

# LLMs and Tool Use

Karim Lalani

# What is Tool Use?

Tool use allows generative AI applications to interact with external applications, tools, and APIs.

These can be:

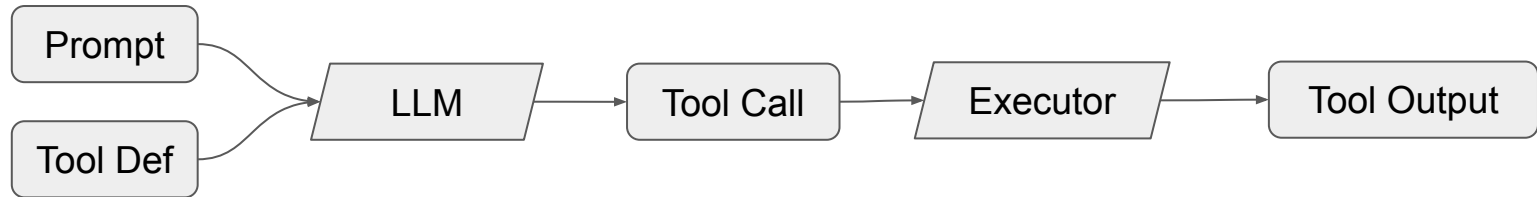
1. Python libraries
2. External APIs
3. Custom code



# How Tool Use works?

Tool use works as follows:

1. You provide the tool definition as a JSON schema to the LLM along with your prompt.
2. LLM transforms your prompt into JSON output based on the JSON schema provided.
3. A tool executor can then process this JSON output and handle the tool execution and response retrieval for further processing.



# What makes Tool Use possible?

Tool-use capabilities depend on two factors:

1. Can the model generate JSON output conforming to JSON schema?
2. Can the inference engine preprocess tool definitions and postprocess tool use responses?

# From OllamaFunctions to tool-calling-llm

## OllamaFunctions

- an experimental tool use wrapper around *ChatOllama* from *langchain-ollama*.

## ToolCallingLLM

- *OllamaFunctions* refactored and abstracted into reusable Python mixin.
- Can work with any *ChatModel* from LangChain. Not limited to Ollama.

## tool-calling-llm

- Python package published on pypi.org

# tool-calling-llm

Python package

<https://pypi.org/project/tool-calling-llm/>

*pip install tool-calling-llm*

Source code

[https://github.com/lalanikarim/tool\\_calling\\_llm](https://github.com/lalanikarim/tool_calling_llm)

Notebook

[https://github.com/lalanikarim/tool\\_calling\\_llm/blob/main/ToolCallingLLM-Tutorial.ipynb](https://github.com/lalanikarim/tool_calling_llm/blob/main/ToolCallingLLM-Tutorial.ipynb)

● ● ● tool-calling-llm.py

```
1 from tool_calling_llm import ToolCallingLLM
2 from langchain_ollama import ChatOllama
3 from langchain_community.tools import DuckDuckGoSearchRun
4
5
6 class OllamaWithTools(ToolCallingLLM, ChatOllama):
7     def __init__(self, **kwargs):
8         super().__init__(**kwargs)
9
10    @property
11    def _llm_type(self):
12        return "ollama_with_tools"
13
14
15 llm = OllamaWithTools(model="llama3.1", format="json")
16 tools = [DuckDuckGoSearchRun()]
17 llm_tools = llm.bind_tools(tools=tools)
18
19 llm_tools.invoke("Who won the silver medal in shooting in the Paris Olympics in 2024?")
```

● ● ● tool-calling-llm.py

```
1 AIMessage(  
2     content='',  
3     id='run-9c3c7a78-97af-4d06-835e-aa81174fd7e8-0',  
4     tool_calls=[  
5         {  
6             'name': 'duckduckgo_search',  
7             'args': {  
8                 'query': 'Paris Olympics 2024 shooting silver medal winner'  
9             },  
10            'id': 'call_67b06088e208482497f6f8314e0f1a0e',  
11            'type': 'tool_call'  
12        }  
13    ]  
14 )
```



# Paris Olympics 2024 shooting silver medal winner?

Next slide has the answer

# Thank You

