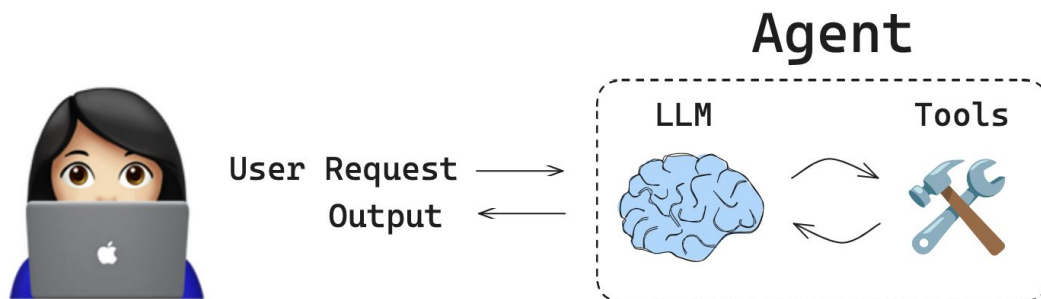


Building ambient agents with  LangGraph

# Chat



## What comes next?

	<b>Chat Agents</b>	<b>?</b>
<b>Agent Trigger</b>	User message / request	Listening to events

## What comes next?

	<b>Chat Agents</b>	<b>?</b>
<b>Agent Trigger</b>	User message / request	Listening to events
<b>Concurrency</b>	~1 user request	Many events

## What comes next?

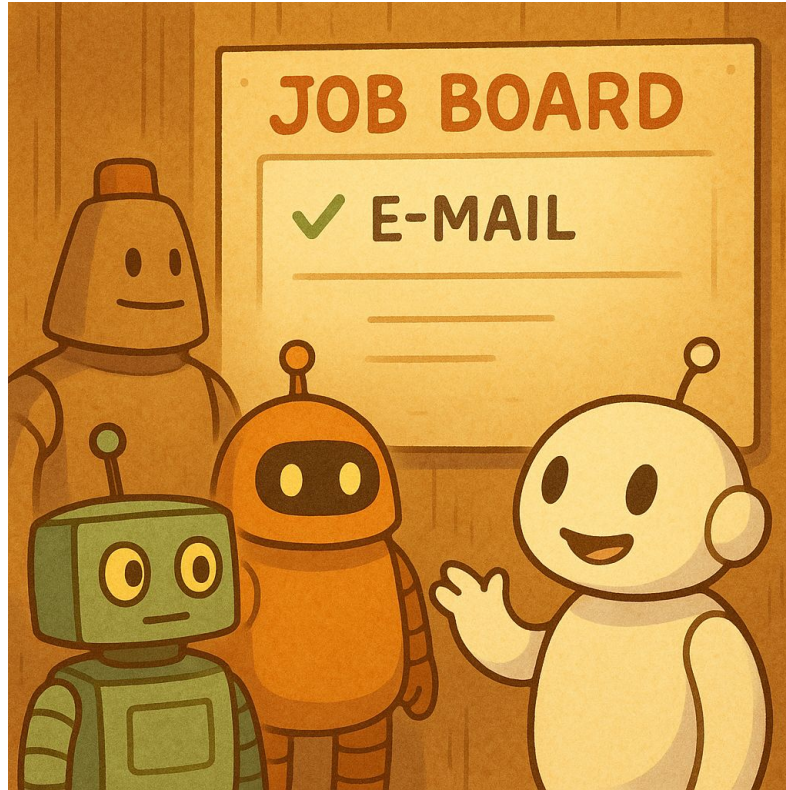
	<b>Chat Agents</b>	<b>?</b>
<b>Agent Trigger</b>	User message / request	Listening to events
<b>Concurrency</b>	~1 user request	Many events
<b>UX</b>	Chat UI / await completion	Background / notify me

## What comes next?

	<b>Chat Agents</b>	<b>?</b>
<b>Agent Trigger</b>	User message / request	Listening to events
<b>Concurrency</b>	~1 user request	Many events
<b>UX</b>	Chat UI / await completion	Background / notify me
<b>Latency Requirements</b>	Needs to be fast	Can take longer

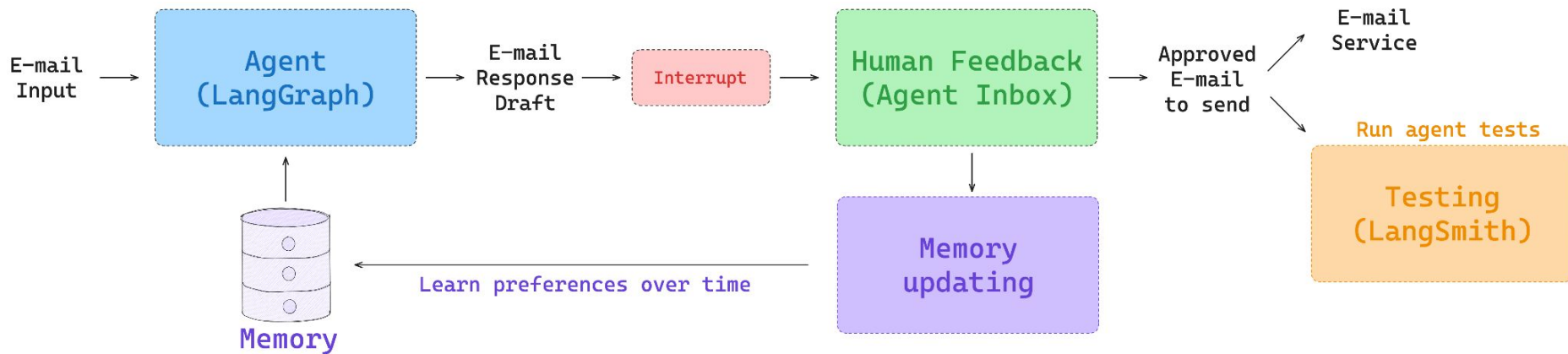
## “Ambient Agents”

	<b>Chat Agents</b>	<b>Ambient agents</b>
<b>Agent Trigger</b>	User message / request	Listening to events
<b>Concurrency</b>	~1 user request	Many events
<b>UX</b>	Chat UI / await completion	Background / notify me
<b>Latency Requirements</b>	Needs to be fast	Can take longer

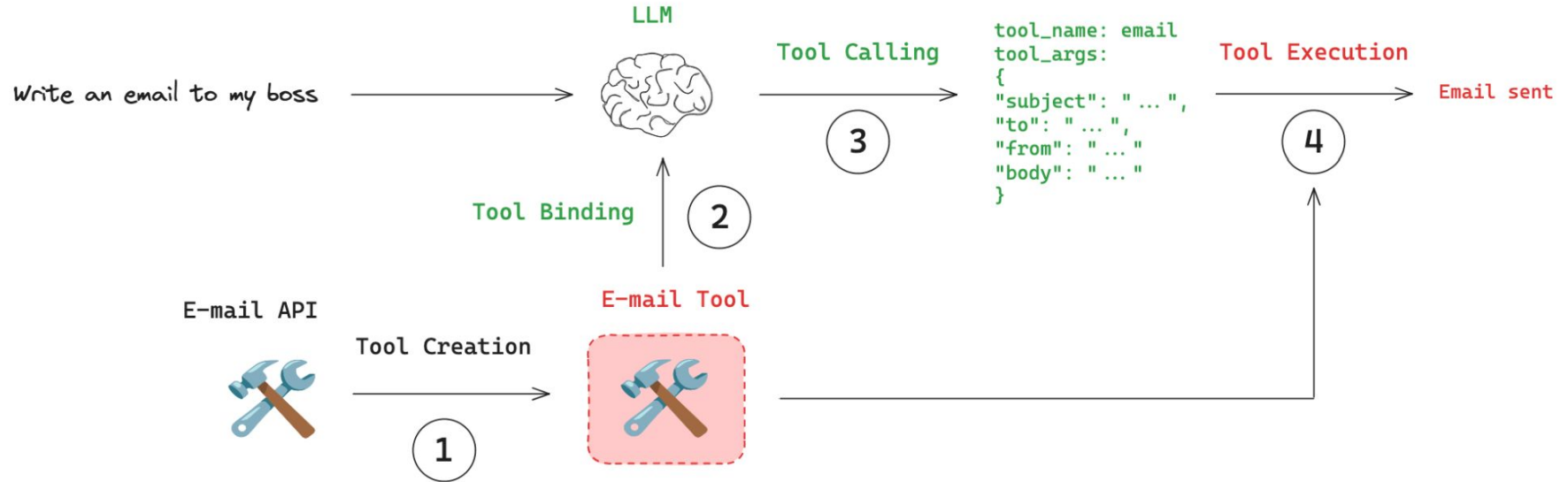




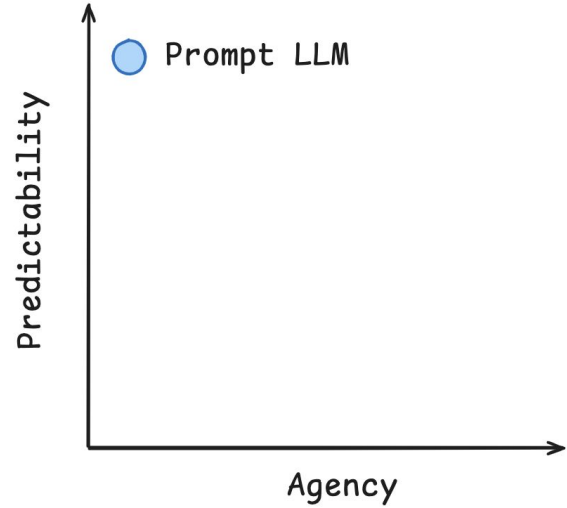
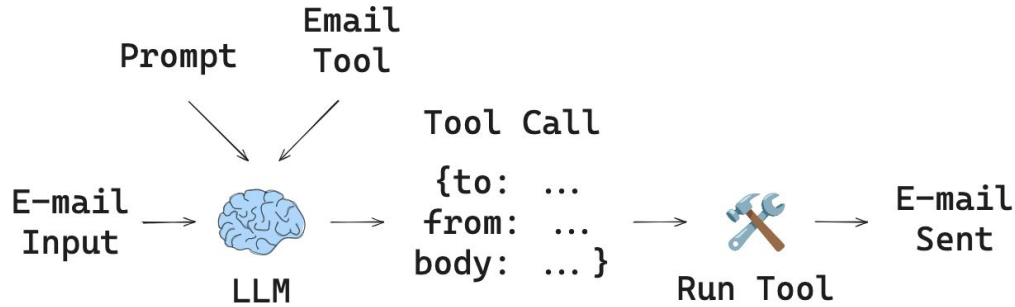
## Where we are going



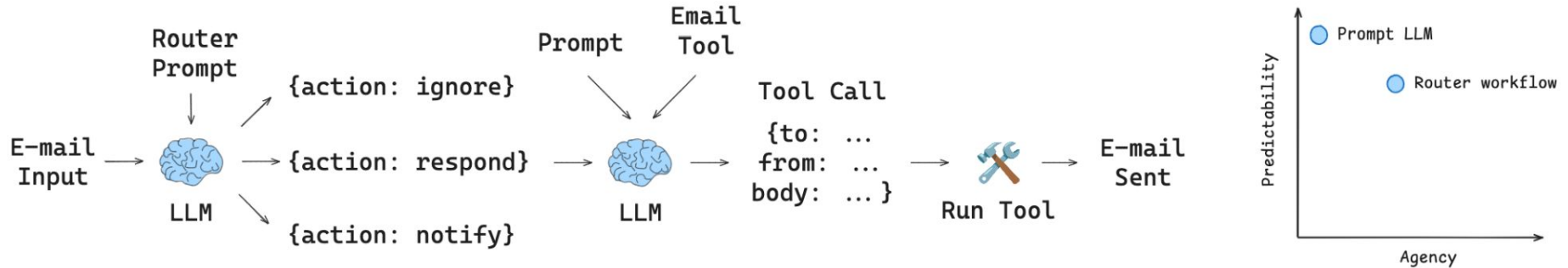
# Central concept is tool calling



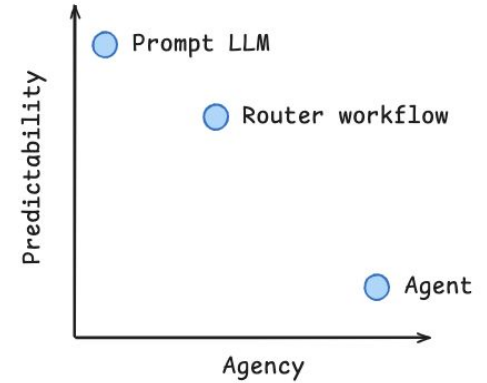
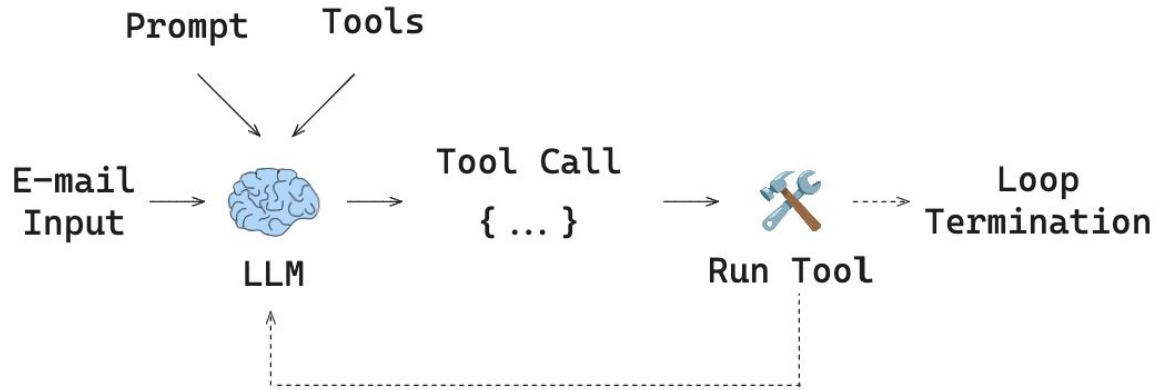
# The simplest possible e-mail assistant



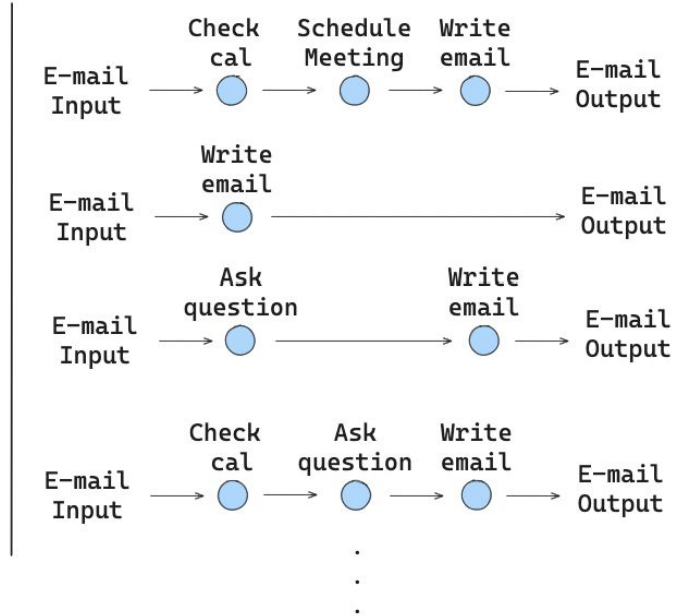
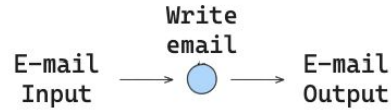
## Workflow: Make decisions along predefined code path



## Agent ~ tool calling in a loop!



## Agent can take many more actions

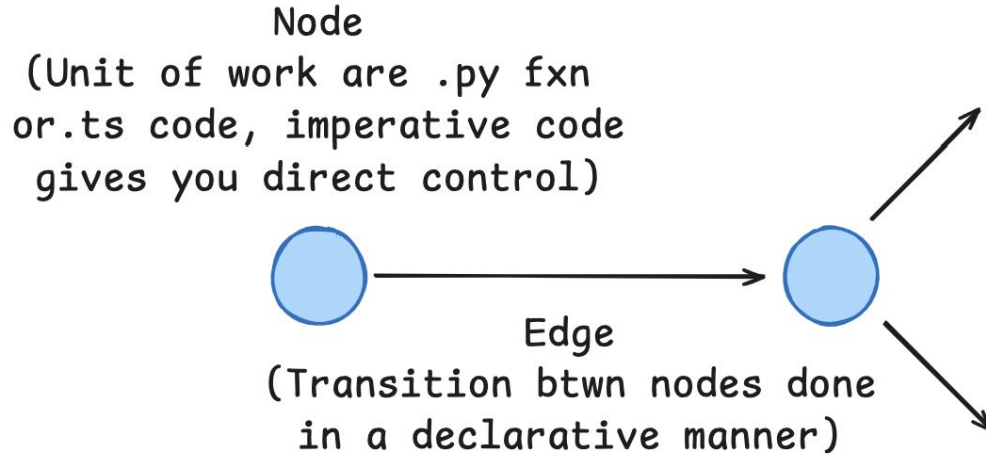


## Workflow v Agent considerations

	Workflow	Agent
<b>Action sequence</b>	Easy to enumerate in advance	Unknown until runtime (depends on input)
<b>Process</b>	Linear or branching control flow	Flexible decision making (backtracking)
<b>Performance</b>	Latency / cost are critical	OK to trade-off for more more reasoning

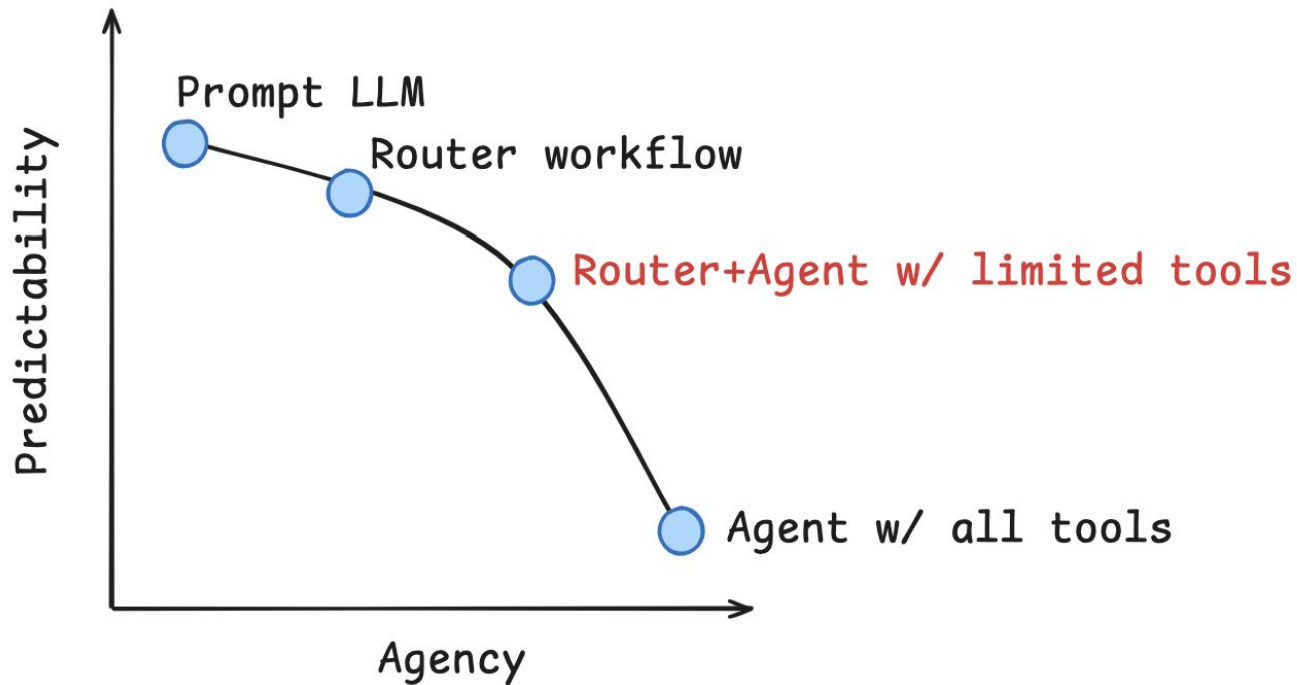
**Heuristic:** If you can easily *draw the control flow on a whiteboard*, a workflow is probably suitable for your application.

# LangGraph (1) lets us flexibly build / combine workflows and agents





LangGraph lets you build all of these!



# LangGraph (2) has a persistence layer for human in the loop + memory



Control flow of nodes, edges

Super-steps



Each sequential node is a separate super-step, while parallel nodes share the same super-step

Checkpoints



State and relevant metadata packaged at every super-step

Thread



Collection of checkpoints

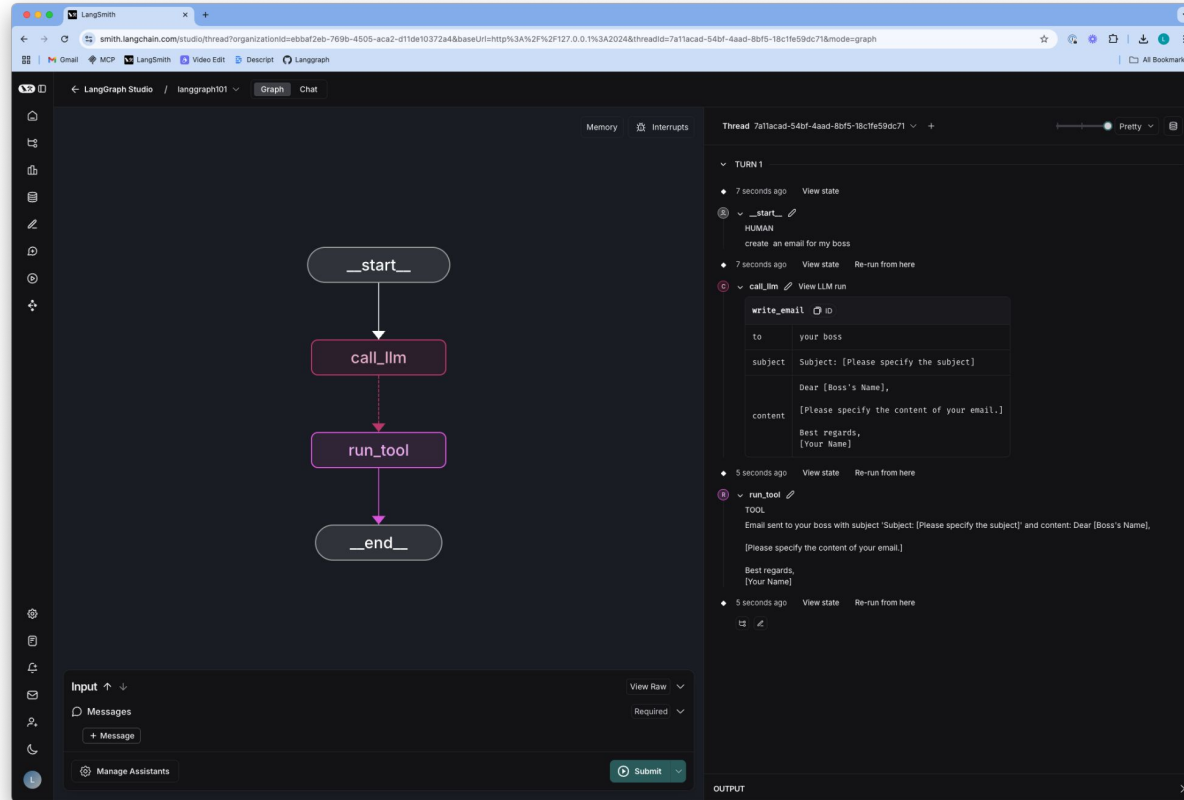
StateSnapshot

StateSnapshot()

StateSnapshot()

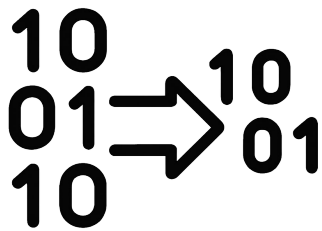
Type for checkpoints

# LangGraph (3) has an interactive agent development environment

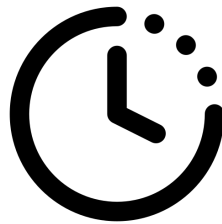


[https://langchain-ai.github.io/langgraph/concepts/langgraph\\_studio/](https://langchain-ai.github.io/langgraph/concepts/langgraph_studio/)

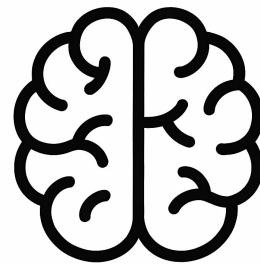
## LangGraph (4) has an easy onramp to deployment with LangGraph Platform



Streaming



Long-Running  
Background



Memory +  
Human in the loop