

Introduction to Langchain

Bryan Figuez -
Software Engineer
El Paso Labs



Agenda

- ★ Why should I do A.I.
- ★ What is LangChain
- ★ What is LangGraph
- ★ What is LangSmith
- ★ What is Streamlit
- ★ A.I. Agent Tools



1990s

Why should I do A.I.?

- ★ High Demand and Career Opportunity: A.I. opens opportunities to explore a variety of career paths
- ★ AI is Transforming Every Industry: medical, finance, cybersecurity are some examples of industries impacted by A.I.
- ★ AI is the Future of Technology: AI is driving the next wave of technological advancements



So what is LangChain?



LangChain

LangChain is used to build AI-powered applications that integrate large language models (LLMs) with external data sources, tools, and APIs for more dynamic and intelligent interactions.



LangChain

FRAMEWORK FOR BUILDING AI-POWERED APPLICATIONS

LangChain is a powerful framework designed to help developers build applications that integrate with LLMs

MODULAR AND EXTENSIBLE

It provides pre-built components like prompt templates, memory, chains, and agents

INTEGRATION WITH EXTERNAL TOOLS AND DATA SOURCES

LangChain enables AI models to interact with APIs, databases, and external knowledge bases



LangChain Features

- ★ LangGraph
- ★ LangSmith
- ★ Agent Tools



LangGraph



LangGraph

LangGraph is a framework for building stateful, multi-step AI workflows and agent-based systems using graphs, enabling complex decision-making and reasoning with large language models.



LangGraph

GRAPH-BASED AI WORKFLOW FRAMEWORK

LangGraph allows developers to design AI workflows as directed graphs, enabling structured and multi-step decision-making processes.

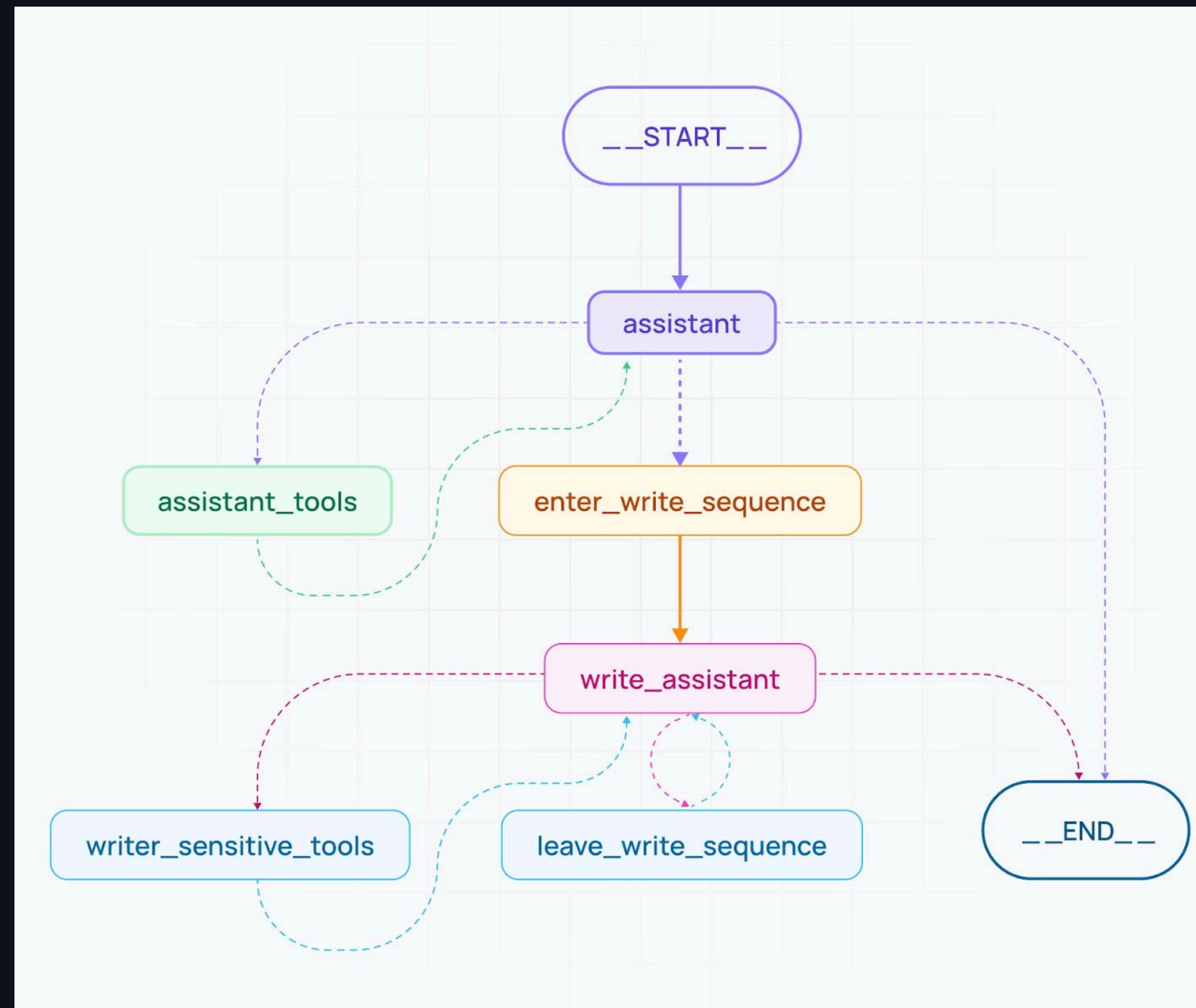
STATEFUL AND MULTI-AGENT CAPABILITIES

It supports persistent state management and coordination between multiple AI agents

FLEXIBLE AND SCALABLE

LangGraph integrates with LLMs and external tools, providing a scalable way to build intelligent agents





Multi-Agent strategies

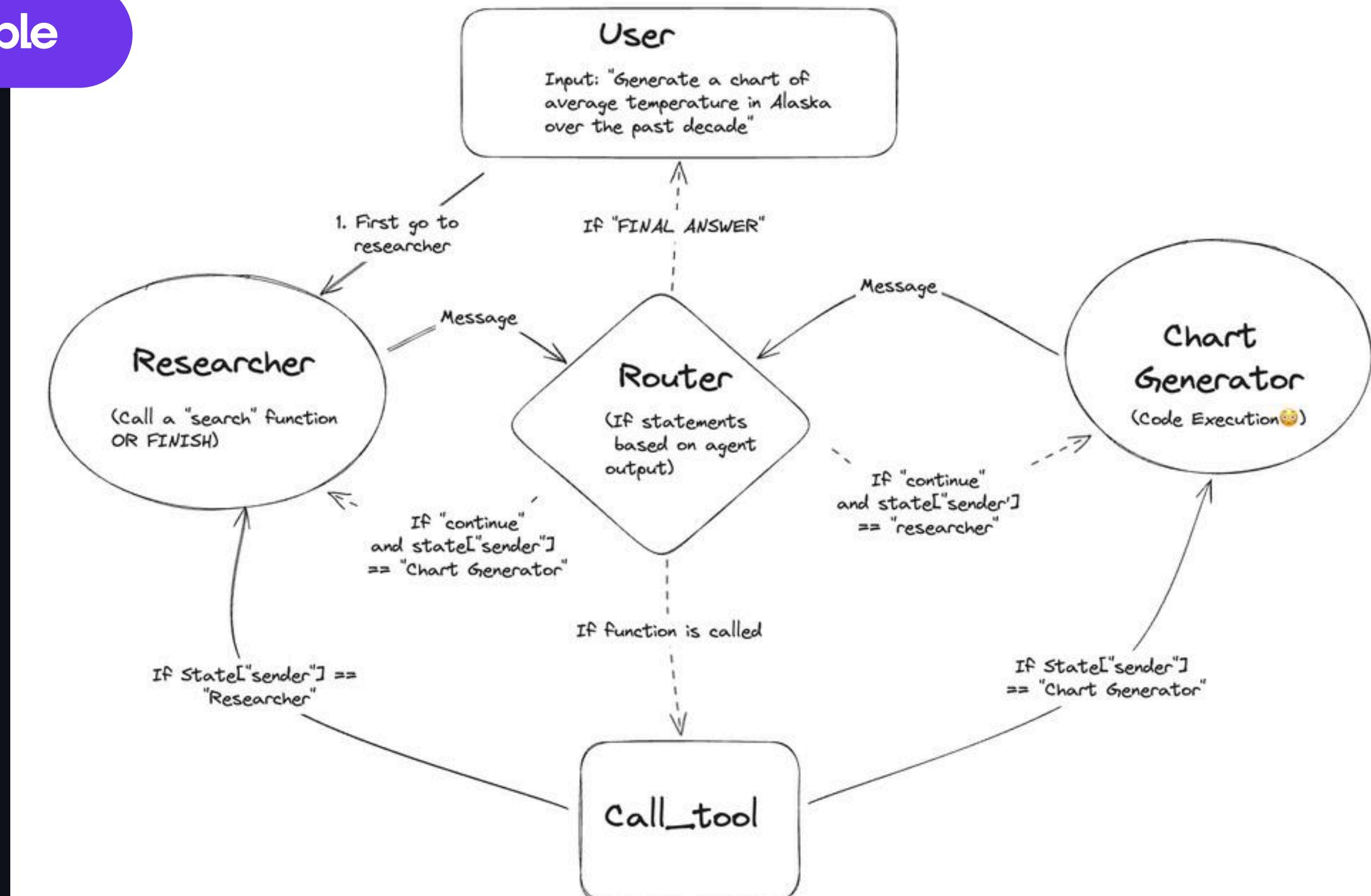


Multi-Agent

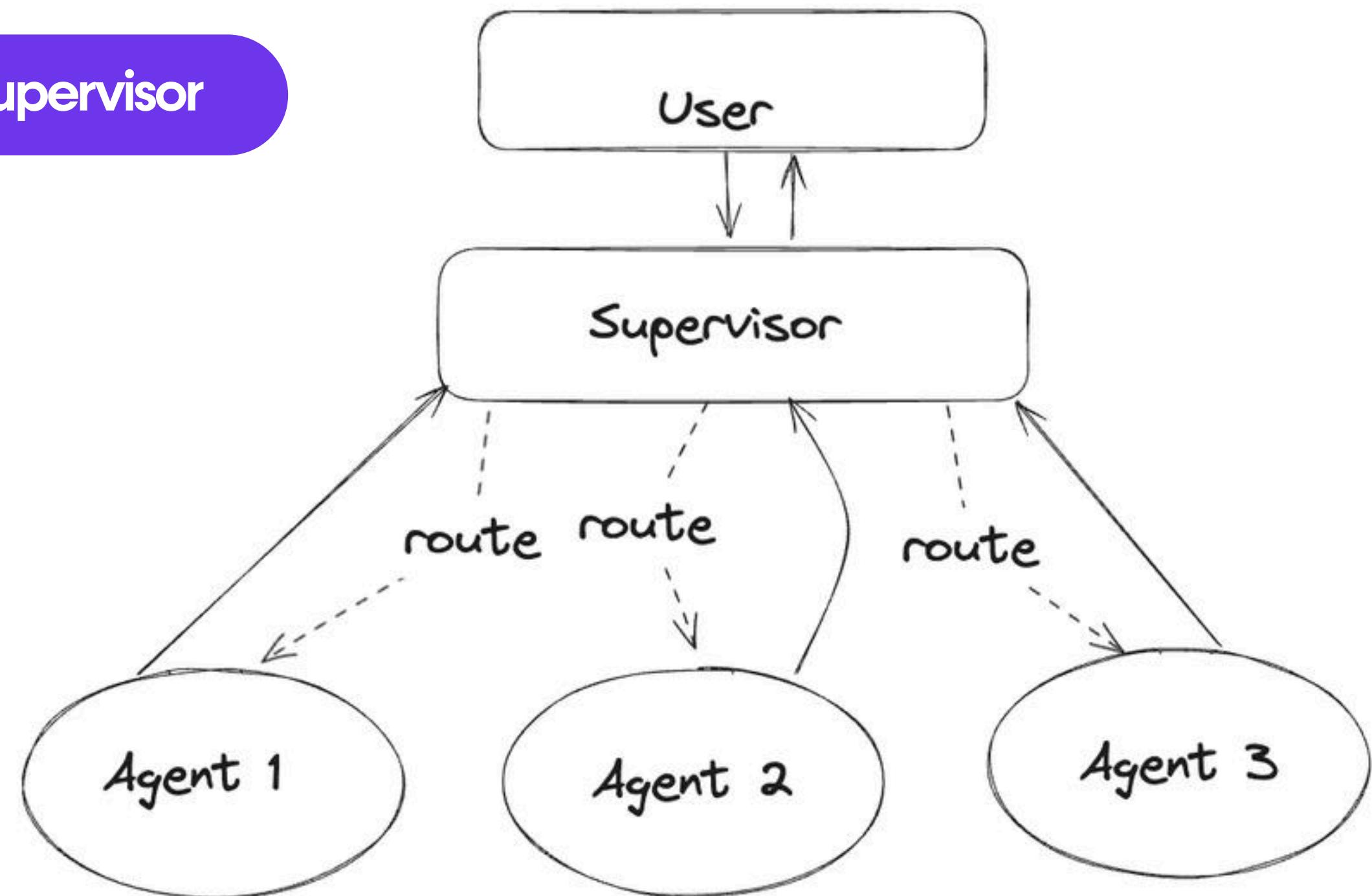
As the name suggests it, they are multiple agents with their own set of prompt, LLM, and tools that collaborate together.



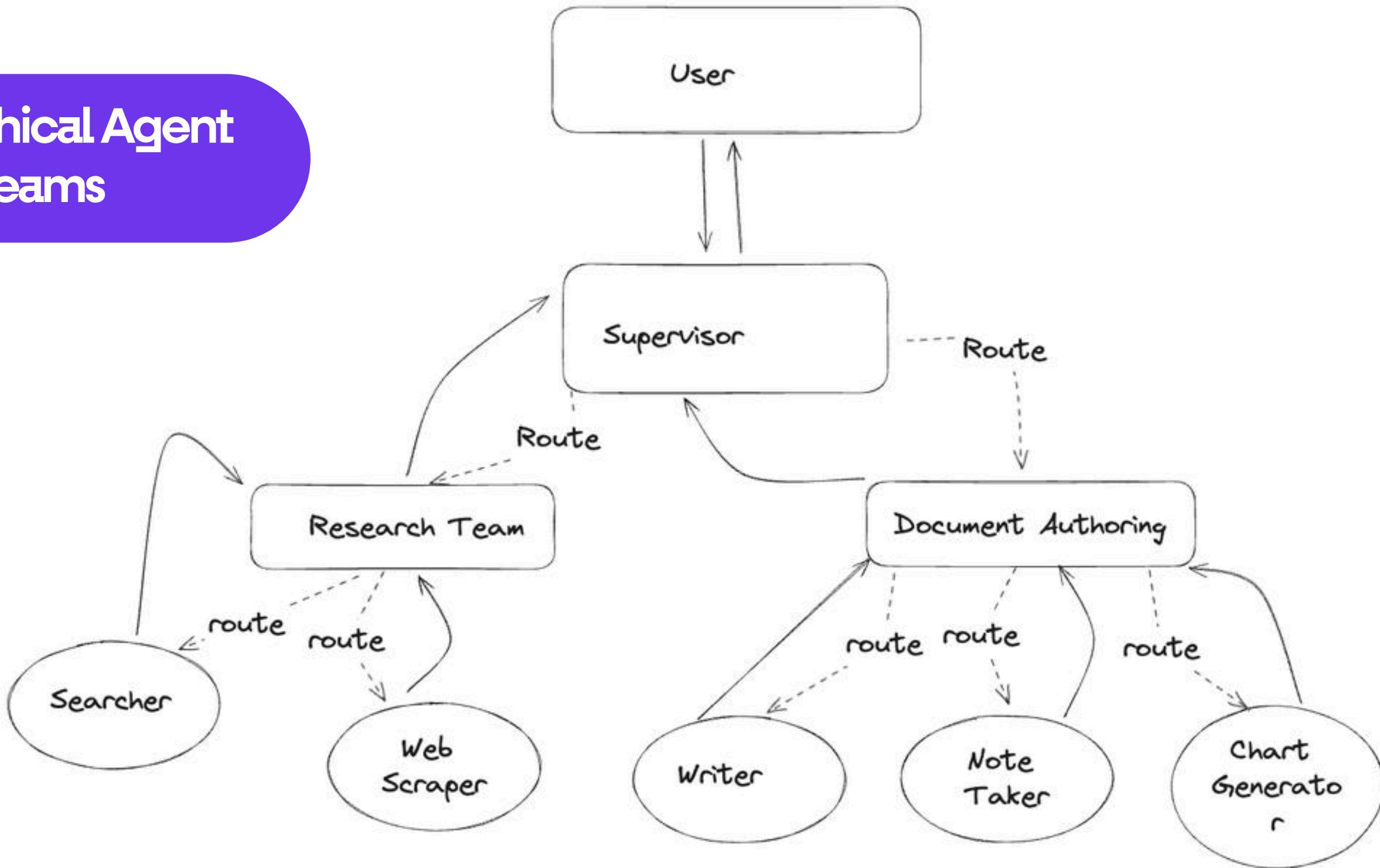
Example



Agent Supervisor



Hierarchical Agent Teams



LangSmith



LangSmith

**LangSmith is a debugging,
monitoring, and evaluation
platform for AI applications built
with LangChain, helping
developers improve performance,
trace errors, and optimize LLM-
based workflows.**



LangSmith

DEBUGGING AND MONITORING TOOL

LangSmith provides detailed tracing and logging to help developers diagnose issues and optimize AI application performance.

EVALUATION AND TESTING FRAMEWORK

It enables systematic evaluation of language model outputs, ensuring reliability and quality in AI-driven applications.

SEAMLESS INTEGRATION WITH LANGCHAIN

Designed to work with LangChain, LangSmith enhances development by offering insights into prompt effectiveness, latency, and model behavior.



Example

The screenshot displays the LangGraph interface, which integrates a trace viewer and an AI interaction log.

Left Panel (Trace):

- Header: TRACE
- Buttons: Waterfall, Show All, and a search icon.
- Log entries:
 - LangGraph (3.13s, 460 events)
 - _start_ (0.00s)
 - fetch_user_info (0.00s)
 - ChannelWrite<fetch_user_info,message> (0.00s)
 - primary_assistant (3.10s)
 - RunnableSequence (3.09s)
 - ChatPromptTemplate (0.00s)
 - AzureChatOpenAI (gpt-4o-mini) (3.09s)
 - ChannelWrite<primary_assistant,message> (0.00s)
 - route_primary_assistant (0.00s)

Right Panel (LangGraph):

- Header:** LangGraph (ID: 1234567890)
- Run Tab:** Shows the current run status.
- Feedback Tab:** Placeholder for user feedback.
- Metadata Tab:** Placeholder for metadata.
- Input Section:**
 - User Info (expanded):
 - MESSAGES (messages: 2):
 - AI: Hello, how can I help you?
 - HUMAN: Hello, I would like help with my history research about WWII
- Output Section:**
 - AI: Hello, how can I help you?
 - HUMAN: Hello, I would like help with my history research about WWII

StreamLit



Streamlit

Streamlit is an open-source Python framework for quickly building and deploying interactive web applications, especially for data science and machine learning projects.



Streamlit

SIMPLE PYTHON SYNTAX

- ★ Streamlit allows developers to create a custom UI only using Python, without CSS or HTML

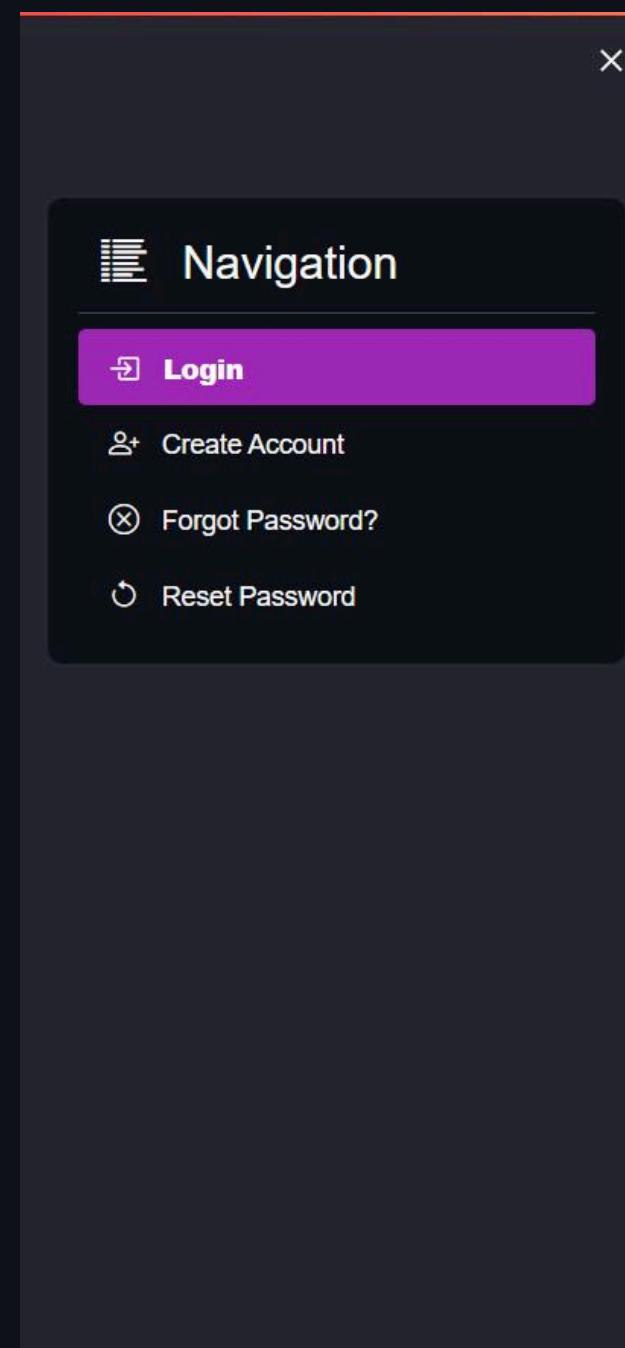
EVALUATION AND TESTING FRAMEWORK

- ★ It enables systematic evaluation of language model outputs, ensuring reliability and quality in AI-driven applications.

SEAMLESS INTEGRATION WITH LANGCHAIN

- ★ Designed to work with LangChain, LangSmith enhances development by offering insights into prompt effectiveness, latency, and model behavior.





Made with Streamlit

Username

Your unique username

Password

Your password

Login

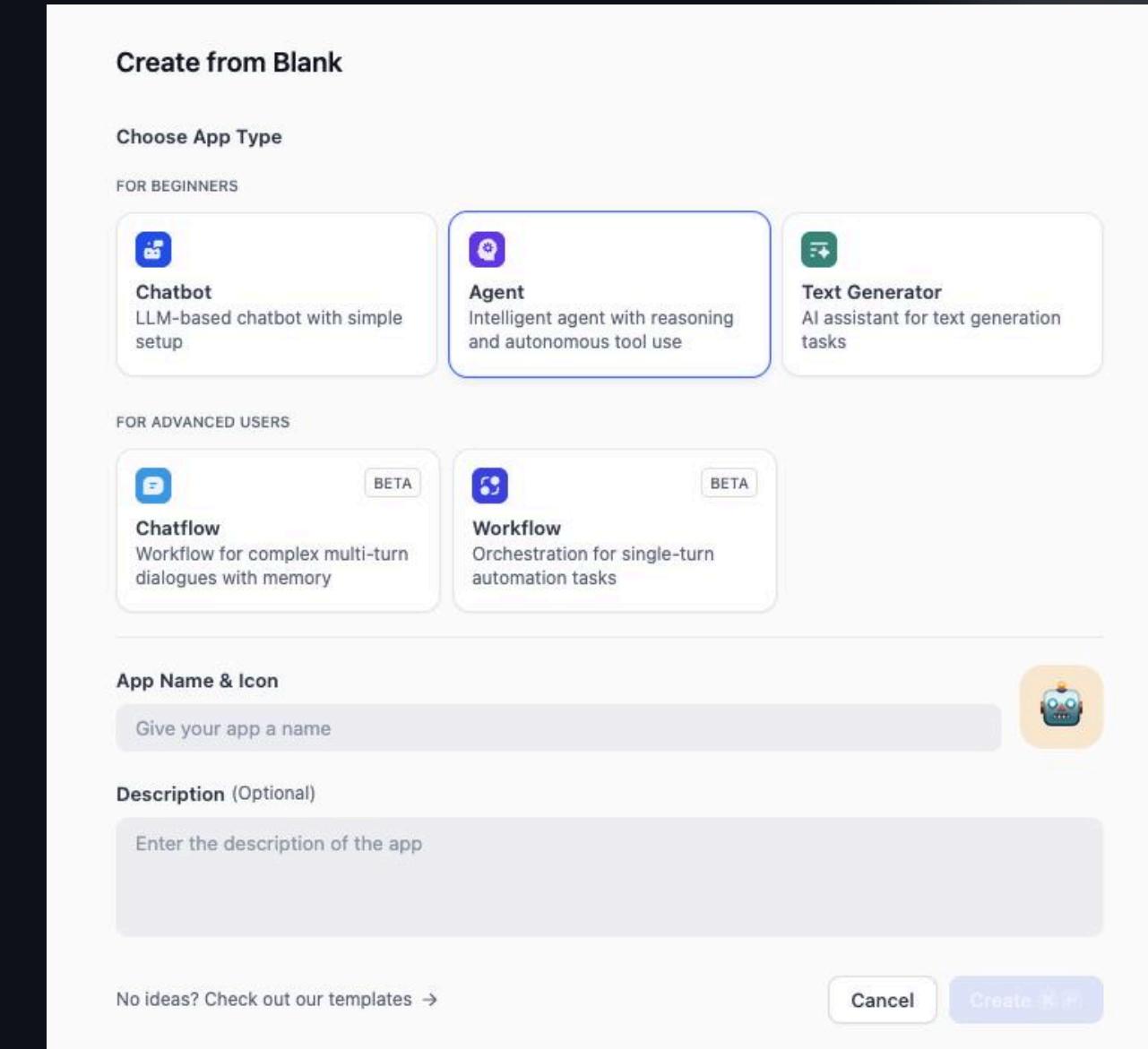


How to build an agent with Dify



STEP 1

★ **Create a blank agent project
in the studio page**



STEP 2

★ Create a prompt and set some variables that you'll need for your agent

INSTRUCTIONS ⓘ

```xml  
<instruction>  
You are tasked with distilling meetings into concise summaries. Your summary should include discussion topics, key takeaways, and action items. Follow the steps below to complete the task:  
1. Identify and list the main discussion topics covered in the meeting.  
2. Summarize the key takeaways from each discussion topic.  
3. List any action items that were decided upon during the meeting, including who is responsible for each action item and any relevant deadlines.  
Ensure that your output is clear, concise, and free of any XML tags.  
<example>  
<input>  
Meeting Date: 2023-10-01  
2349

**Variables** ⓘ

+ Add

|                                           |                 |
|-------------------------------------------|-----------------|
| {x} meeting_date · meeting_date           | REQUIRED string |
| {x} attendees · attendees                 | REQUIRED string |
| {x} discussion_topics · discussion_topics | REQUIRED string |
| {x} key_takeaways · key_takeaways         | REQUIRED string |
| {x} action_items · action_items           | REQUIRED string |



STEP 3

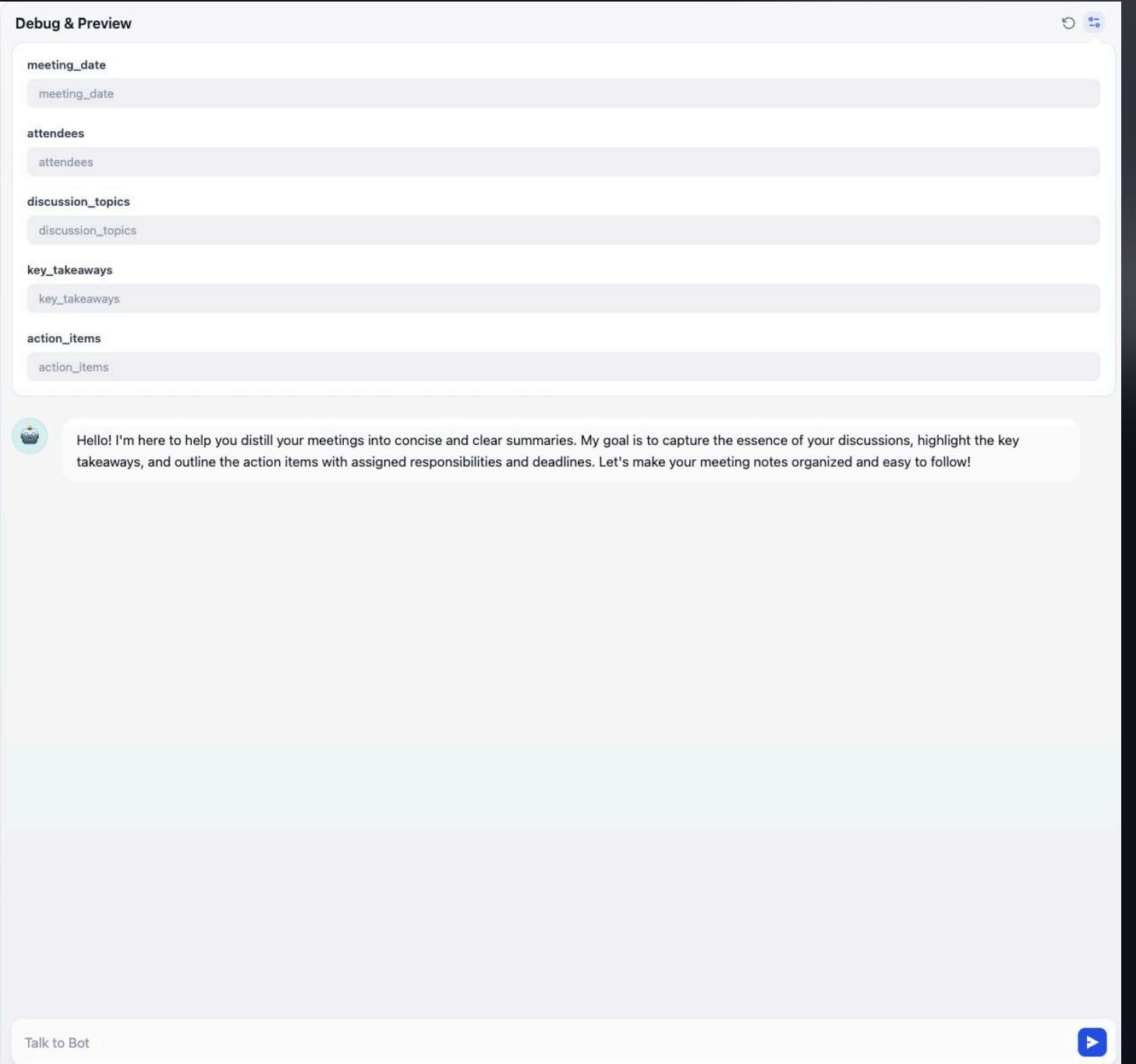
★ Set up your models in the settings page under model provider

The screenshot shows the 'Model Provider' settings page within a workspace. The left sidebar lists 'WORKSPACE' and 'GENERAL' sections, with 'Model Provider' selected. The main area displays four model providers:

- OpenAI**: Includes tabs for LLM, TEXT EMBEDDING, SPEECH2TEXT, MODERATION, and TTS. Shows QUOTA 0, API-KEY red, 0 Message Credits, and a 'Setup' button. An 'Add Model' button is also present.
- ANTHROPIC**: Shows QUOTA 0, API-KEY red, 0 Tokens, and a 'Setup' button. An 'Add Model' button is also present.
- Azure OpenAI Service Model**: Includes tabs for LLM, TEXT EMBEDDING, SPEECH2TEXT, and TTS. Shows 3 Models > and an 'Add Model' button.
- Cohere**: Includes tabs for LLM, TEXT EMBEDDING, and RERANK. Shows API-KEY green, a 'Setup' button, and an 'Add Model' button.

At the top right, there is a 'System Model Settings' button.





STEP 4

★ **Lastly, test and fine tune your agent to meet your projects standards**



**STEP 5**



**There are additional features/tools that could be added to your chatbot such as text2speech or image scraping**

**Features**  
Enhance web app user experience

**Conversation Opener**  Hello! I'm here to help you distill your meetings into concise and clear summaries. My goal is to capture the essence of yo...

**Follow-up**  Setting up next questions suggestion can give users a better chat.

**Text to Speech**  Conversation messages can be converted to speech.

**Speech to Text**  Voice input can be used in chat.

**Citations and Attributions**  Show source document and attributed section of the generated content.

**Content moderation**  Secure model output by using moderation API or maintaining a sensitive word list.

**Annotation Reply**  You can manually add high-quality response to the cache for prioritized matching with similar user questions.

**Image**

| Image | Tool                                             | Description                                                                                 | Count                          |
|-------|--------------------------------------------------|---------------------------------------------------------------------------------------------|--------------------------------|
|       | DALL-E<br>langgenius / dalle                     | DALL-E art                                                                                  | 5,693 · # Image # Productivity |
|       | Azure DALL-E<br>langgenius / azuredalle          | Azure DALL-E art                                                                            | 792 · # Image # Productivity   |
|       | S. Stability<br>langgenius / stability           | Activating humanity's potential through generative AI                                       | 1,421 · # Image                |
|       | Stable Diffusion<br>langgenius / stablediffusion | Stable Diffusion is a tool for generating images which can be deployed locally.             | 2,119 · # Image                |
|       | ComfyUI<br>langgenius / comfyui                  | ComfyUI is a tool for generating images which can be deployed locally.                      | 3,285 · # Image                |
|       | Novita AI<br>langgenius / novitai                | Innovative AI for Image Generation                                                          | 536 · # Image # Productivity   |
|       | SiliconFlow<br>langgenius / siliconflow_tool     | The image generation API provided by SiliconFlow includes Flux and Stable Diffusion models. | 3,355 · # Image                |
|       | getimg.ai<br>langgenius / getimgai               | GetImg API integration for image generation and scraping.                                   | 889 · # Image                  |

