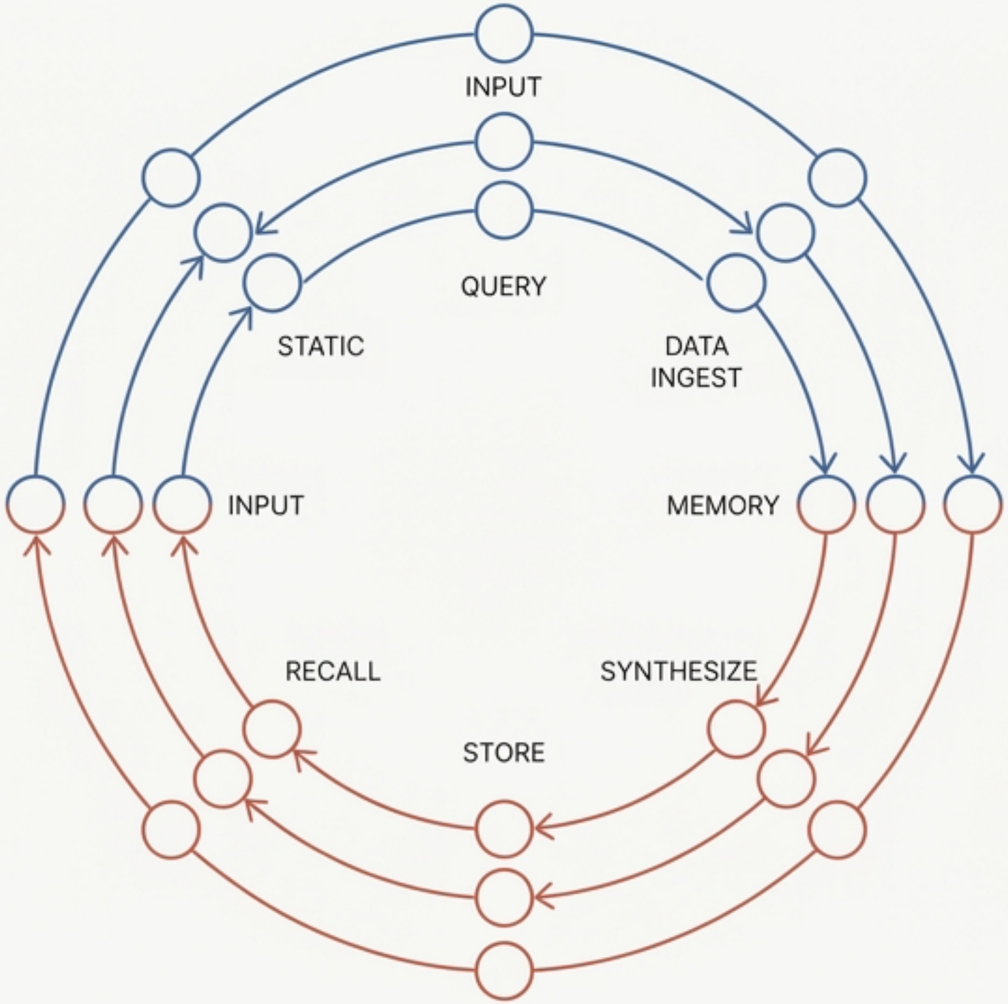
mem-rag: The Architecture of Agentic Memory

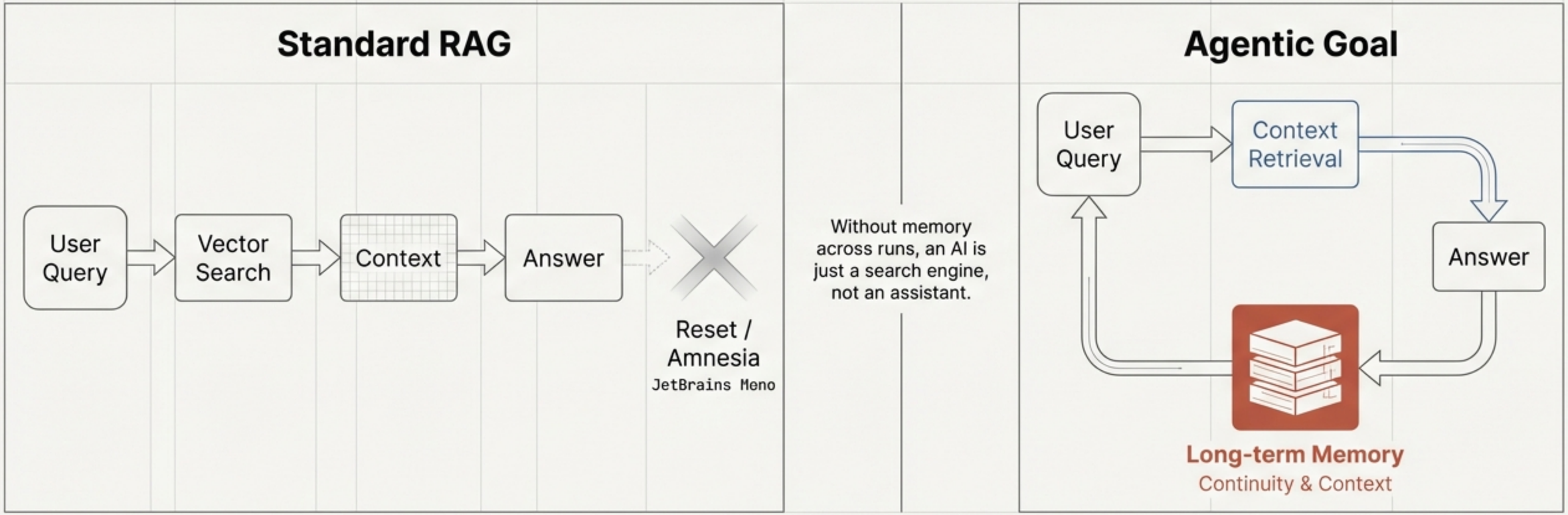
A deep dive into the single-agent personal research assistant CLI.



Powered by Mastra + RAG-Stack IR/CM/PE

Standard RAG suffers from statelessness

Traditional Retrieval-Augmented Generation retrieves context for a query but forgets the interaction immediately after. It is powerful, but transient. True agency requires continuity.



A Personal Research Assistant that persists

Project: venikman/mem-rag

mem-rag is a single-agent CLI tool designed for local research tasks. It combines PDF/Markdown ingestion with agentic long-term memory.

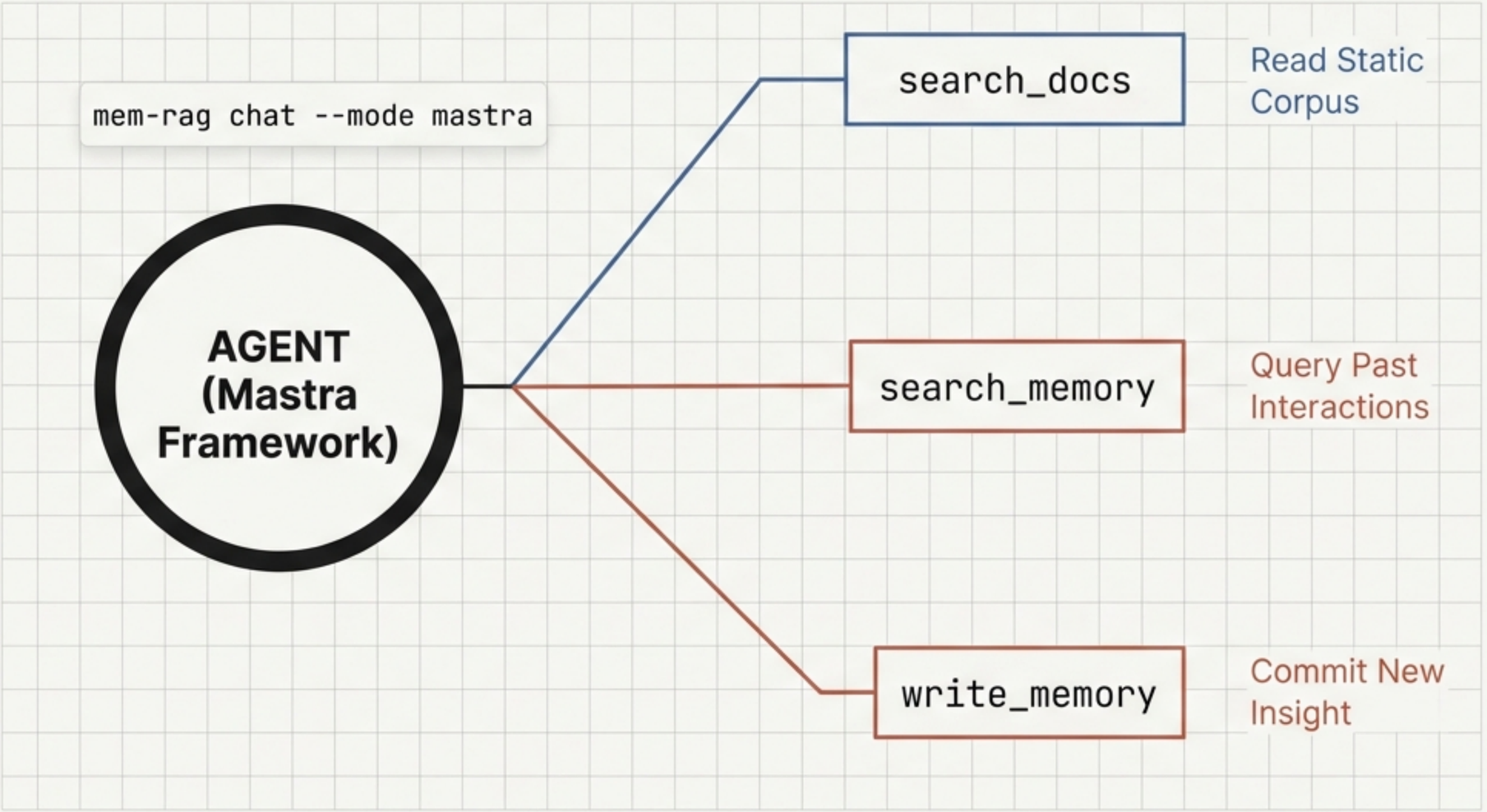
- Ingestion into SQLite + Embeddings
- Episodic & Semantic memory formation
- Optimization via RAG pipeline evaluation

```
>_

$ mem-rag chat
> Welcome to mem-rag assistant.
> Loading agentic memory...
> Ready. Session ID: 8f92-a1b2

What did we discuss about the architecture
yesterday?

> Searching episodic memory...
```

Moving from Passive Retrieval to Active Agency

Standard RAG is read-only. mem-rag utilizes the Mastra framework to enable tool-calling modes.

The agent doesn't just receive data; it decides how to acquire it.



The Agent’s Toolset

Discrete verbs available in Mastra tool-calling mode.

`search_docs`

Querying the static corpus (PDFs/Markdown). The foundation of external knowledge.

`search_memory`

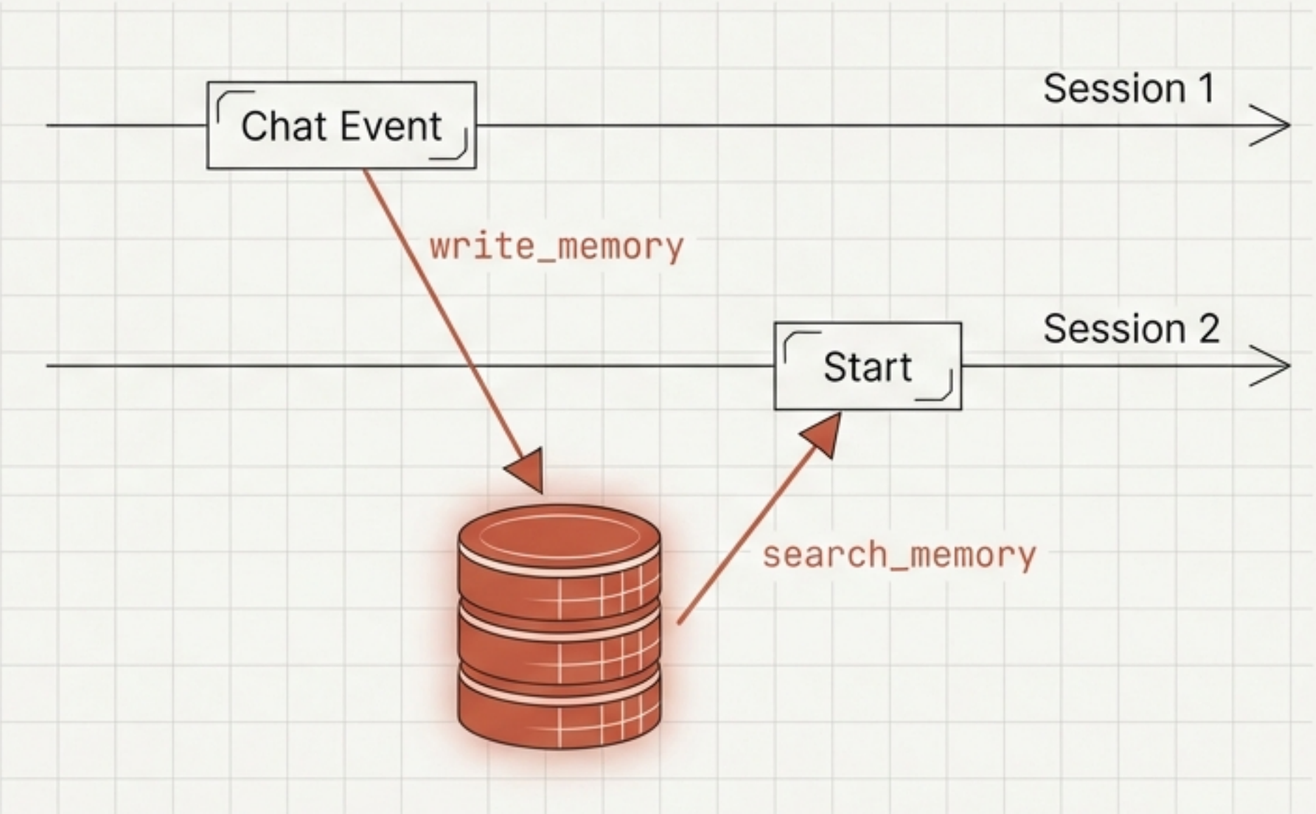
Querying past interactions and synthesized facts. Accessing the agent’s autobiography.

`write_memory`

The ability to commit new insights to storage. Active learning during the session.

The ‘Write’ Capability: Learning Across Runs

Unlike standard chat history which is often limited to a context window, **mem-rag** commits information to persistent storage. This enables the agent to build a personalized knowledge base based on user interaction, effectively “learning” over time.

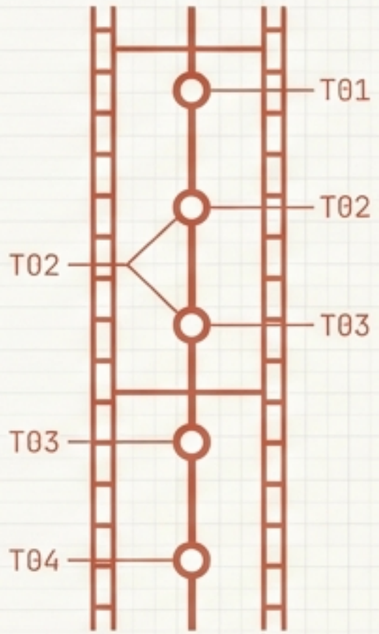


"Agentic long-term memory (episodic + semantic) across runs."

Dual-Track Memory Architecture

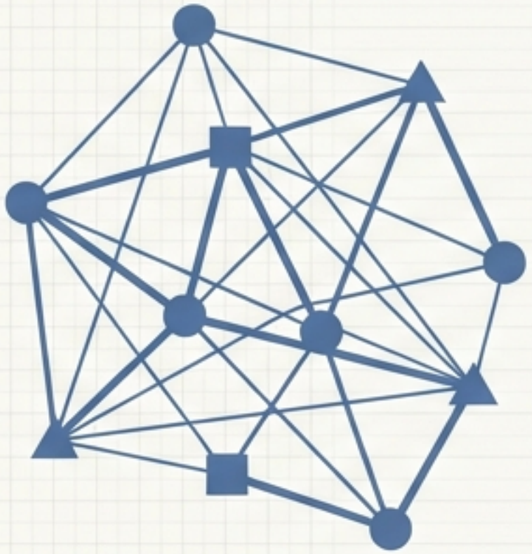
The system structures memory into two distinct categories to maximize retrieval relevance.

Episodic Memory



The Autobiography (Events)

Semantic Memory



The Encyclopedia (Facts)

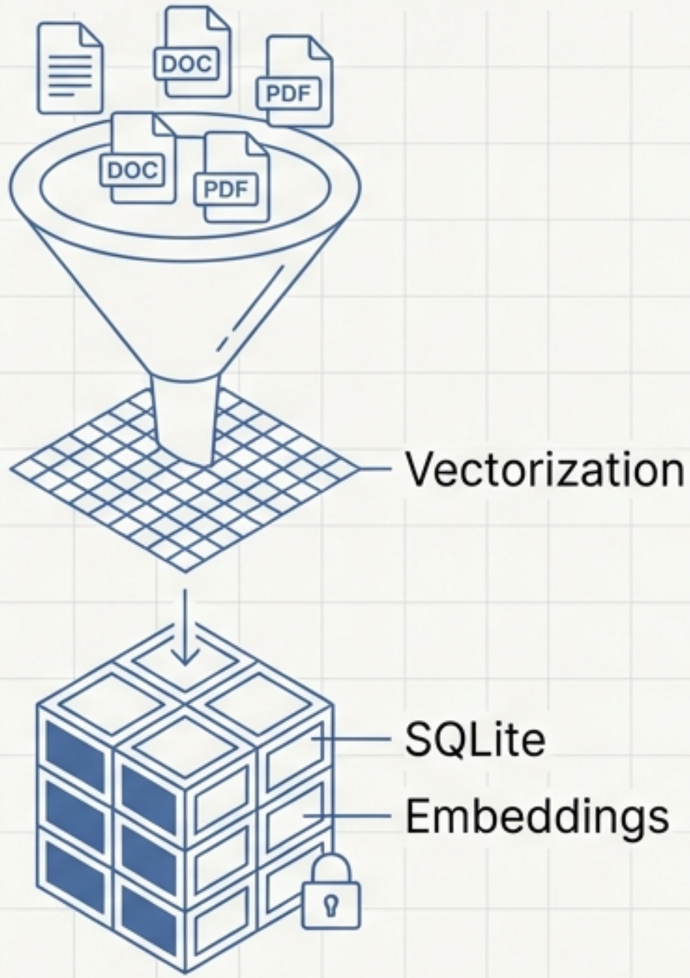
Stored in SQLite + Embeddings

Ingestion: Building the Static Corpus

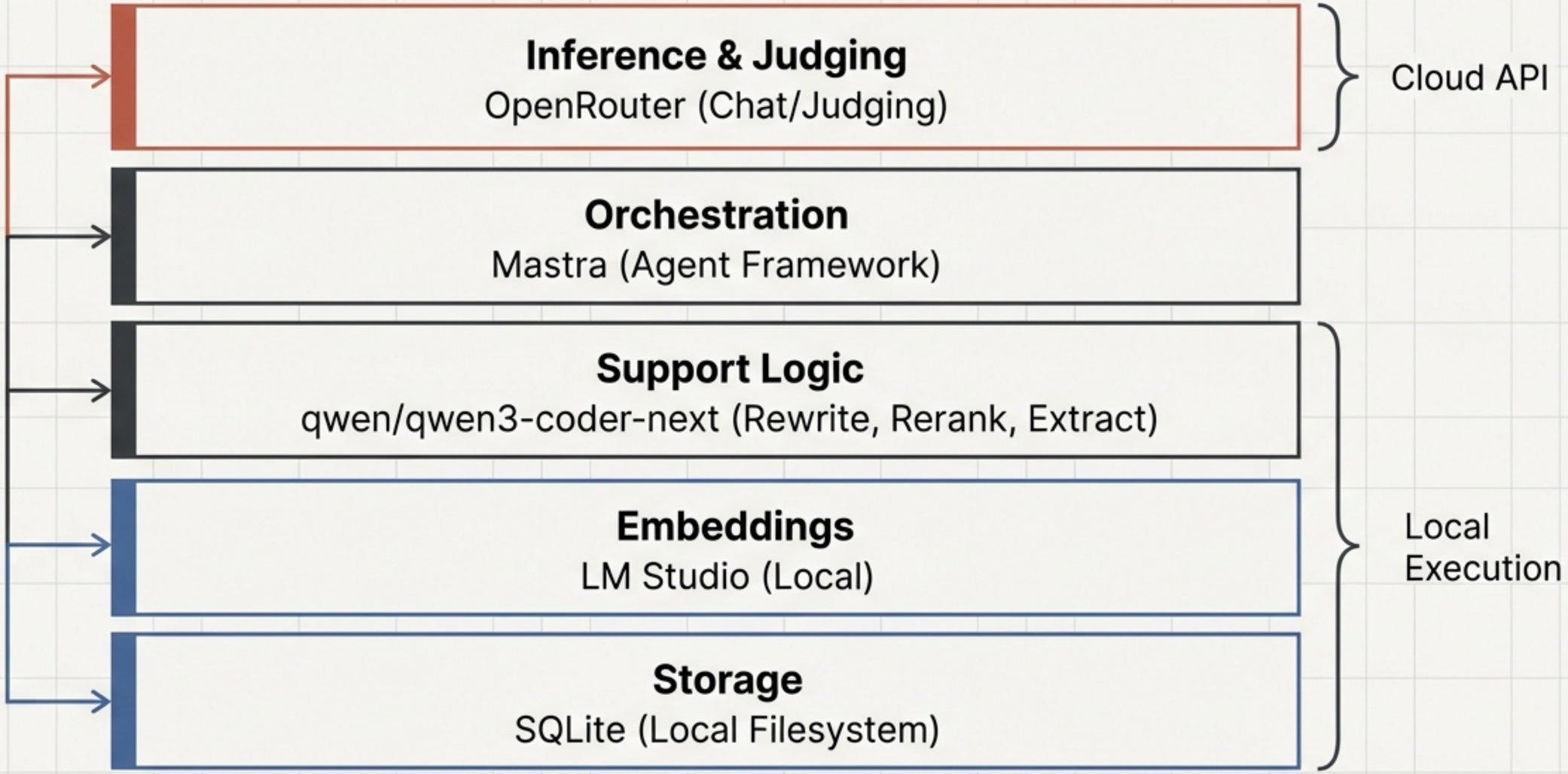
Before memory begins, the foundation is laid through robust ingestion of user data.

Formats	Engine	Storage
**/*.pdf **/*.md	pdftotext (Poppler)	Local SQLite + Embeddings
Inter Regular	Inter Regular	Inter Regular

```
mem-rag ingest ../ --include "**/*.pdf"
   --include "**/*.md"
```

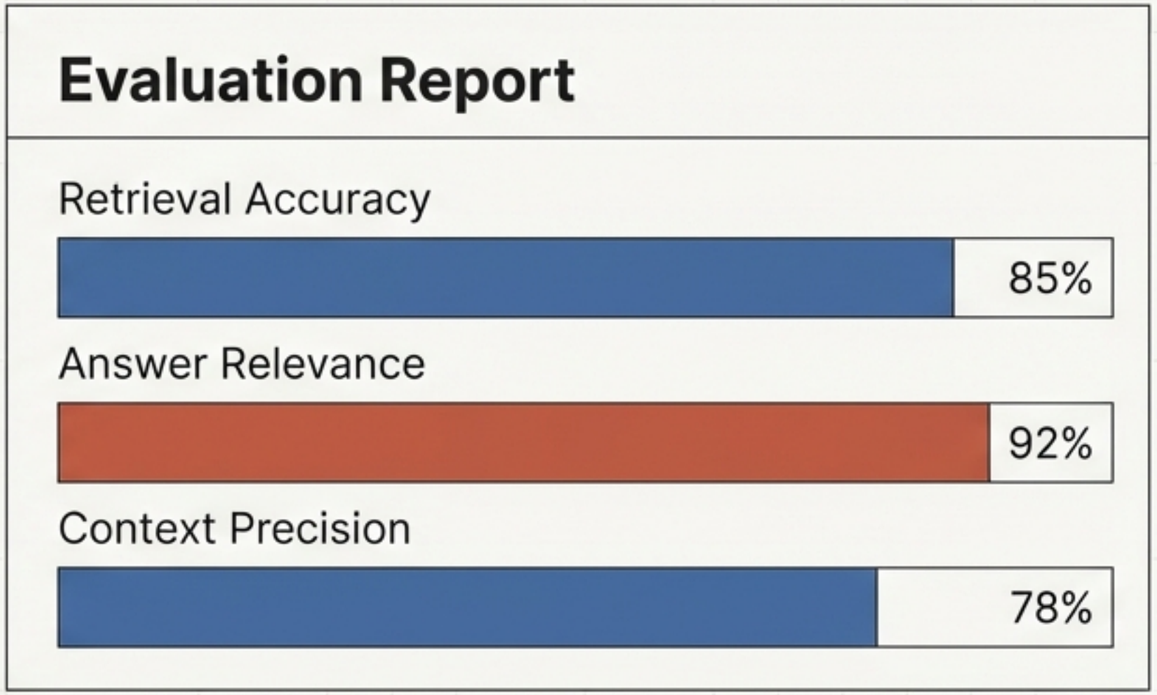


Under the Hood: The Tech Stack



RAG-PE: Pipeline Evaluation

mem-rag includes a dedicated evaluation mode to grade answers against a set of questions, moving beyond questions, moving beyond "vibes-based" testing.



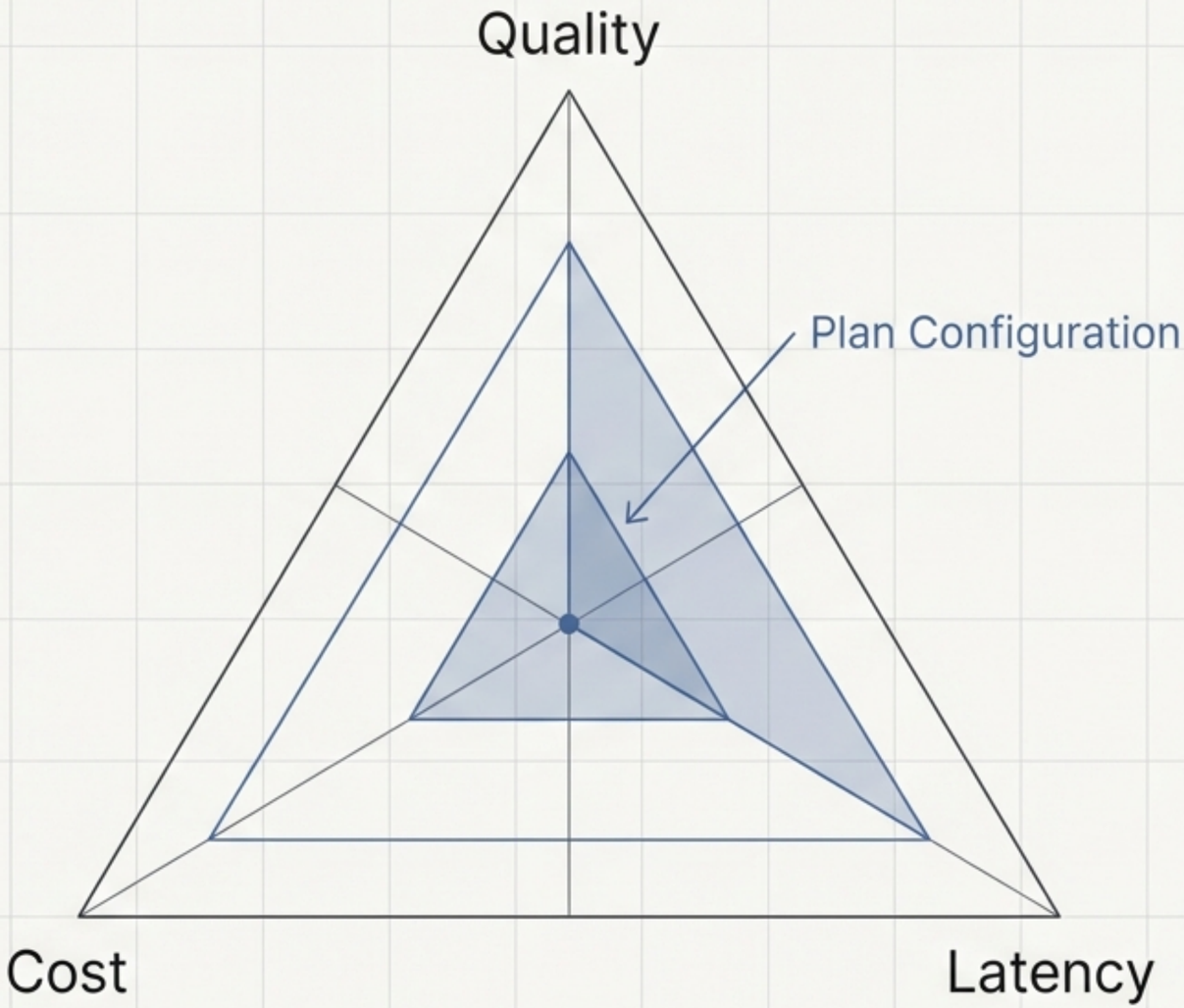

```
mem-rag eval --questions eval/questions.jsonl
```

Plan Exploration and Optimization

The system allows for configuration tuning to balance trade-offs between quality, cost, and latency.

```
mem-rag optimize --questions
eval/questions.jsonl
```

Transforms the project from a demo into a tunable product.



Why mem-rag is Unique

01

True Agency

The agent chooses when to read or write memory via Mastra tool-calling.

02

Persistence

Memory survives the session through local SQLite implementation.

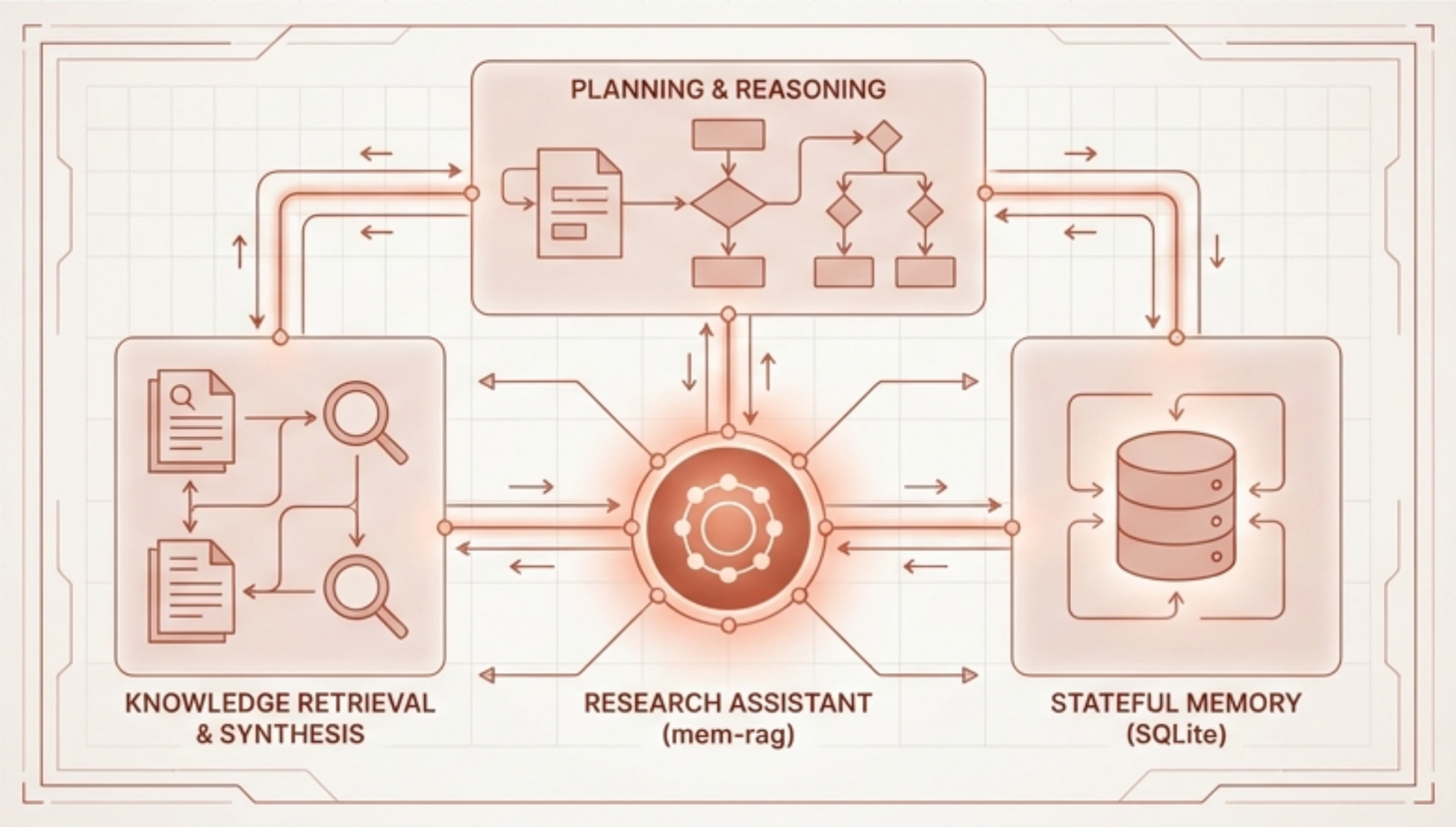
03

Self-Correction

Built-in optimization loop (RAG-PE) ensures continuous improvement.

From ‘Chat with PDF’ to ‘Collaborate with Researcher’

mem-rag provides the blueprint for a local, private, and stateful research assistant that grows with your work.



venikman/mem-rag