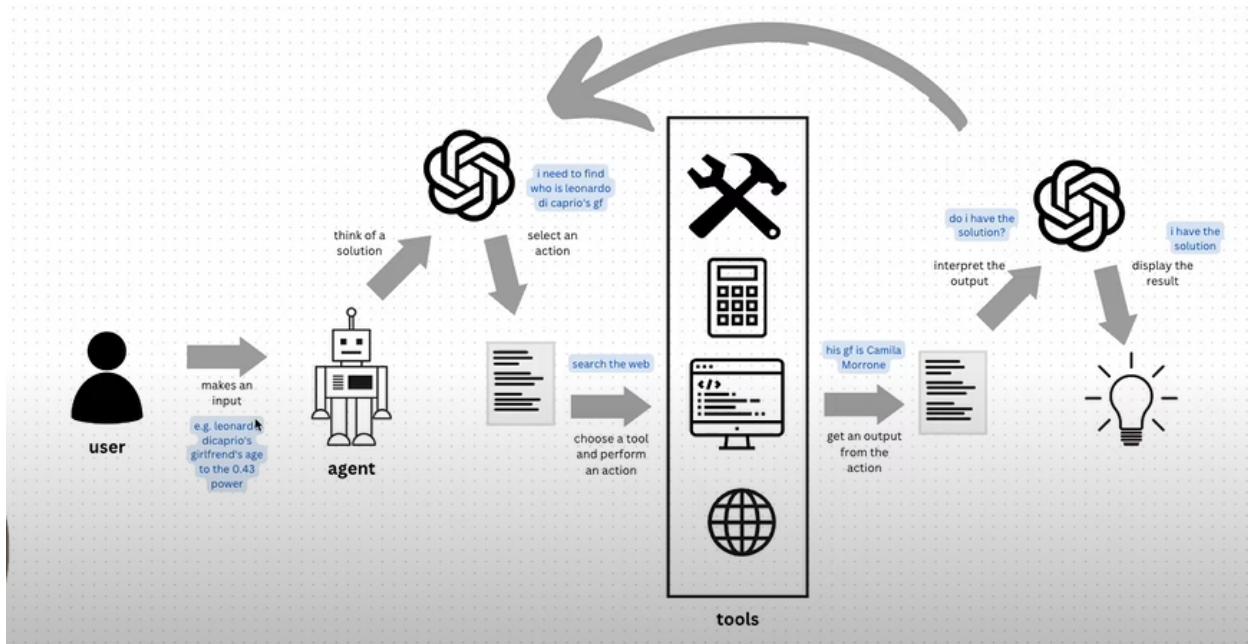


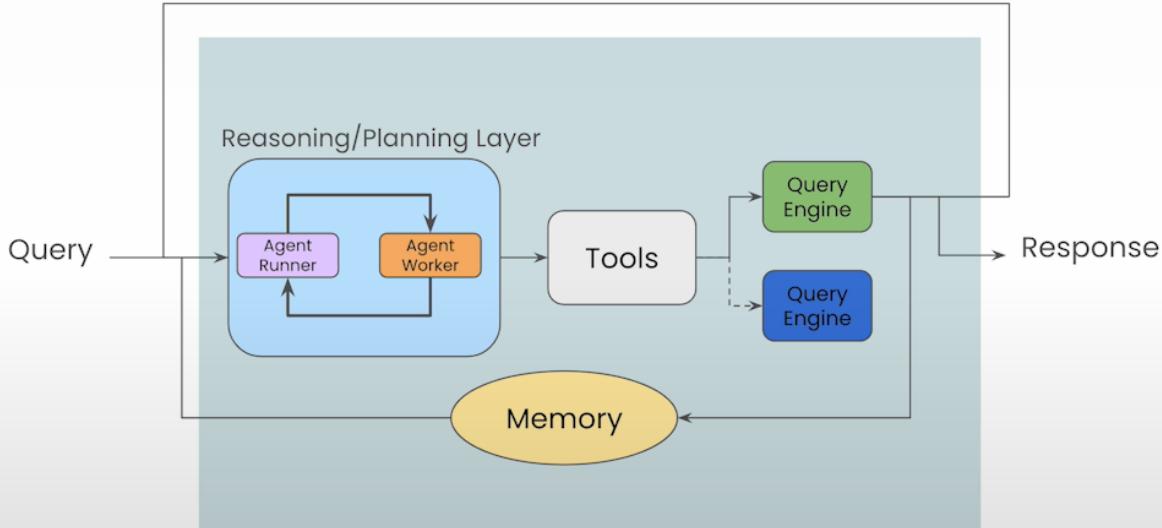
LangChain Agents



LLAMAINDEX AGENTS

Full agent reasoning loop

Full Agent Reasoning Loop

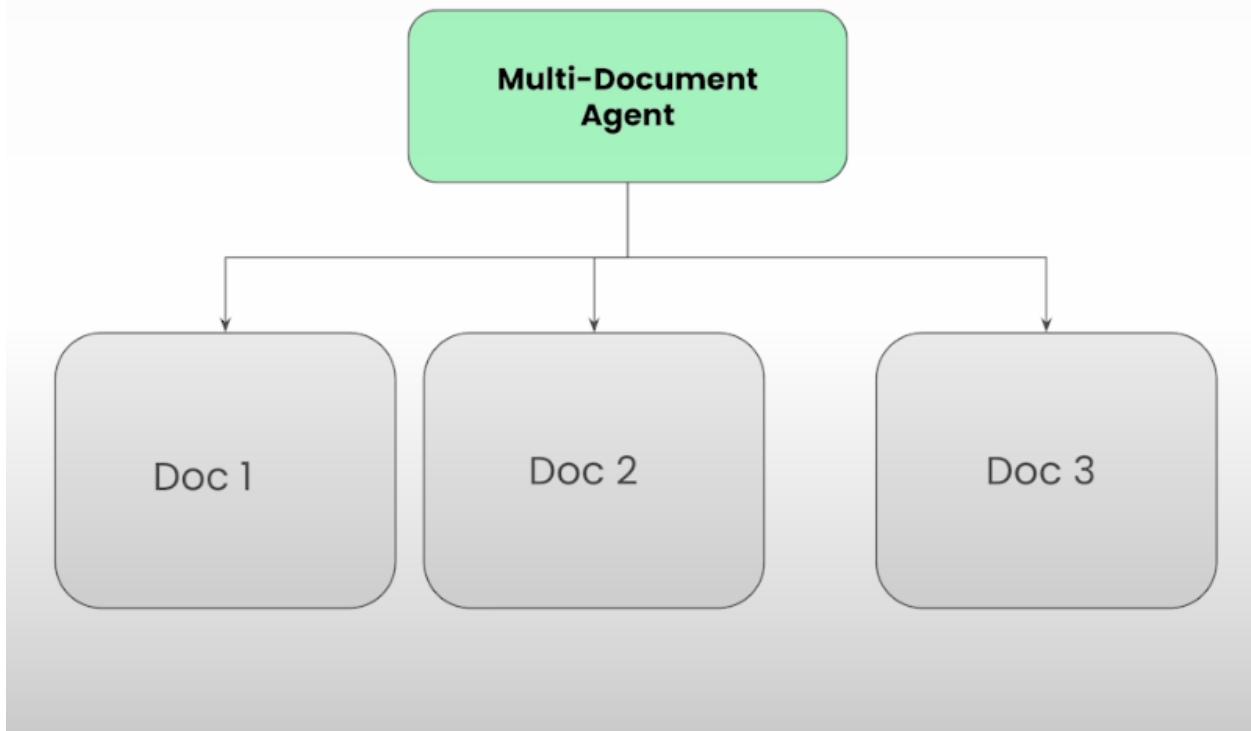


Agent Control

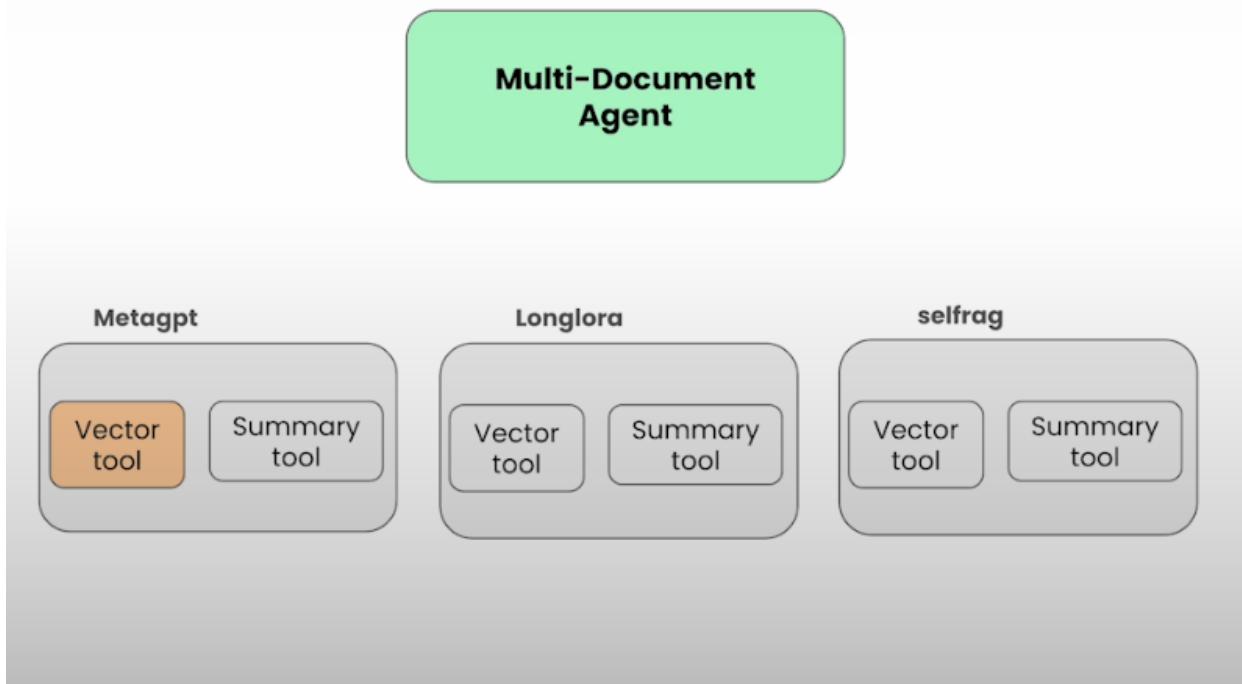
The key benefits:

- **Decoupling of Task Creation and Execution:**
Users gain the flexibility to schedule task execution according to their needs.
- **Enhanced Debuggability:** Offers deeper insights into each step of the execution process, improving troubleshooting capabilities.
- **Steerability:** Allows users to directly modify intermediate steps and incorporate human feedback for refined control.

Multi-Document Agent



Three-documents Agent



Agent worker includes all these 6 tools and as well as the LLM .

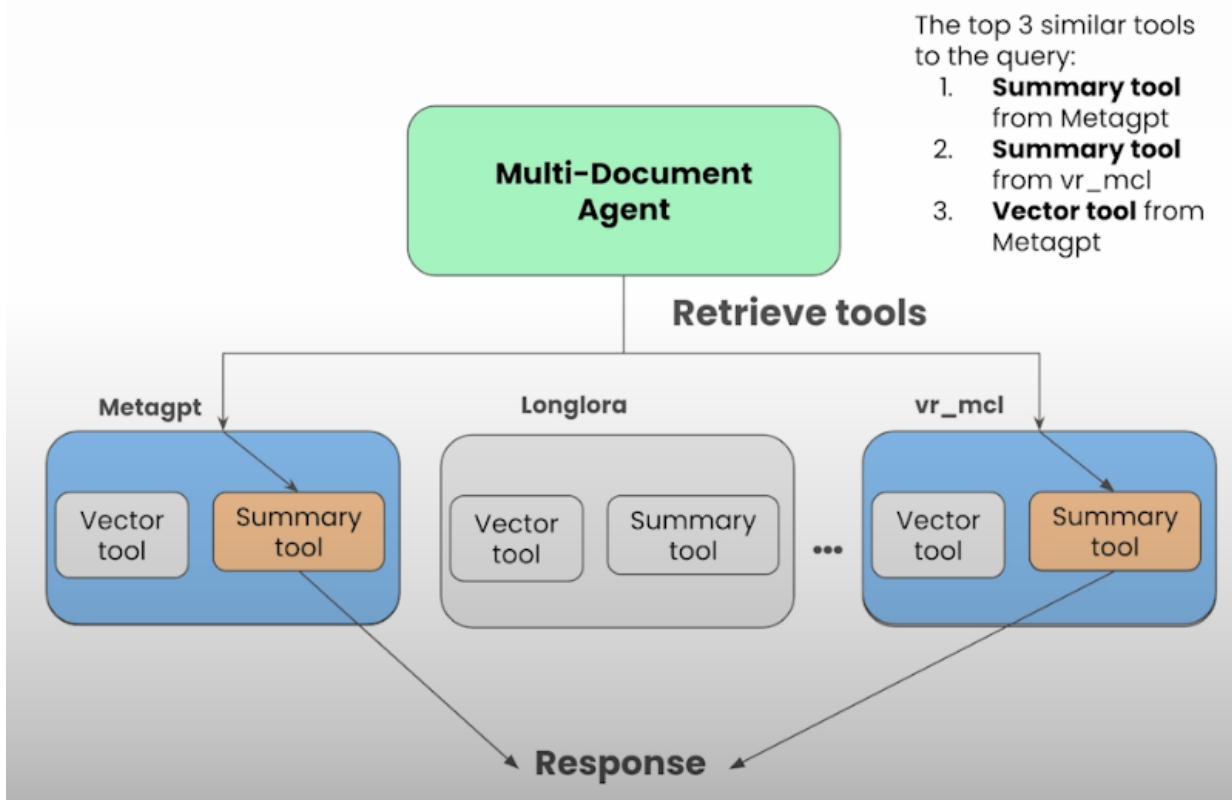
Retreival Augmented Generation at the level of Agent Tools

let's say if had 11 documents and there were two tools for each of those document and there were like 22 tools which are too more and LLM can confuse which tool to select to solve this problem

we EXTEND THE AGENT WITH TOOL RETRIEVAL.

Eleven-documents Agent

Query: Summarize the Metagpt and the VR_MCL papers.



* Building A generic RAG
↳ a framework to help you build Research agents.

* Autonomous Research Agent.

Building a full agent:

Step 1: Routing

We add decision making to route requests to multiple tools.

Step 2: Tool use.

where you create an Interface(ux) for agents to select a tool as well as generate right arguments for that tool.

Step 3: Multi step reasoning with tool use.

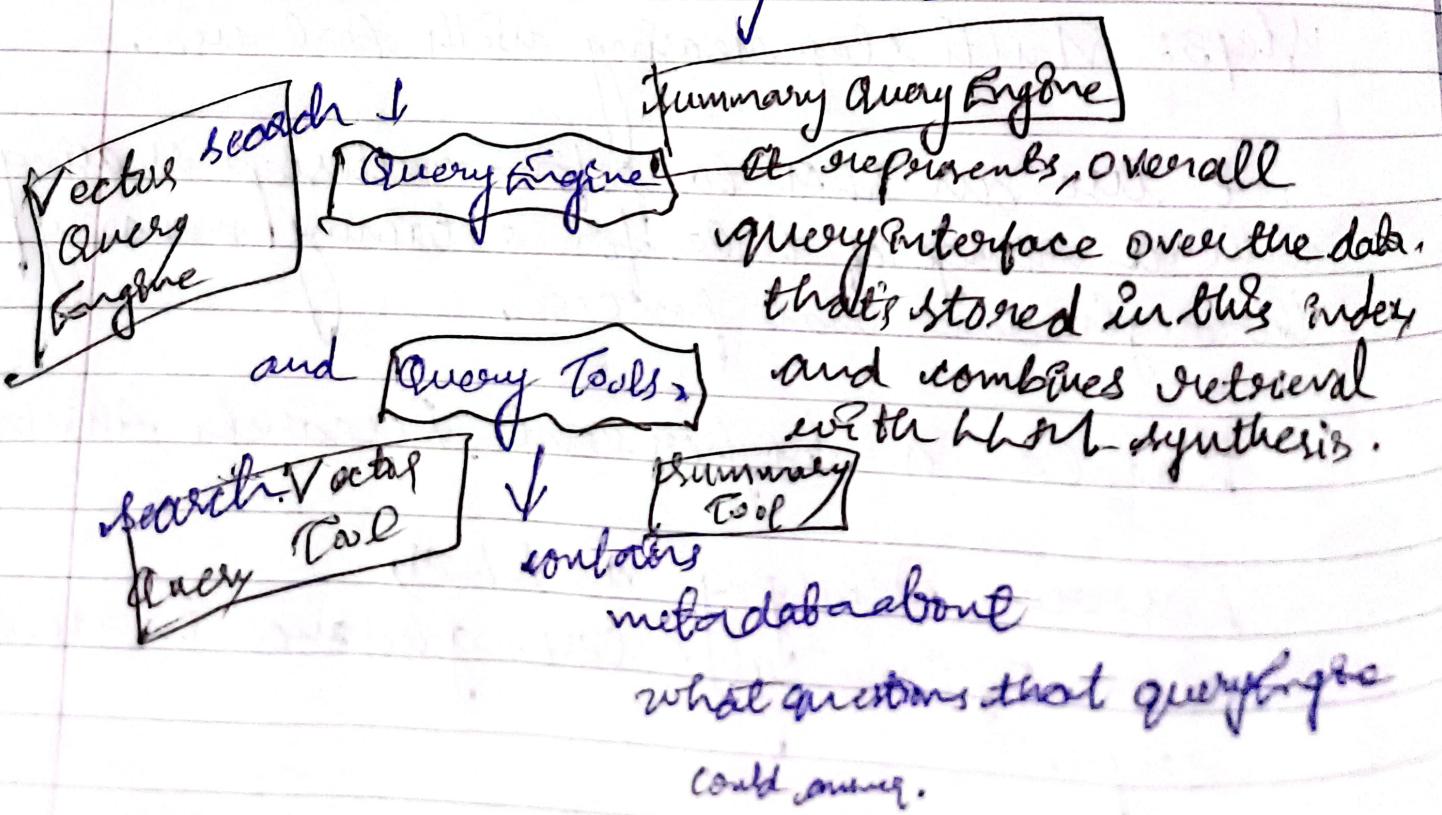
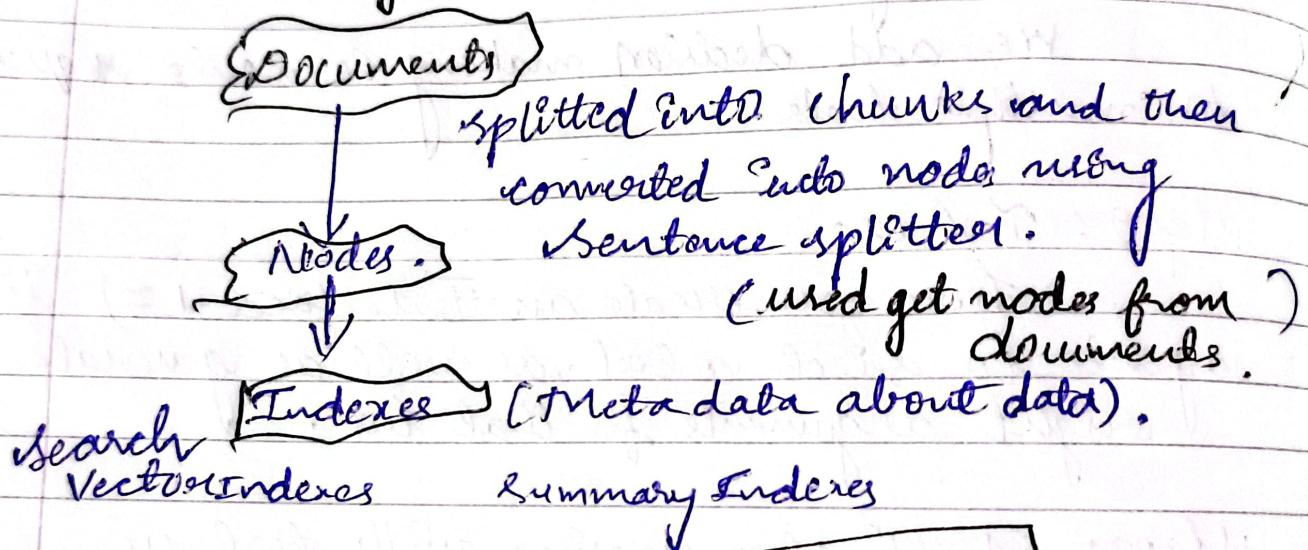
We use an LLM to perform multi-step reasoning with a range of tools for retaining memory throughout out that process.

Create a higher level Research Assistant.

Ensure Debugging on LLM,
user optionally enter guidance in middle steps.

* Building Router Over a single document
that could handle both question and
Answering and summarization.

* Router Engine. simple Directory Reader to read



Router → How to use LLM's to make a decision by picking a choice of different pipelines.

* Selectors to build Router. choice of different pipelines has distinct attribute.

LLM selector: prompting an LLM to output a JSON that is parsed, and the corresponding indices are queried.

Pydantic selector:

use the openai function calling API to produce pydantic selection objects rather than parsing raw JSON.

Defining Router Query Engines

Selectors and Query tools

are passed to

Router Query Engine.

This will select the right tool as per prompt.

* Tool Hailing.

Not only pick a function to execute but inference arguments to pass through the function.

* Tool calling continuations

One of the promises of our LHM's is their ability to take action and interact with External Environment.

But how do we use LHM's to take action and interact with external Environment?

* LHM's to take action and also interact with external environment.

* How? Tool calling.

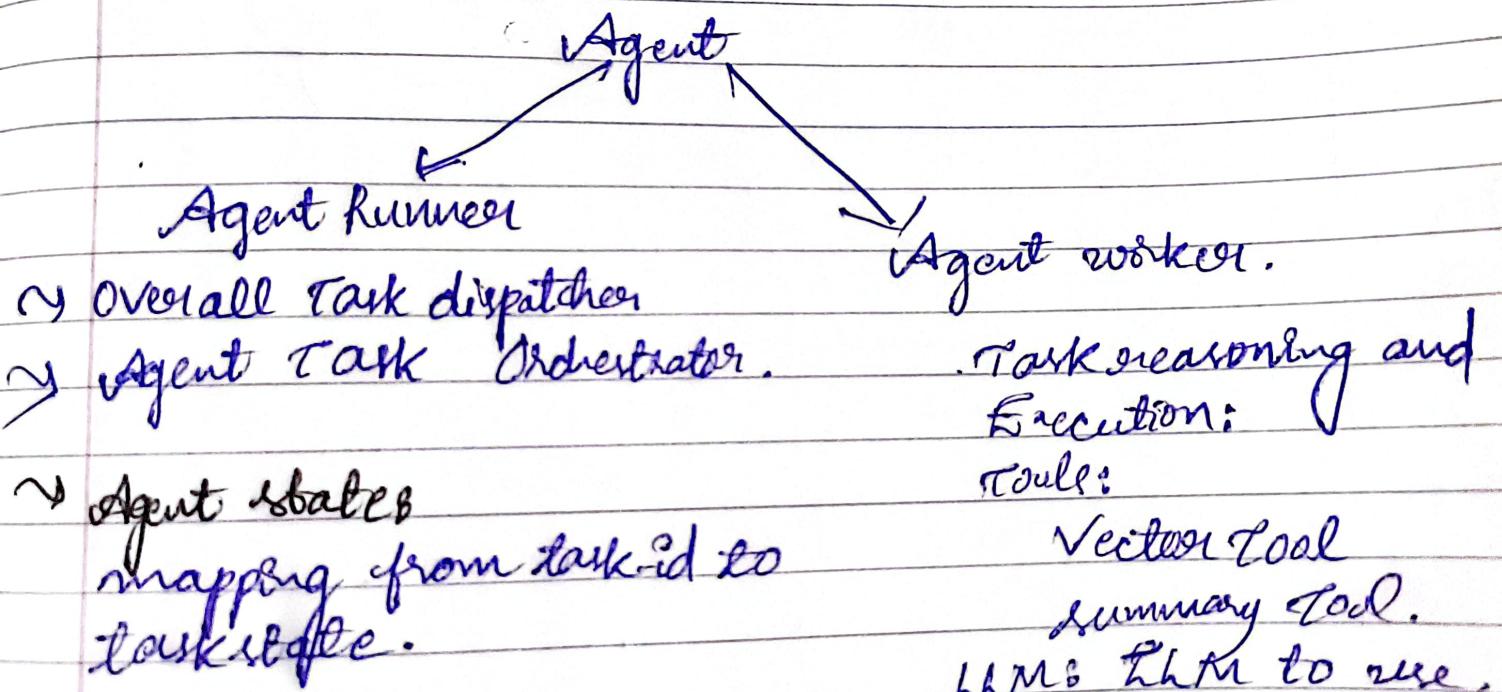
Tool calling helps LHM's to interact with external environment through a dynamic interface where tool calling helps :-

- 1) choosing the right tool
e.g inference object argument
predict and call²
function not only helps picking out
also helps in setting parameters,

Function¹call

- * Building an Agent Reasoning Tool.
- * agent reasoning loop over tools to pick the right one for multiple steps.
- * Function calling

Agent in ThamaIndex.



Task statee

- Task
- Completed steps.
- step Queue.

Memory

Conversation Memory.

shared message bus &
subscription mechanism