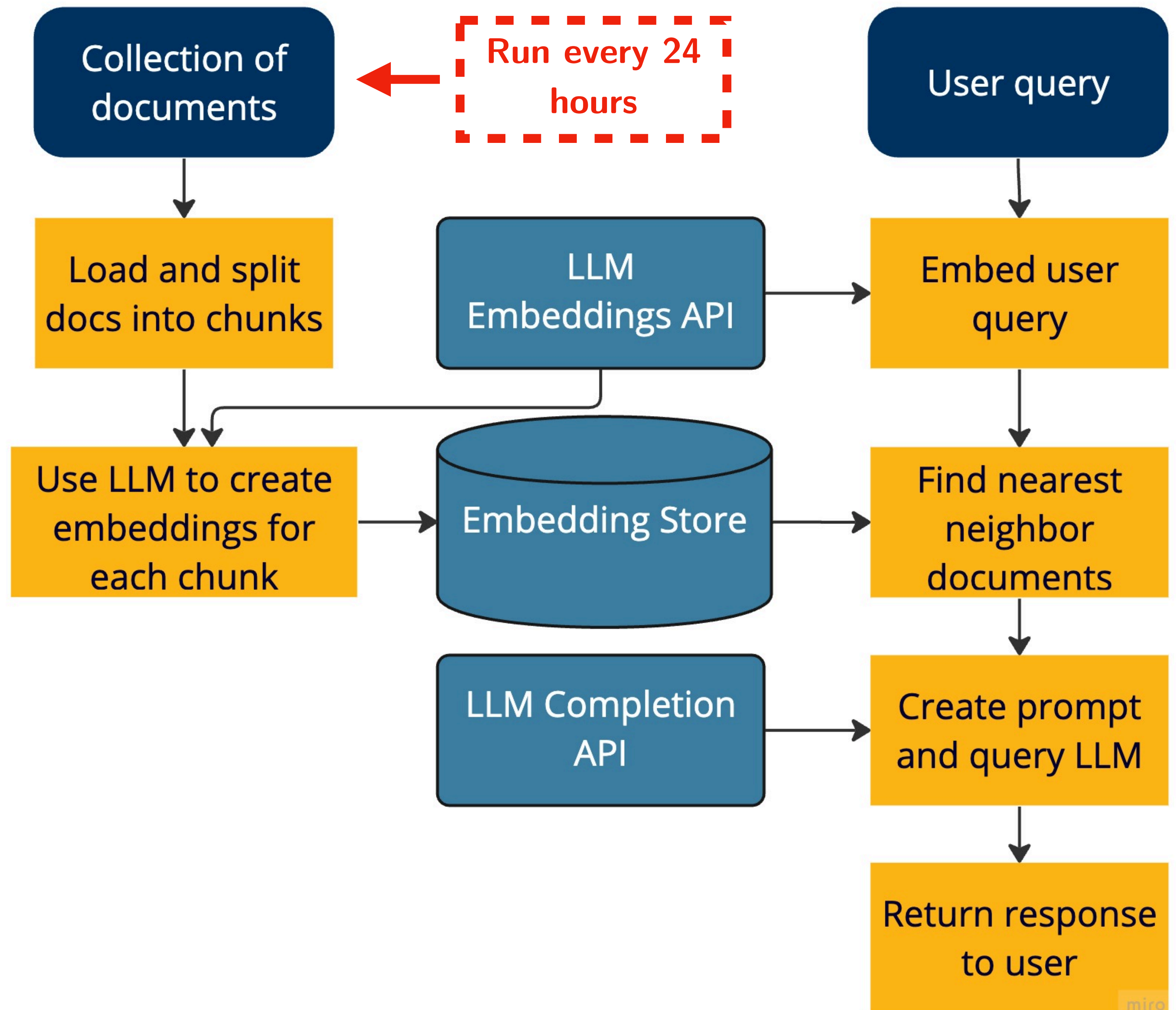
- From [LangChain](#) post



MLOps-ifying demos

Production isn't static!

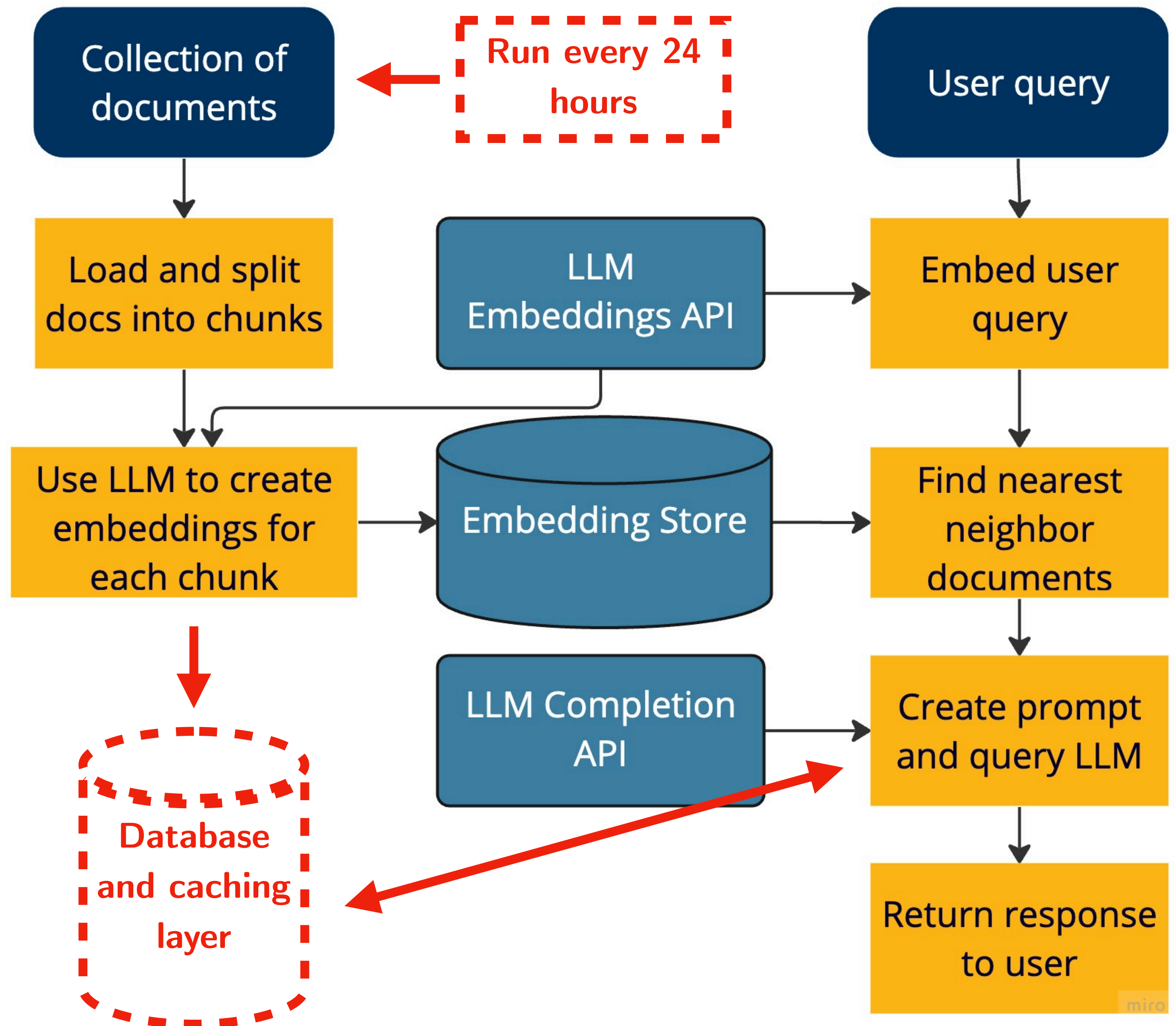
- What happens when there's a new document or wrong document?
- Maybe let's run the pipeline every day?
- Gotta set up a new machine or background job for this...



MLOps-ifying demos

Production isn't static!

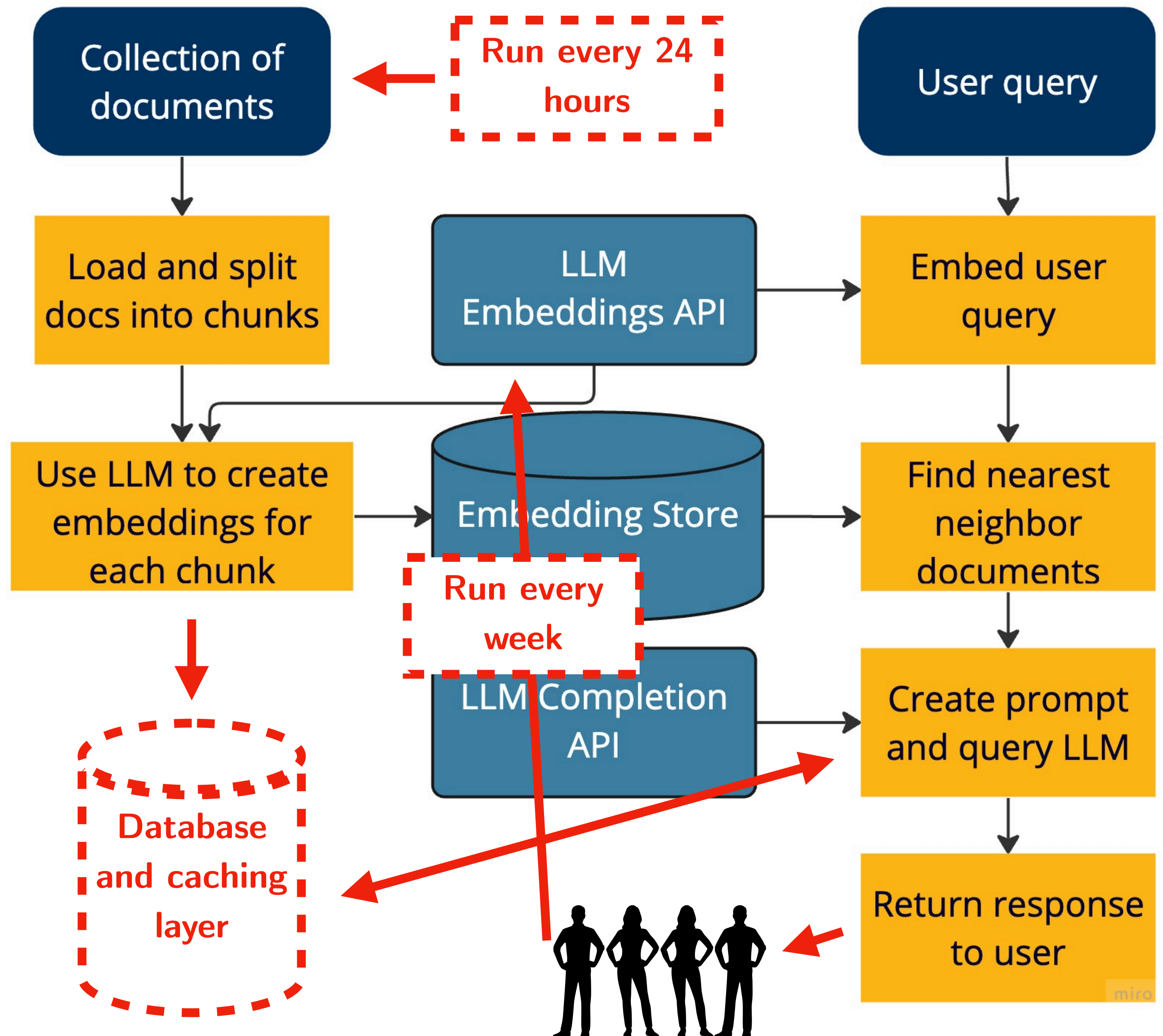
- What happens with a duplicate query?
- Maybe let's add a database to store all queries and responses
- Setting up and integrating a database 🙄



MLOps-ifying demos

Production isn't static!

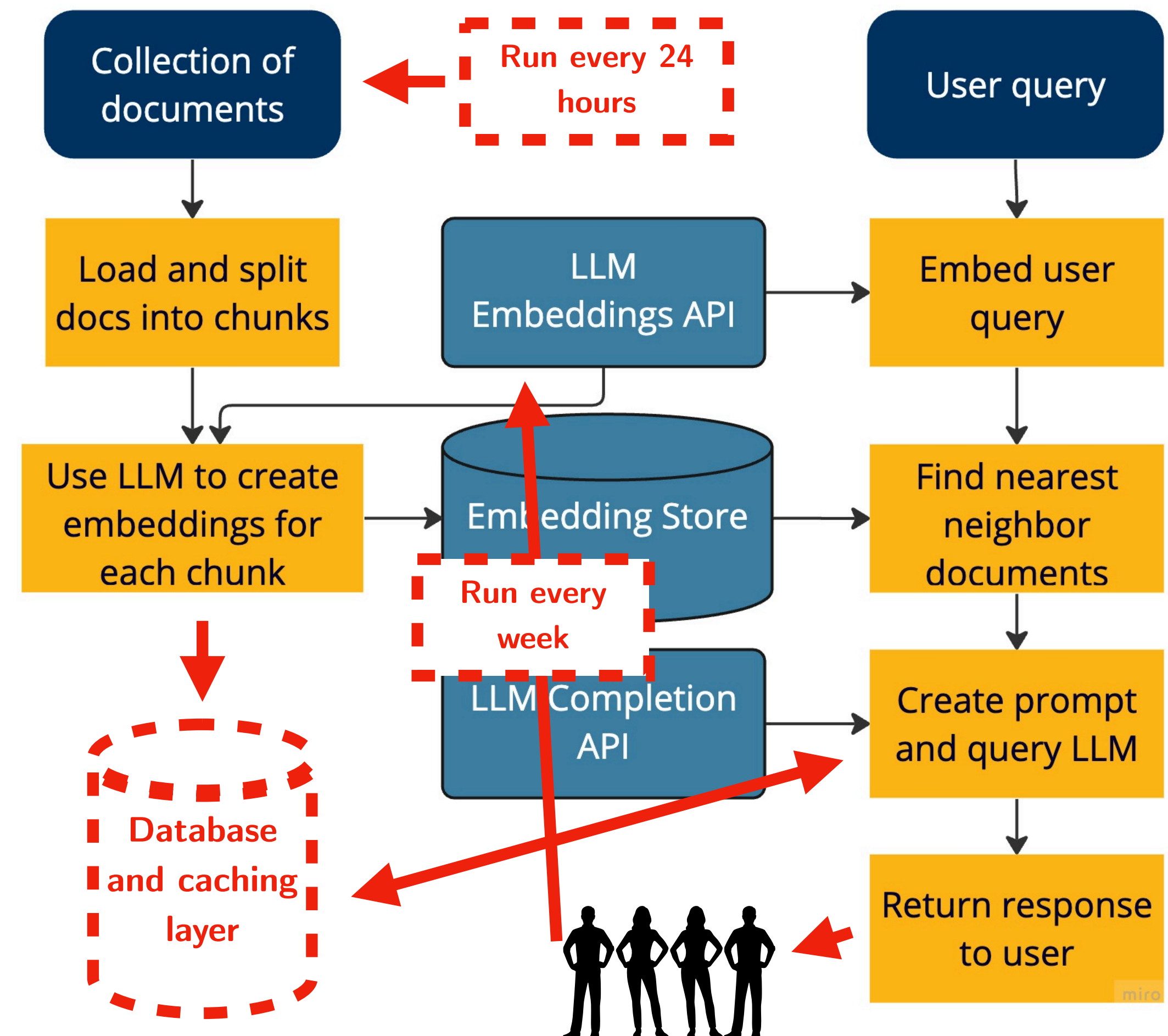
- How to incorporate user feedback (e.g., whether an answer is good)?
- Maybe let's have a team monitoring prompts and responses to select data to fine-tune on
- And then let's fine-tune the model every week!



Pipelines Galore

What could possibly go wrong?

- Each pipeline is being updated **independently** and in an **ad-hoc** way
- Wasteful redundancy and cost
- ML pipelines don't share state
 - No developer wants to get in on an existing complicated pipeline
- Experimentation almost never accounts for this wild setup
 - People are surprised to find performance drops in production!



Motion: Our ML Framework Under Development

Motion

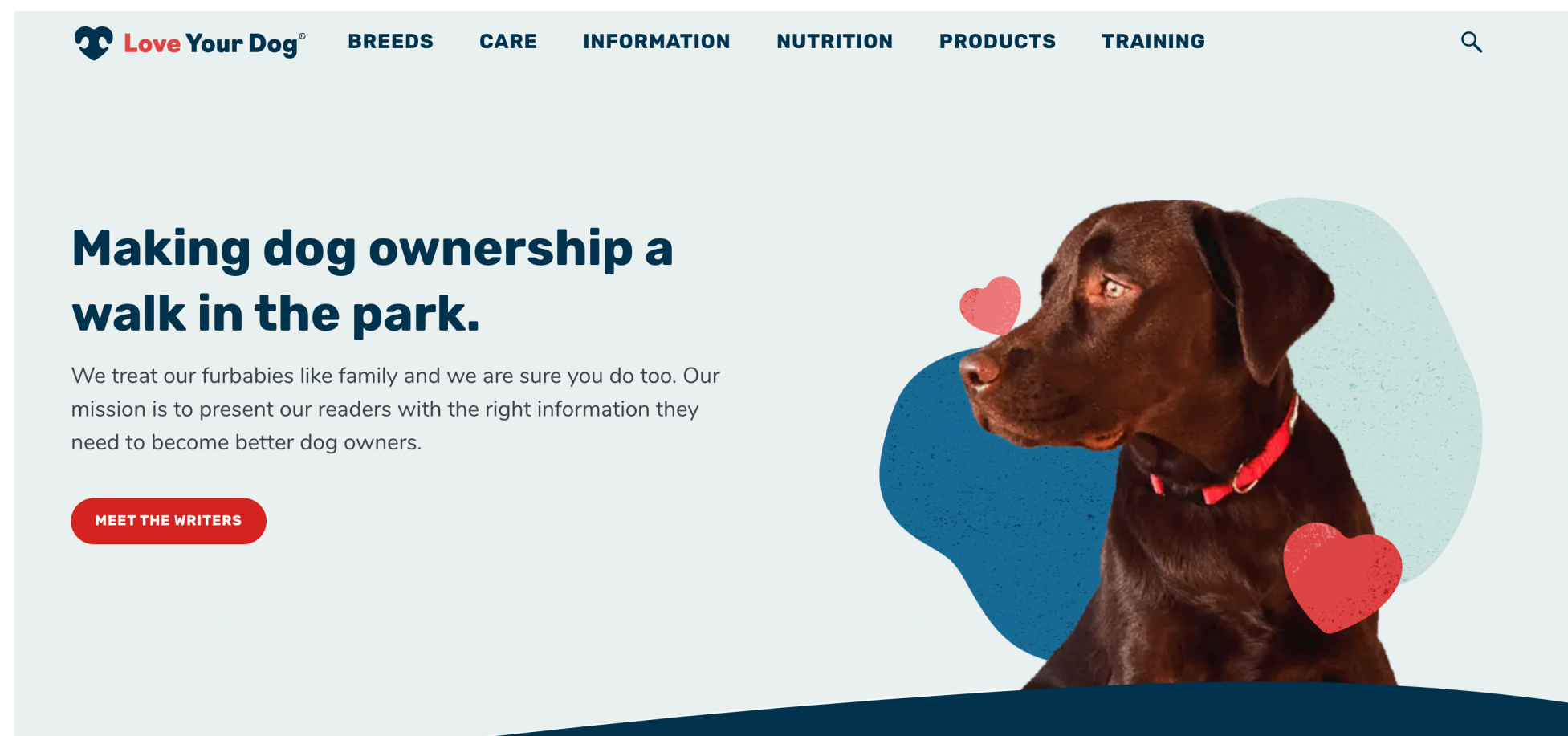
Yet another ML framework?

- A framework for building ML applications in Python with **continually-updating state**
 - Trigger stateful operations when adding data to a store
- Philosophical principles
 - **State** (e.g., models, vector indexes, prompt templates) **changes whenever there is new data**, often incrementally
 - Experimentation should consider these incremental updates
 - Multiple pipelines can benefit from shared state

Building with Motion

Chatbot Example 🤖

- We are getting a puppy this weekend! 🐶
- I would like a chatbot to ask my dog questions to...



<https://dm4ml.github.io/motion/>

Motion

A departure from the traditional workflow

Traditional Workflow	Motion Workflow
<ol style="list-style-type: none">1. Write script to scrape blog posts and save to disk2. Write script to load posts from disk, chunk and embed them, and save them to vector store3. Write script to load model, connect to embedding store, run a query, and return a response4. Deploy script 3 to some machine to run whenever there's a new query5. Change scripts 1 and 2 to use cloud storage6. Deploy scripts 1 and 2 to some other machine to run on schedule7. Change script 3 to log queries/responses to a DB8. Write and deploy a script to another machine to fine-tune on a schedule9. Change script 3 to read the latest model	<ol style="list-style-type: none">1. Define data relations with schemas2. Define triggers to run when data changes in a relation. Triggers have <code>setUp</code>, <code>infer</code> (foreground, state read-only), and <code>fit</code> (background, state writes-allowed) methods.3. Deploy!4. Add routes in triggers with <code>fit</code> methods to fine-tune on user feedback



Motion Demo

Building with Motion

A departure from the traditional workflow

	Traditional Workflow	Motion Workflow
Pre-Deployment	<p>Low upfront effort 🚀</p> <ul style="list-style-type: none">● Flexibility to look at and operate on full batches of data● No need to specify data and dependencies● No need to think about fine-tuning	<p>Higher upfront effort 🧑‍🔧</p> <ul style="list-style-type: none">● Must define schema● Must separate logic into state read-only and write-allowed (infer vs fit)
Post-Deployment	<p>High ops effort 😞</p> <ul style="list-style-type: none">● Need to rewrite existing pipelines when adding new functionality (e.g., ingesting new documents, fine-tuning)● Need to validate data and monitor for shift● Need to coordinate different jobs	<p>Low ops effort 💰</p> <ul style="list-style-type: none">● Can add new functionality without modifying existing pipeline code● Data is type-checked, validated, and monitored● All jobs done on one machine (unless explicitly outsourced in infer or fit methods)

Work in Progress

Improving Experimentation Support

- Inject parameters into the config and log runs with experiment trackers
- To prompt engineer or to fine tune?
 - Probably different for every task
 - Goal: allow users to easily answer this question in Motion (swap out code in *fit* methods)

Auto-refit based on data drift

- Some state only requires recomputation when data drifts (e.g., seasonally)
- Profile data within relations to check for drift
 - Compute summaries on daily or weekly partitions
 - Run anomaly detection on partition summaries
 - Moving Fast with Broken Data (Shankar et al.)



Looking ahead

- Field is moving lightning fast ⚡
- ML is becoming mature enough to have reusable triggers
 - Reach goal: build continual ML applications with natural language
- Still many task-specific challenges to solve
 - “What guardrails do I put on model outputs?”
 - “Should I put multimodal data in prompts? How?”

Thank you!

 shreyashankar@berkeley.edu