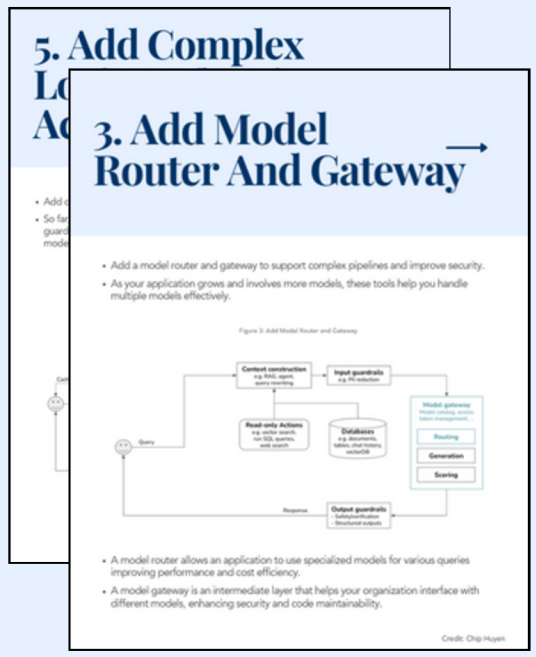
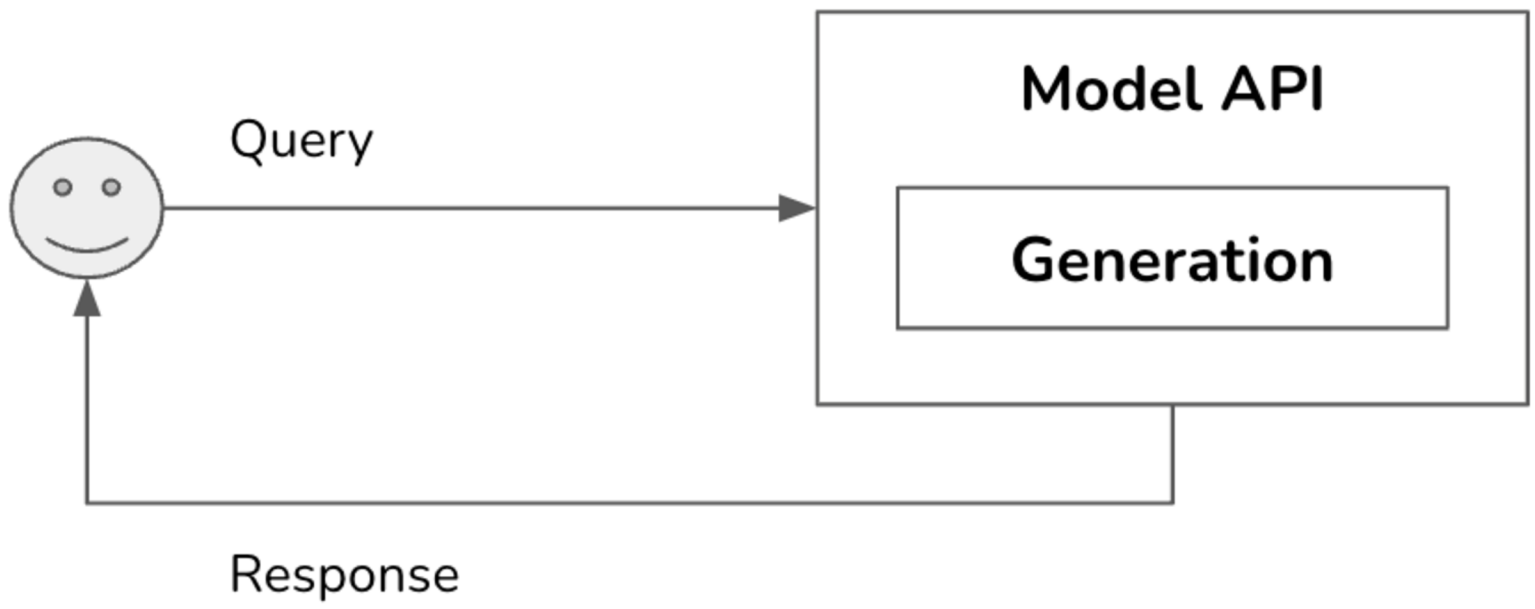
5 Steps To Build A Generative AI Platform



0. Basic Architecture

- In its simplest form, your application sends a query to the model, which generates a response and returns it to the user. There are no guardrails, augmented context or optimizations.

Figure 0: Basic Architecture



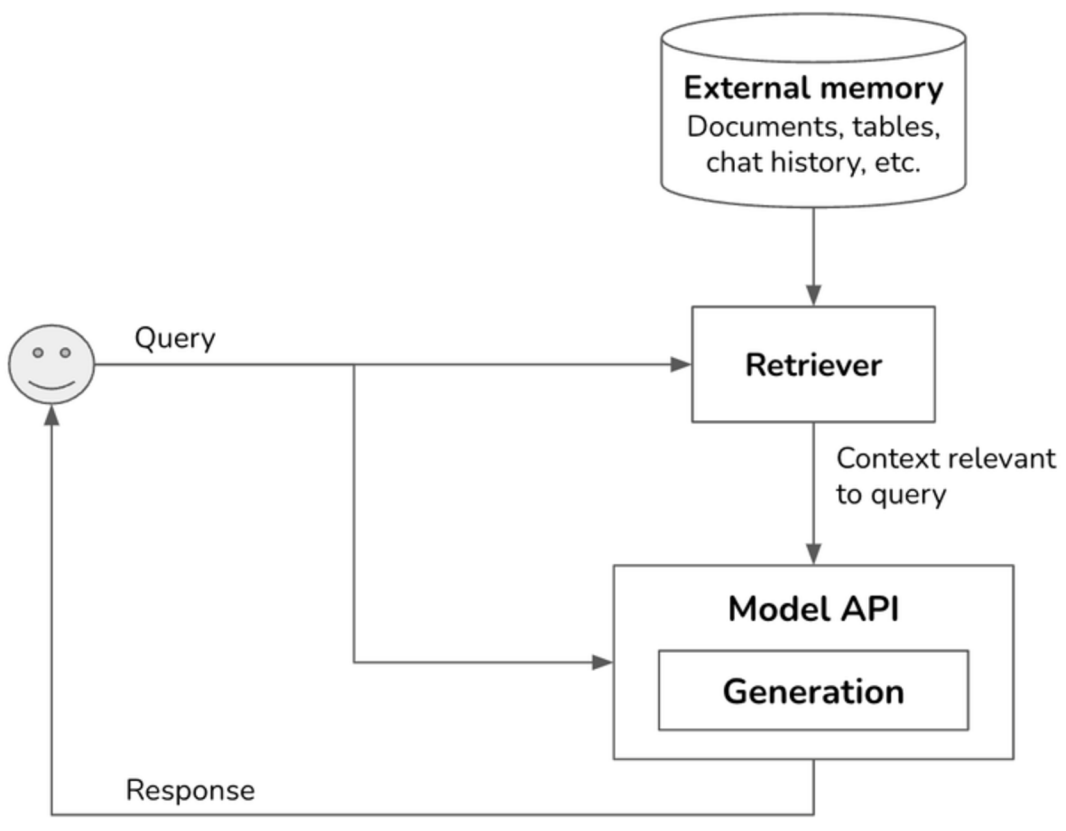
- The "Model API" can refer to both third-party APIs (like OpenAI, Google, Anthropic) and self-hosted APIs.

Credit: Extracted/summarized from Chip Huyen original

1. Enhance Context

- Enhance context by giving it access to external data sources and tools.
- The most common method for enhancing context is RAG (Retrieval-Augmented Generation), which combines a generator (e.g. a language model) with a retriever to pull in relevant external information.

Figure 1: Enhance Context with RAG



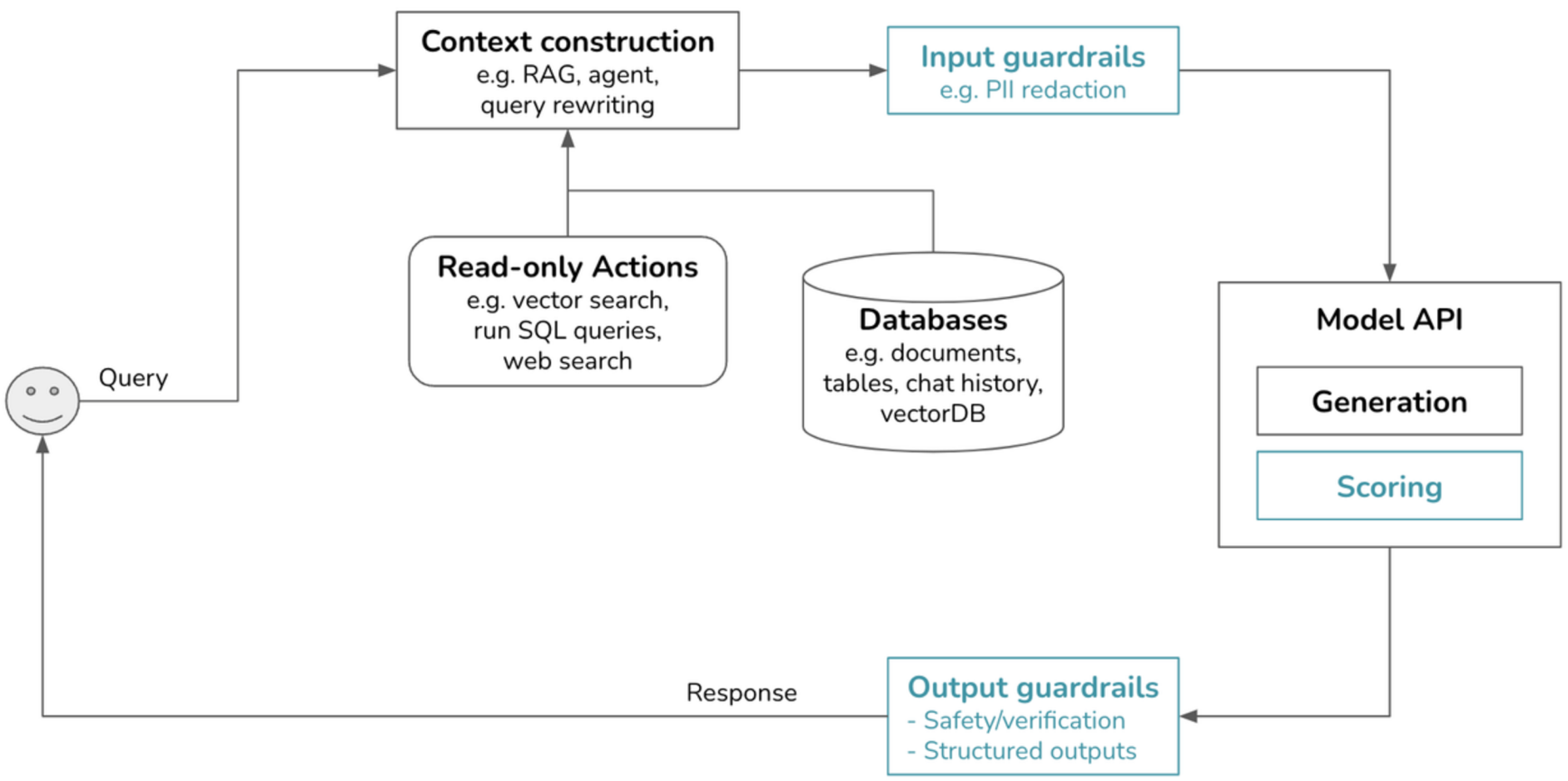
- External data can also be structured, like dataframes or SQL tables. The Internet is a key data source. A web search tool like Google or Bing API allows the model to access up-to-date information queries.
- Often, a user query needs to be rewritten to improve the chances of finding the right information.

Credit: Extracted/summarized from Chip Huyen original

2. Put In Guardrails

- Implement guardrails to protect both your system and users, reducing AI risks. They should be in place where failures might occur. Two types are covered: input and output guardrails.

Figure 2: Put in Guardrails



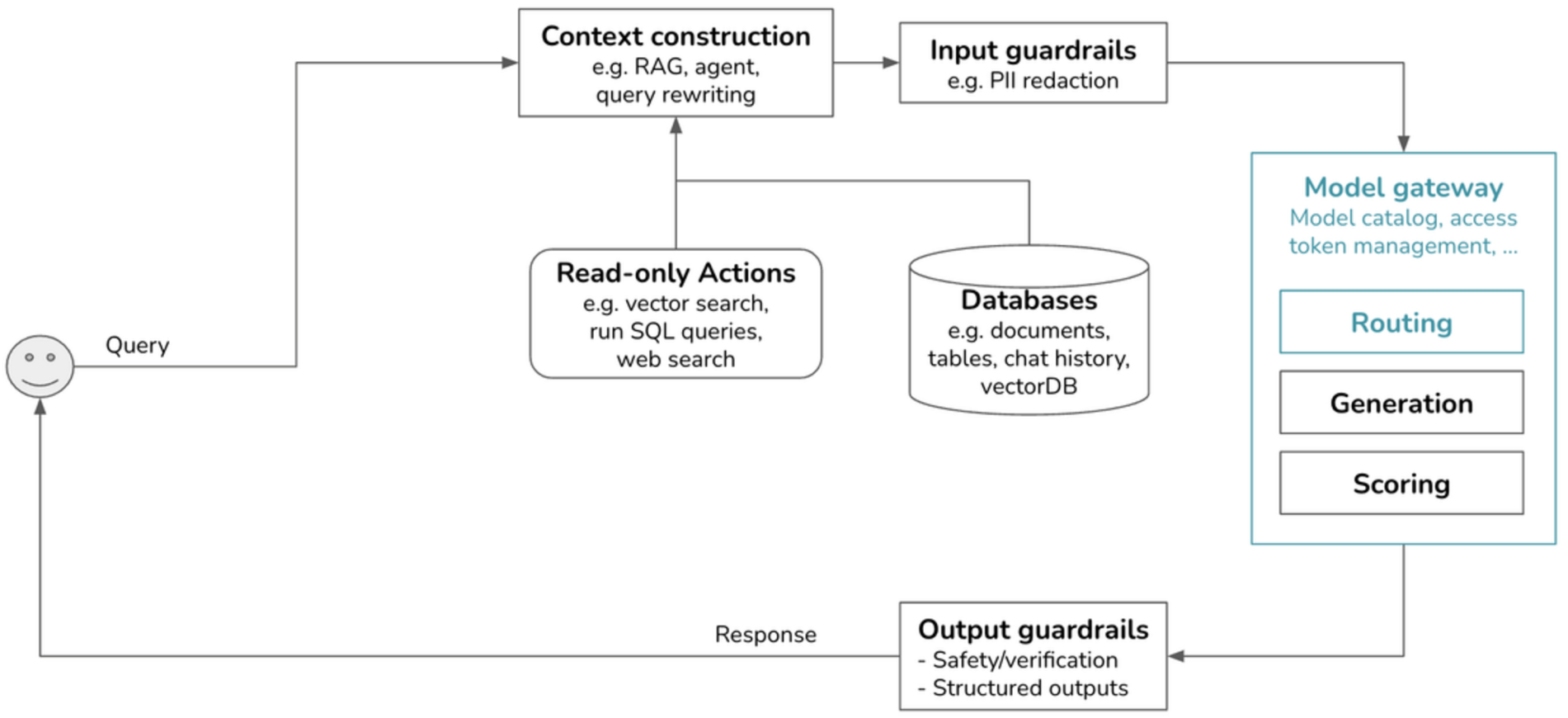
- Input guardrails typically protect against two main risks: leaking private data to external APIs and executing bad prompts that could compromise your system (model jailbreaking).
- Output guardrails enhance application reliability by evaluating the quality of each generation and defining policies to handle different failures.

Credit: Extracted/summarized from Chip Huyen original

3. Add Model Router And Gateway

- Add a model router and gateway to support complex pipelines and improve security.
- As your application grows and involves more models, these tools help you handle multiple models effectively.

Figure 3: Add Model Router and Gateway



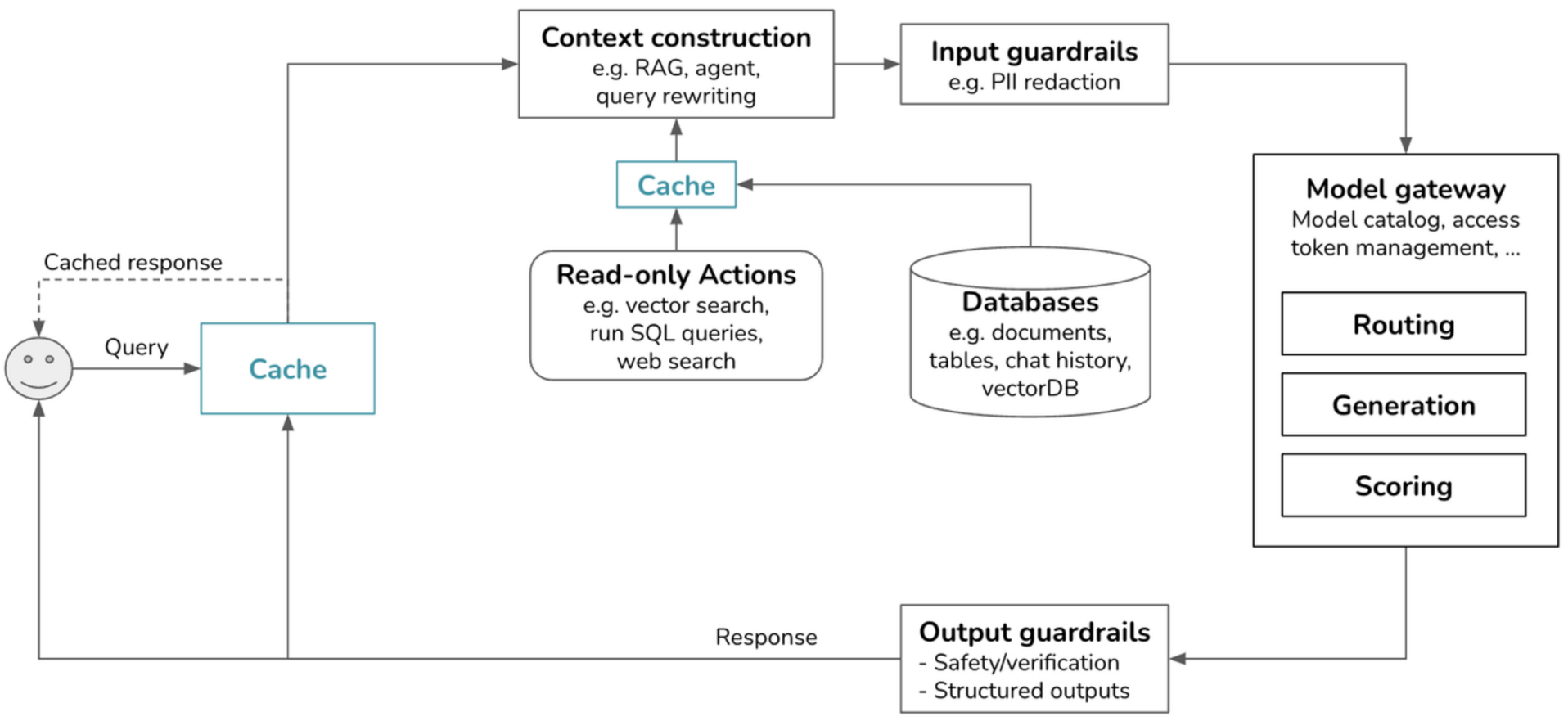
- A model router allows an application to use specialized models for various queries improving performance and cost efficiency.
- A model gateway is an intermediate layer that helps your organization interface with different models, enhancing security and code maintainability.

Credit: Extracted/summarized from Chip Huyen original

4. Reduce Latency With Cache

- Add a model router and gateway to support complex pipelines and improve security.
- As your application grows and involves more models, these tools help you handle multiple models effectively.

Figure 4: Reduce Latency with Cache



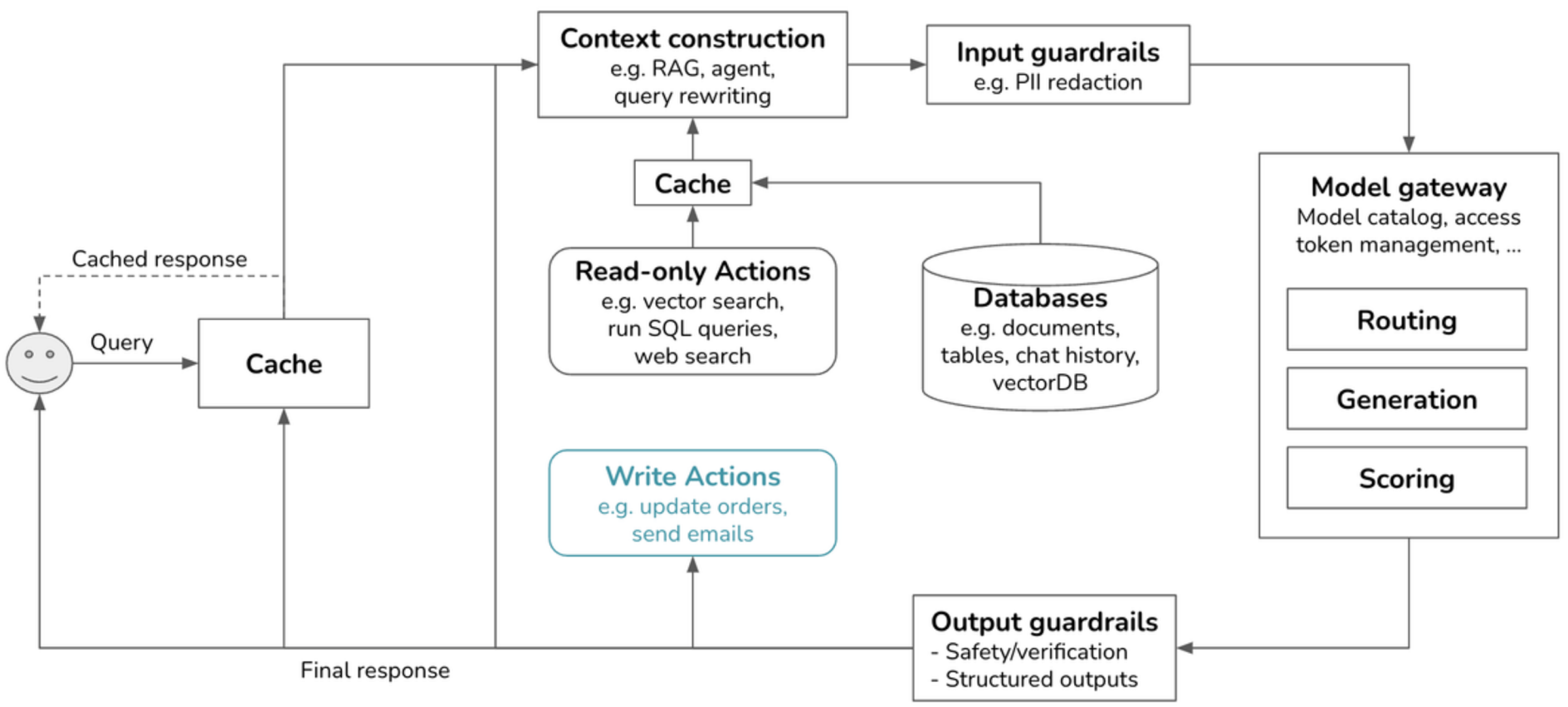
- Many inference APIs offer prompt caching. When choosing an inference library, check which caching mechanisms it supports.

Credit: Extracted/summarized from Chip Huyen original

5. Add Complex Logic And Write Actions

- Add complex logic and write actions to maximise your system’s capabilities.
- So far, we've covered simple flows where outputs are returned to users (unless they fail guardrails). But your application can include loops and conditional branching, and use model outputs for actions like composing emails or placing orders.

Figure 5: Add complex logic and write actions



Credit: Extracted/summarized from Chip Huyen original

Want to level up with Generative AI? Follow  Lewis Walker

Repost this to help your network