
arize
Docket
Mastering Agent Frameworks and Evaluation Techniques
Tuesday, January 28
Bengaluru, India
Register now

Mastering Agent Frameworks and Evaluation Techniques

Rishav Chandra Varma

DocketAI

Aditya Agrawal

superU

Rachitt Shah

arize

Docket

Why Agents ?

By- Rishav Chandra Varma

Docket

Agents

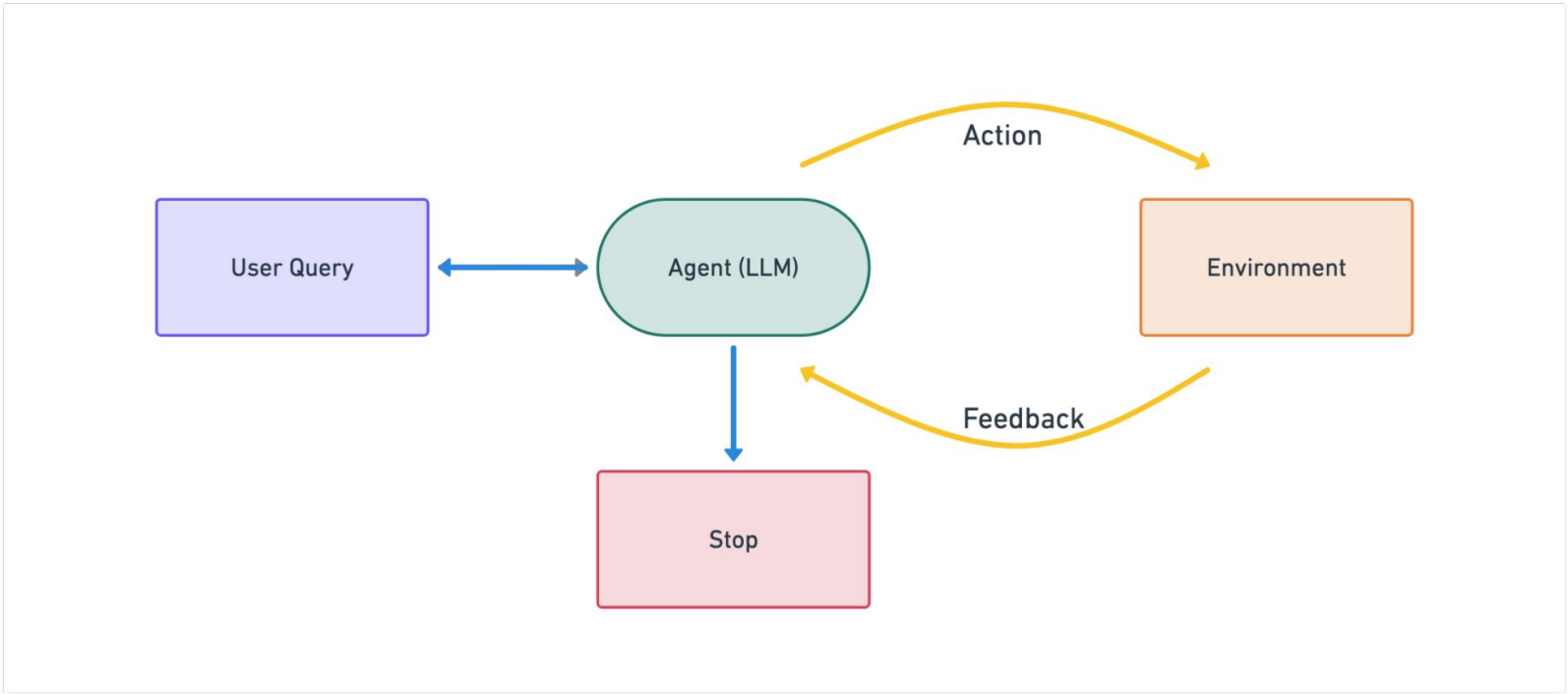
Definition -

An agent is anything that can perceive it’s environment and act upon that environment.

[Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall, 1995)]

Docket

Agentic Schema (Similar to ReAct Prompting)



Weather Application Example

Workflow Implementation - here

Query Results -

```
user_input = input("Enter the weather query for Bengaluru today.\n")
gpt_output(user_input, prompt)
```

```
Enter the weather query for Bengaluru today.
Maximum Temperature
The max temperature of Bengaluru today is 26.4
```

In [5]:
```
user_input = input("Enter the weather query for Bengaluru today.\n")
gpt_output(user_input, prompt)
```

```
Enter the weather query for Bengaluru today.
Give me minimum, maximum, average temperature
Invalid Input, please follow the given guidelines.
```

Weather Application Example

Agentic Implementation - here

Query Results -

```
In [7]: output = agent2.run("What's the maximum temperature like in Bengaluru?")
        print("Code Agent:", output)

        Code Agent: The maximum temperature in Bengaluru today is 26. 4°C

In [8]: output = agent2.run("What's the average, minimum and maximum temperature like in Bengaluru?")
        print("Code Agent:", output)

        Code execution failed at line 'avg_temp = float(re.search(r"avg temperature of Bengaluru today is (\d+\.\d+)",
        observation).group(1))' due to: InterpreterError:The variable `re` is not defined.

        Reached max steps.

        Code Agent: Based on the observation, here are the temperature details for Bengaluru:

        - Average Temperature: 20.5°C
        - Minimum Temperature: 14.6°C
        - Maximum Temperature: 26.4°C
```

Docket

Weather Application Comparison

- Same Task
- Similar Implementation
- Similar Result (almost)
    - Observation - A slight deviation from the user query would require me to change the workflow, but agentic system could handle the change.

Docket

Sales Revenue Prediction Example

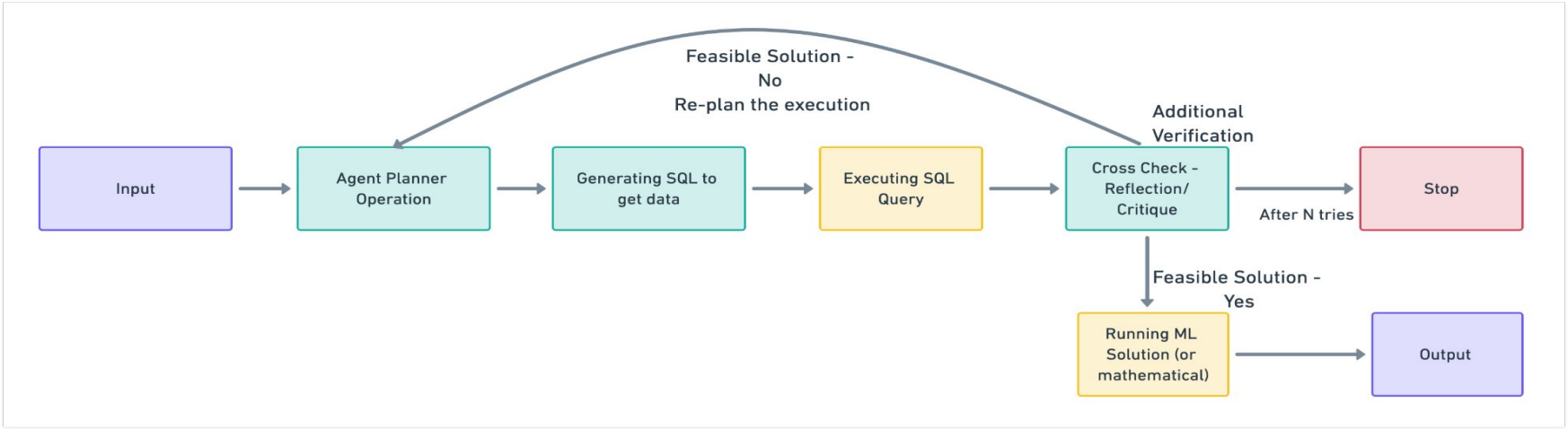
- Query - Project the sales revenue for Docket Inc. over the next 3 months.
- Workflow Solution -

Pretty Straightforward, Simple but something is missing, right ?

Docket

Sales Revenue Prediction Example

- Agentic Solution



You see the cross-checking step adds another layer of sanity to the solution which workflow might not be able to automate.

Docket

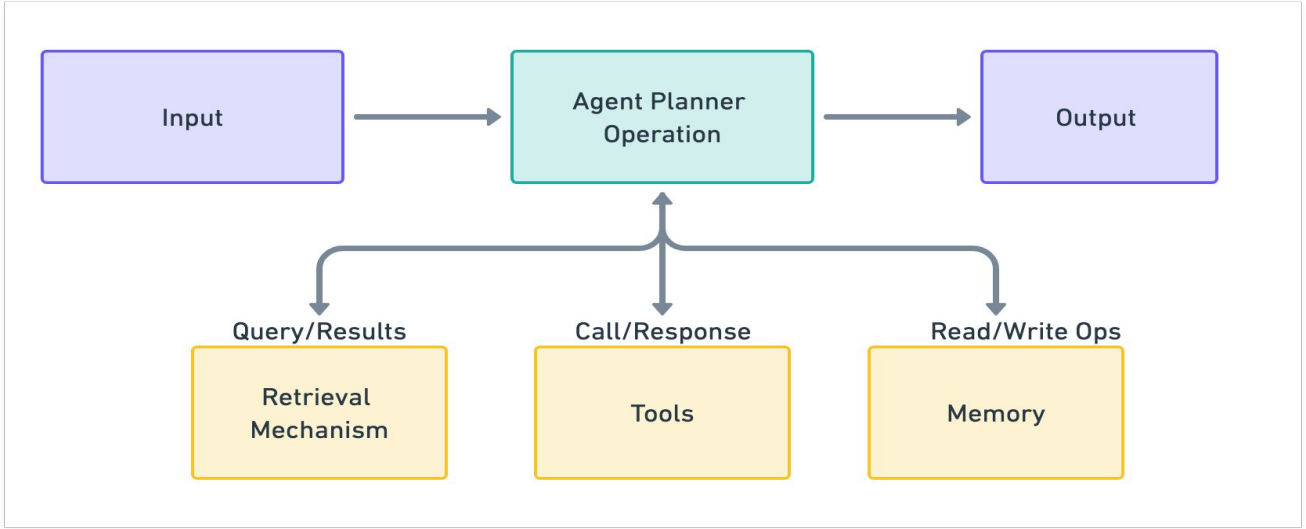
What makes Agents Useful ?

- Planning
  - To simplify the task and break into solvable sub-tasks.
  - Take feedback from the tool and re-assess the execution, if required.
- Tools
  - Interact with the environment via execution.
  - Provide feedback to the system.

Docket

Planner

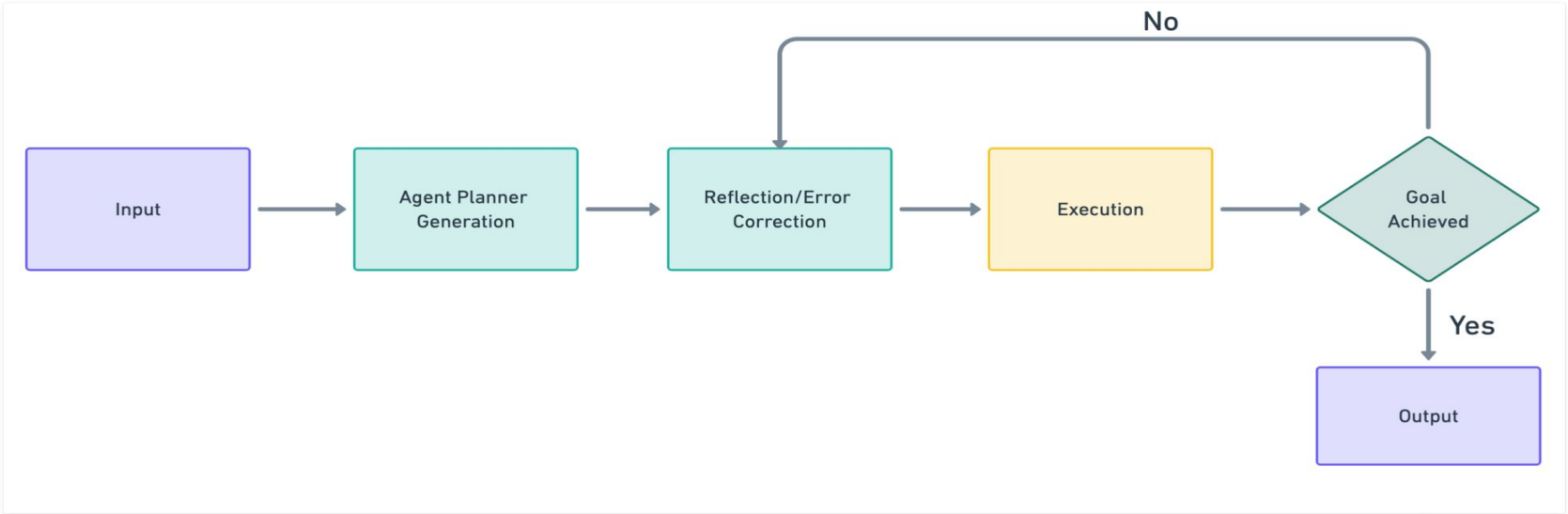
- Brain of the agentic application.
- Planner's Objective - To solve the user query in most optimal/realistic way.
- Majority effort in building an agentic solution would go into building the planner.



Docket

Planner

Planner works wonder with reflection or critique flow.



Docket

Tools

- Interaction medium for agents with the world.
- They come in many forms -
  - Knowledge Augmenters
    - Web Search
    - Database Search (retrieval tasks, etc.)
  - Interaction Mediums
    - Web Browser
    - Code Interpreter
  - Content Creators
    - Website Creation
    - Text/Image/Video Generation

Docket

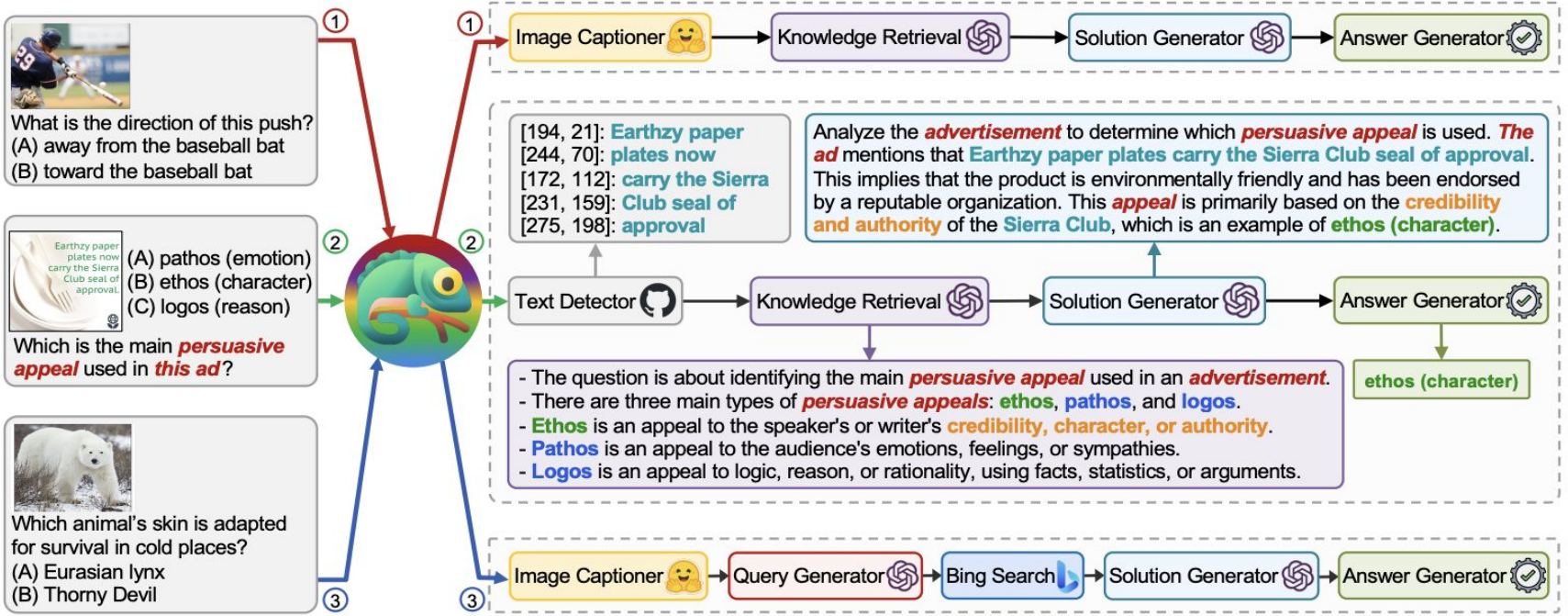
Tools

- Few examples of Tools from Chameleon Paper.

Tool Types	Tools
OpenAI	Knowledge Retrieval, Query Generator, Row Lookup, Column Lookup, Table Verbalizer, Program Generator, Solution Generator
Hugging Face	Image Captioner
Github	Text Detector
Web Search	Bing Search
Python	Program Verifier, Program Executor
Rule-based	Answer Generator

Docket

Agent in the Wild - Example - Chameleon Paper



Docket

Demo Slide

Docket Docs - Example

Document Creation Idea - Integrations at Docket

Docket

Working Principle for Agents

Agent's Should -	Agent's Shouldn't
Ensure Transparency	Overstep Human Oversight
Maintain Ethical Standards	Neglect Security and Privacy
Prioritize User Empowerment	Sacrifice Explainability for Efficiency
Enable Flexibility and Adaptability	

Docket

References

1. Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models [https://arxiv.org/abs/2304.09842]
2. Building Effective Agents [https://www.anthropic.com/research/building-effective-agents]
3. Agents section of AI Engineering (2025) by Chip Huyen [https://huyenchip.com/2025/01/07/agents.html]
4. SmolAgents Library by Hugging Face [https://github.com/huggingface/smolagents]
5. ReAct Prompting [https://react-lm.github.io/]

Get in touch

- https://www.linkedin.com/in/rishav-chandra-varma/
- https://github.com/reichenbch

Thank You!