



Weaviate

Building Agentic RAG Systems

Erika Cardenas

Technology Partner Manager



Outline



Vanilla RAG vs Agentic RAG

Vanilla RAG

Agent Components

Agentic RAG



Agent Ecosystem

LLM + Function Calling

Agent Frameworks

Observability



Generative Feedback
Loops

Weaviate Agents

GFL Applications



Vanilla RAG vs Agentic RAG



Vanilla RAG

Agent Components

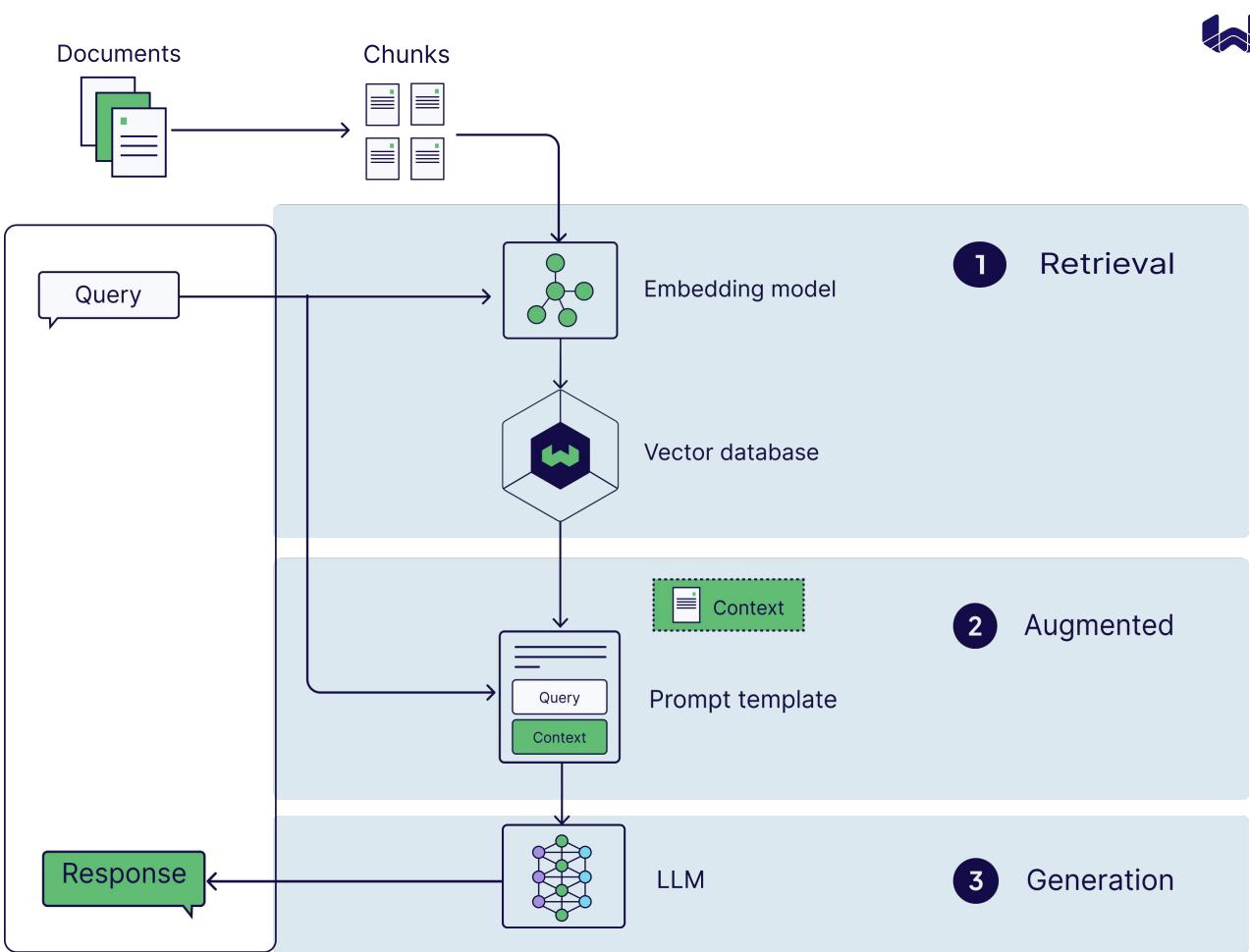
Agentic RAG



RAG

Vanilla RAG Workflow

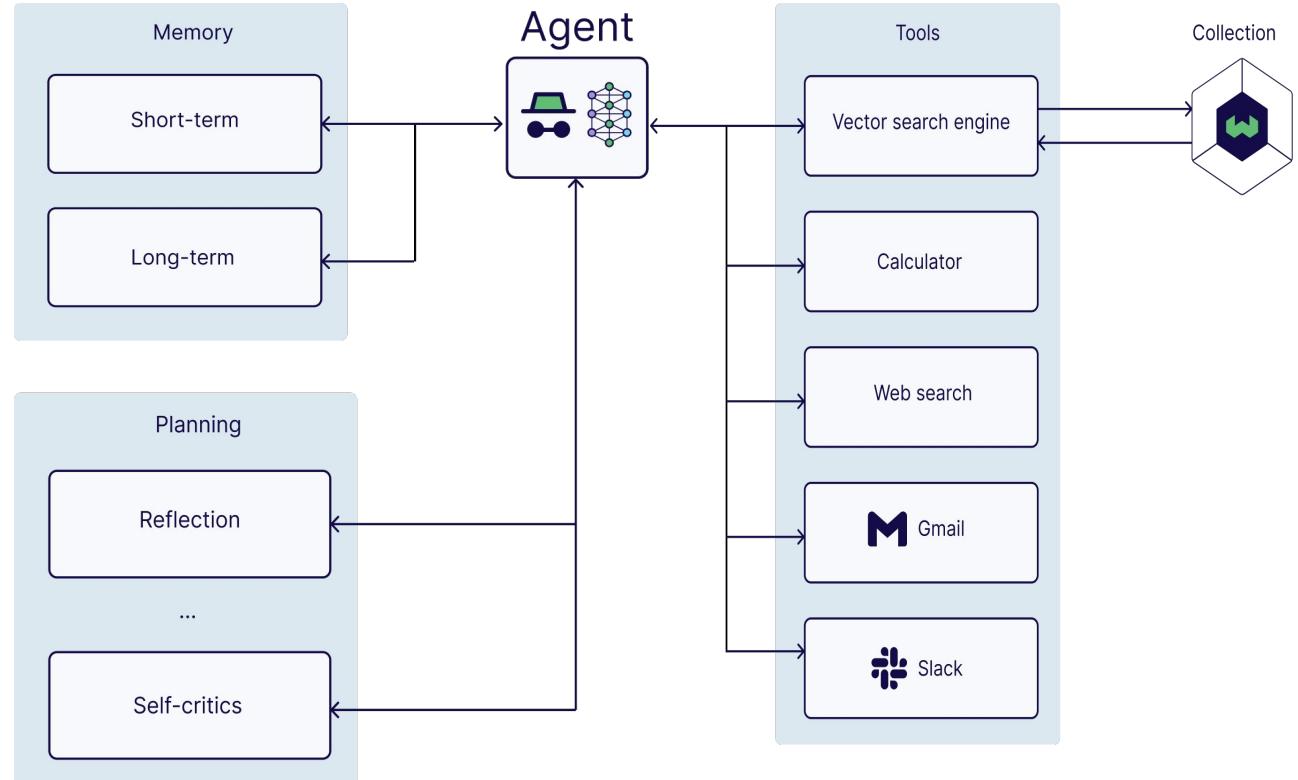
1. Retrieve
2. Augment
3. Generate





Agent Components

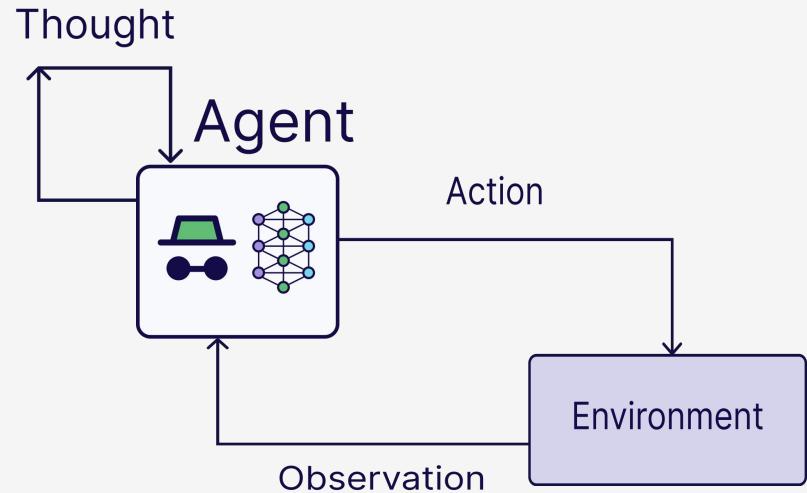
1. Language Model
2. Memory (short and long term)
3. Planning (CoT, ReAct, etc.)
4. Tools





Agentic RAG with ReAct

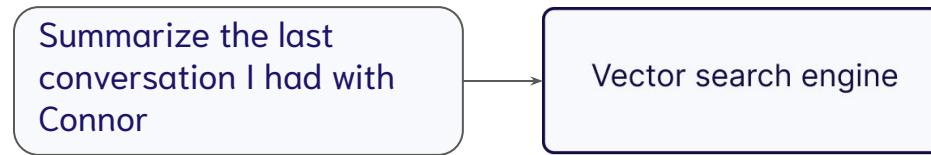
1. Thought: Reason about the action to take based on the user query and past observations
2. Action: Decide an action and execute the tool (database queries)
3. Observation: Observe the feedback from the action
4. Loop until the agent responds to the user





Query Example where Vanilla RAG Fails

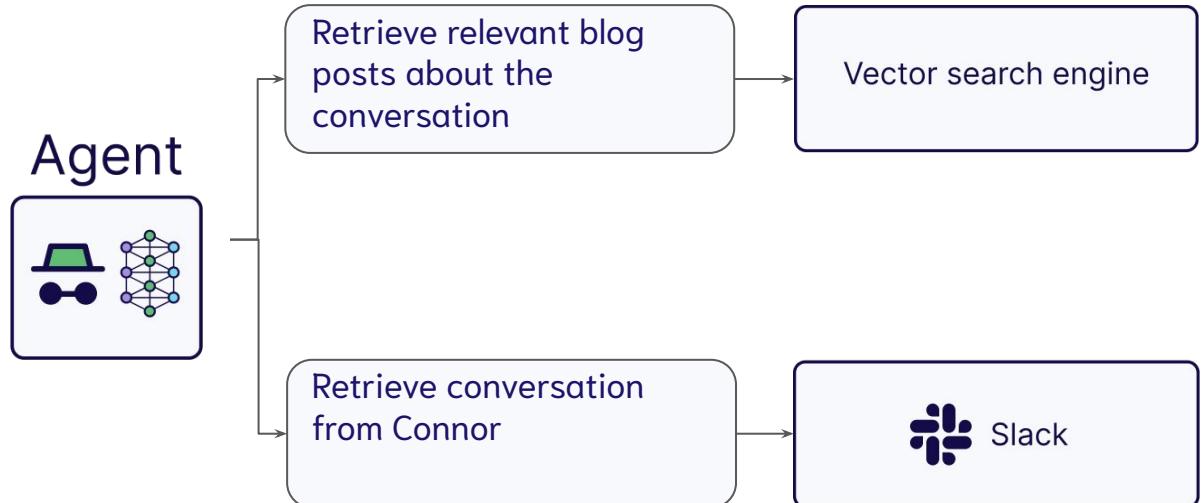
"Summarize the last conversation I had with Connor"





Query Example where Agentic RAG Succeeds

"Summarize the last conversation I had with Connor"





Benefits

1. Format the search query from the prompt
2. Call tools in parallel
3. Navigate your database
4. Iteratively search

Limitations

1. Latency
2. Inference Cost

Vanilla RAG vs Agentic RAG



45 Weaviate FAQs

What is the role of the Binary Independence Model in the BM25 algorithm used by Weaviate's hybrid search?

Why might vector libraries not be suitable for applications that require real-time updates and scalable semantic search?

What guide does the document recommend for learning about LangChain projects?

...

LLM as Judge



Agentic RAG



Vanilla RAG



Exploring Multi-Agent Systems

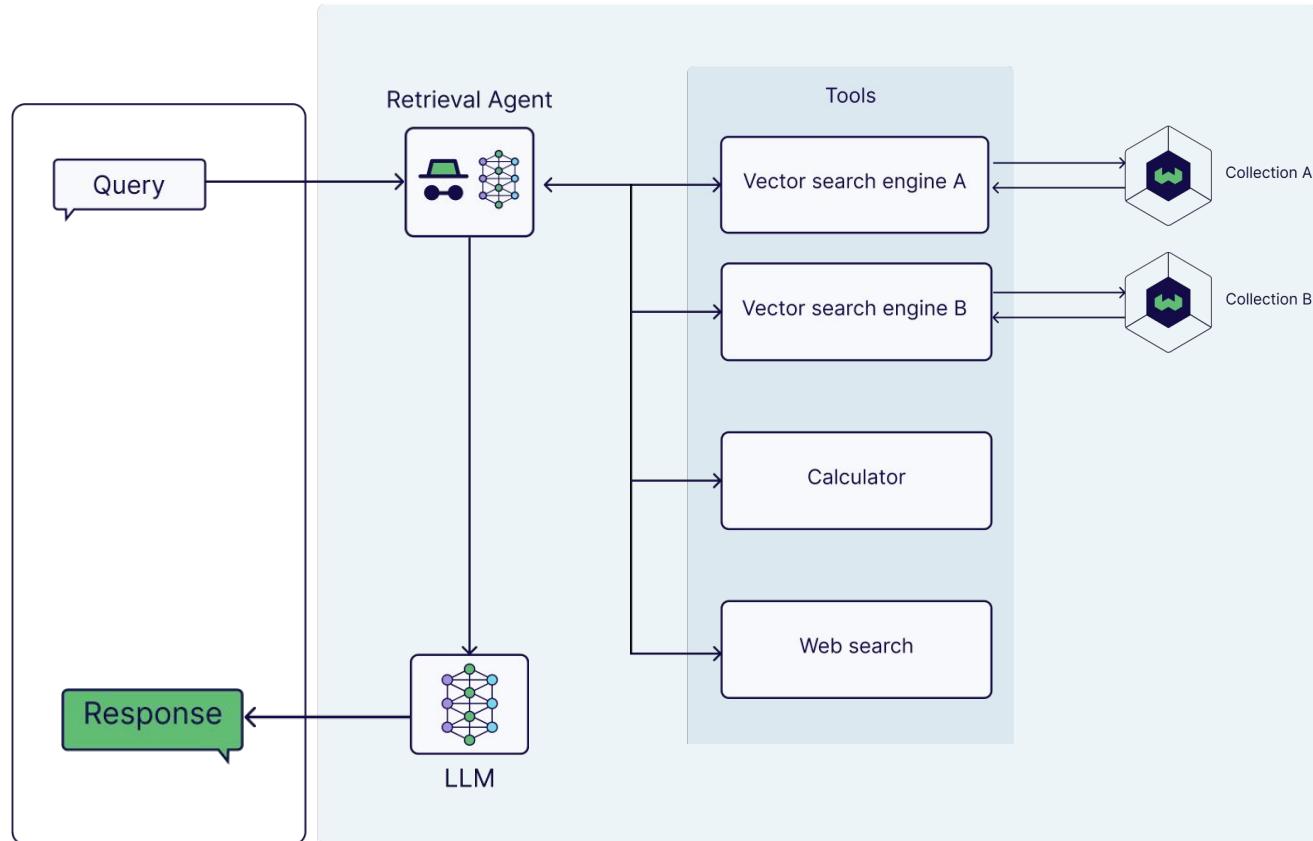
- Roles and Routing
- Sharing and Isolating Tool Use



RAG

Workflow

1. Retrieval agent
that accesses
external tools
2. Language model
(summarize, etc.)

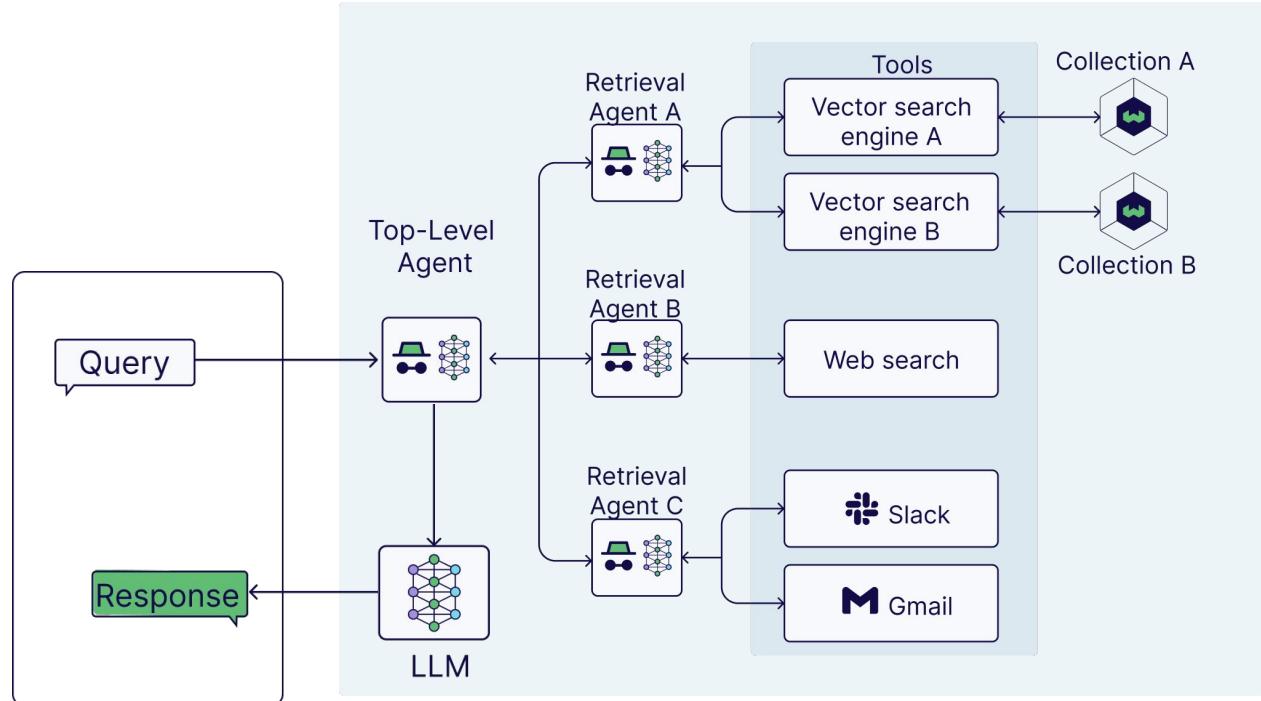




RAG

Multi Agent Retrieval Workflow

1. Top-level agent
2. Call the required tool
3. Top-level agent sends the outputs to the other language model





Agent Ecosystem



LLM + Function Calling

Agent Frameworks

Observability



Gemini + Function Calling

Gemini

```
calculator = genai.protos.Tool(
    function_declarations=[
        genai.protos.FunctionDeclaration(
            name='multiply',
            description="Returns the product of two numbers.",
            parameters=genai.protos.Schema(
                type=genai.protos.Type.OBJECT,
                properties={
                    'a':genai.protos.Schema(type=genai.protos.Type.NUMBER),
                    'b':genai.protos.Schema(type=genai.protos.Type.NUMBER)
                },
                required=['a','b']
            )
        )
    ]
)
```



Agent Frameworks



- Variety of tools and reasoning loops



- Supports ReAct agents and Avatar optimization



- Framework for updating agent memory



- LCEL and LangGraph frameworks



- Multi-agent orchestration

Observability



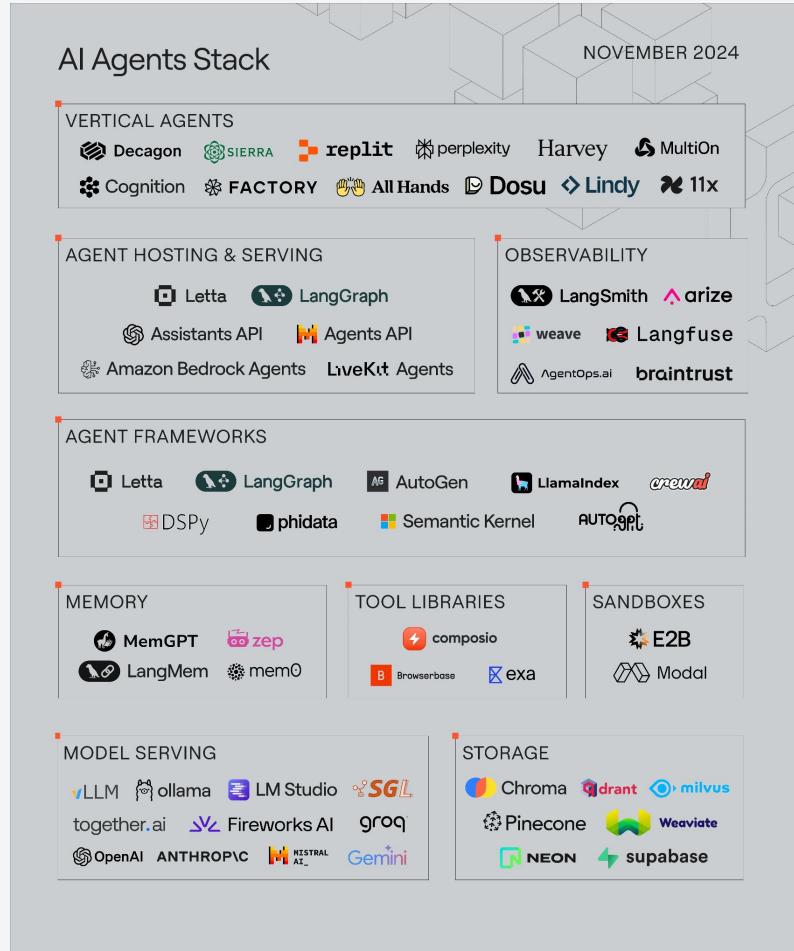
The screenshot shows the Arize AI Observability interface. On the left, there's a sidebar with icons for projects, traces, and other metrics. The main area is titled "Trace Details" and shows a summary of the trace status (OK) and latency (79.22s). The "Traces" tab is selected, displaying a tree view of the trace hierarchy. One node, "agents_call 79.16s", is expanded, revealing several sub-nodes under it, such as "ChatCompletion" and "run_sql_query".

The right side of the interface provides detailed information about a specific trace entry:

- Info:** ChatCompletion
- Feedback:** 0
- Attributes:** Events 0
- Events:** gpt-4o-2024-05-13
- Input Messages:** Tools Input Invocation Params
- system:** You are a helpful assistant that chooses a tool to call based on the user's request. All of your responses should be a tool call or text. Once you receive the results from all of your skills, generate a response to the user that incorporates all of the results. First, identify and make all necessary tool calls based on the user prompt. Ensure that you gather and aggregate the results from these tool calls. Once all tool calls are completed and the final result is ready, return it in a single message. When the task is fully completed, ensure the final message contains the full result, followed by 'TERMINATE' at the very end.
- user: User:** What trends do you see in my traces table?
- Output Messages:** Output
- assistant:** T, E, C

On the far right, there are buttons for "Status" (OK), "Start Time" (10/1/2024 08:26:24 AM), "End Time" (10/1/2024 08:26:25 AM), "Latency" (0.69s), and "Total Tokens" (388). A "Get help" button and a message icon are also present.

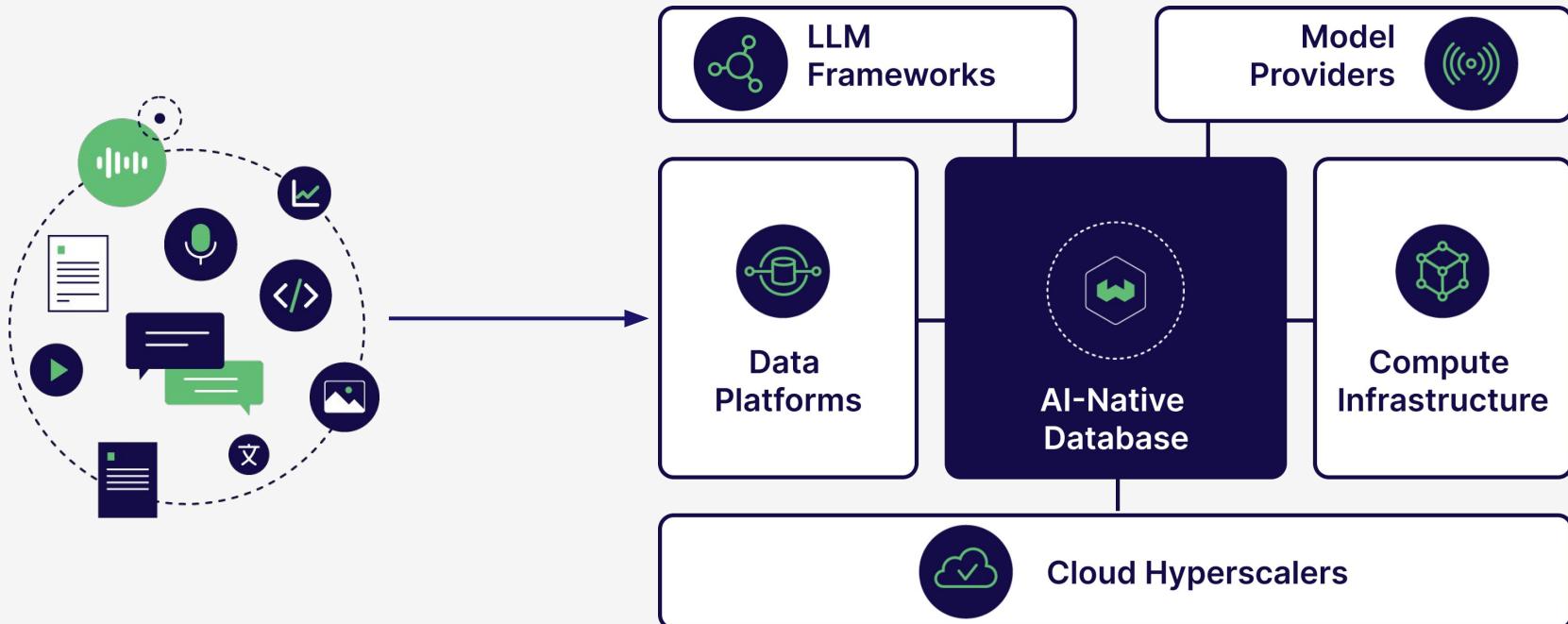
Source:<https://arize.com/blog/what-is-autogen/>



Source: <https://www.letta.com/blog/ai-agents-stack>



A new AI-native stack is emerging



Weaviate Recipes



weaviate / recipes

Type to search | + | 

[Code](#) [Issues](#) [Pull requests](#) [Discussions](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 **recipes** Public

[Edit Pins](#) [Unwatch](#) 32 [Fork](#) 106 [Starred](#) 570

[main](#) [22 Branches](#) [0 Tags](#) [Go to file](#) [Add file](#) [Code](#)

dudanogueira Merge pull request #173 from weaviate/llama-index-update ccb1680 - yesterday 625 Commits

.github Update PR template 3 months ago

integrations Merge pull request #173 from weaviate/llama-index-update yesterday

weaviate-features Remove OctoAI from Recipes 2 weeks ago

.gitignore Move MT and Ollama 6 months ago

README.md Update README 3 weeks ago

About

This repository shares end-to-end notebooks on how to use various Weaviate features and integrations!

[python](#) [vector-search](#) [vector-database](#)
[dspa](#) [generative-ai](#)
[retrieval-augmented-generation](#)
[llm-frameworks](#)

[Readme](#)
[Activity](#)
[Custom properties](#)

570 stars
32 watching
106 forks
Report repository

Contributors 36

+ 22 contributors

Languages

Jupyter Notebook 88.3% MDX 11.4%
Other 0.3%

Suggested workflows
Based on your tech stack

Gulp Configure

Welcome to Weaviate Recipes ❤️





Generative Feedback Loops

GFL API

GFL Applications



Weaviate

Generative Feedback Loops

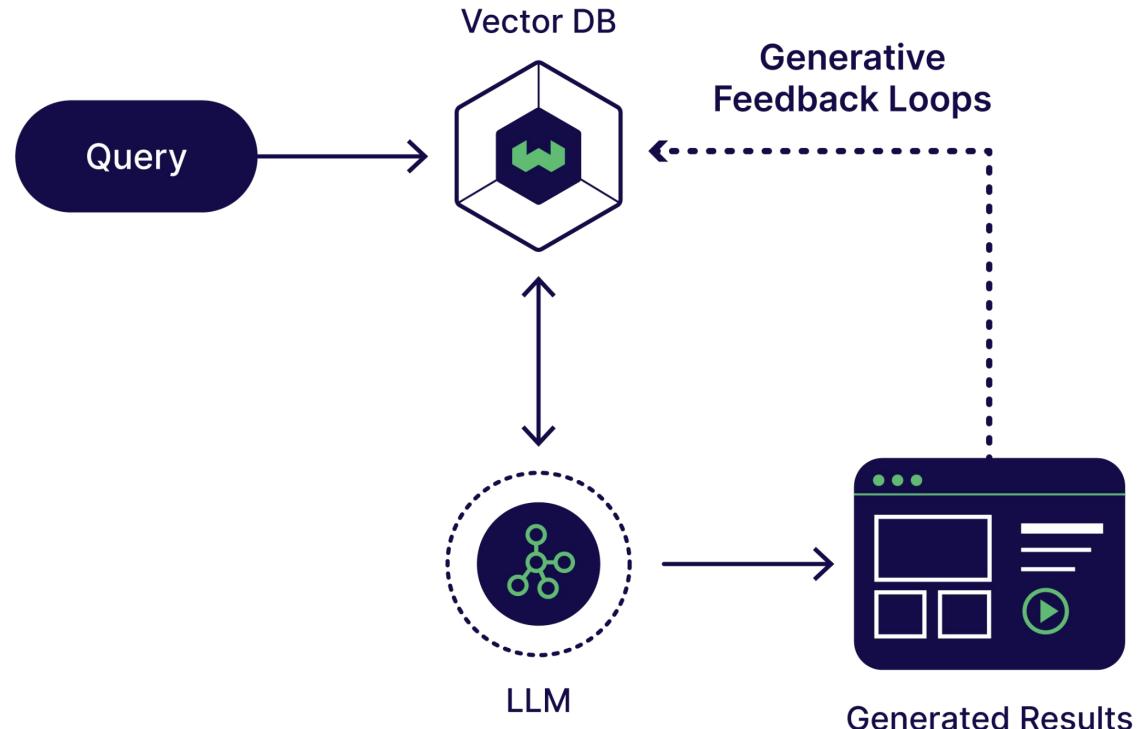


```
instruction = """  
Create an overview of the content. The overview should capture the  
core topics and the main points of the post but does not need to  
include all the details.  
"""  
  
overview_gfl = blogs.gfl.create(  
    property_name="overview",  
    data_type=wvcc.DataType.TEXT,  
    view_properties=["content"],  
    instruction=instruction,  
)
```



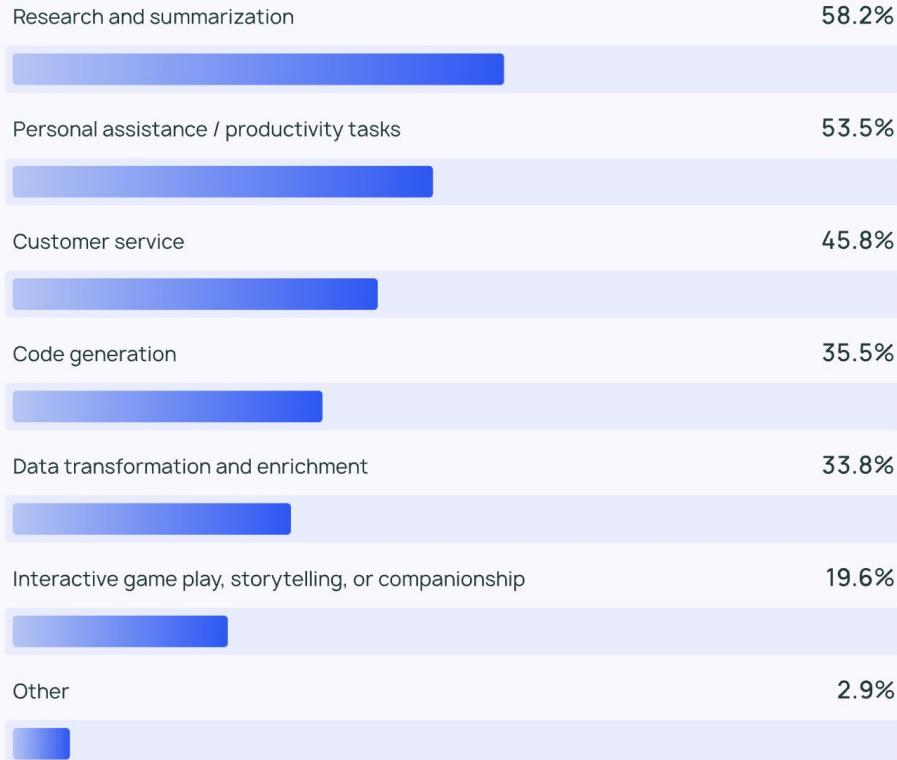
GFL Applications

- Data cleaning
- Chunking text documents
- Generate synthetic data





In your opinion, which tasks are agents best suited to perform today?





Summary



Vanilla RAG vs Agentic RAG

Vanilla RAG

Agent Components

Agentic RAG



Agent Ecosystem

LLM + Function Calling

Agent Frameworks

Observability



Generative Feedback
Loops

Weaviate Agents

GFL Applications

Resources

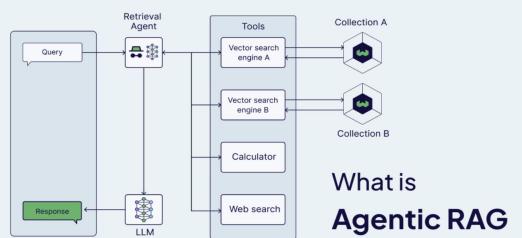


What is Agentic RAG

November 5, 2024 · 11 min read

Erika Cardenas
Technology Partner Manager

Leonie Monigatti
Machine Learning Engineer



What is
Agentic RAG

Agentic RAG

Erika Cardenas
Weaviate



#109



recipes (Public)

Edit Pins Unwatch 32 Fork 106 Starred 570

main 22 Branches 0 Tags

Go to file Add file Code

dudanogueira Merge pull request #173 from weaviate/llama-index-update ccb1680 · yesterday 625 Commits

.github Update PR template 3 months ago

integrations Merge pull request #173 from weaviate/llama-index-update yesterday

weaviate-features Remove OctoAI from Recipes 2 weeks ago

.gitignore Move MT and Ollama 6 months ago

README.md Update README 3 weeks ago

README

About

This repository shares end-to-end notebooks on how to use various Weaviate features and integrations!

python vector-search vector-database
dspy generative-ai
retrieval-augmented-generation
llm-frameworks

Readme
Activity
Custom properties

570 stars



Thank you!



weaviate.io



[weaviate/weaviate-recipes](https://github.com/weaviate/weaviate-recipes)



[@ecardenas300](https://twitter.com/ecardenas300)



[/ecardenas300/](https://www.linkedin.com/in/ecardenas300/)