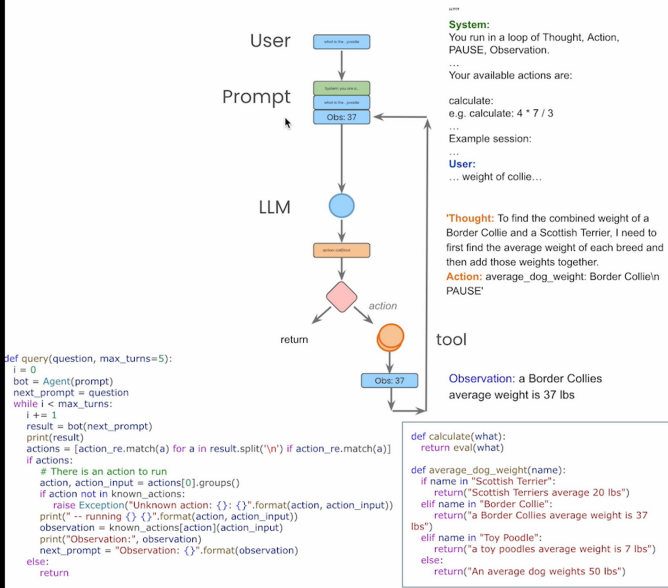


# Langgraph components

## Break Down



First, we give a **system message** to the **LLM** so it knows which tools can use, as well as the 3 cycles it can have (Thought, Action...etc)

**Thought:** I need to find the average weight of a Border Collie and a Scottish Terrier, then add them together to get the combined weight.  
**Action:** average\_dog\_weight: Border Collie  
**PAUSE**  
 -- running average\_dog\_weight Border Collie  
**Observation:** a Border Collies average weight is 37 lbs  
**Action:** average\_dog\_weight: Scottish Terrier  
**PAUSE**  
 -- running average\_dog\_weight Scottish Terrier  
**Observation:** Scottish Terriers average 20 lbs  
**Action:** calculate: 37 + 20  
**PAUSE**  
 -- running calculate 37 + 20  
**Observation:** 57  
**Answer:** The combined average weight of a Border Collie and a Scottish Terrier is 57 lbs.

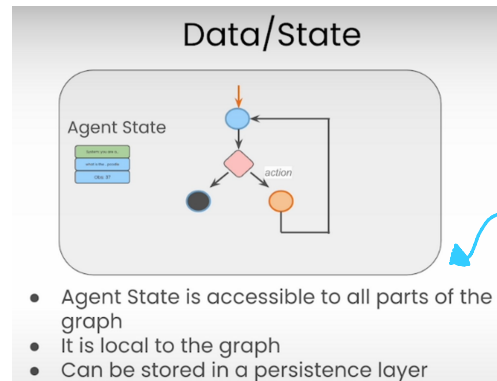
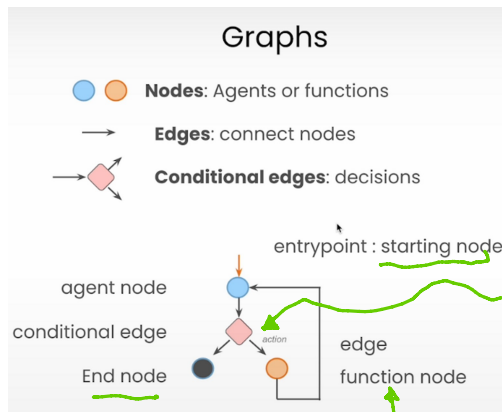
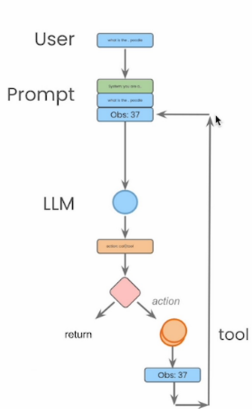
**Thought:** Now, the LLM broke down into Small parts in order to do the task

**Action:** The LLM starts calling the first tool it will need to use

After it gets a result from the tool, it will get An **OBSERVATION**, which the LLM will interpret and it will move onto the next step in the task

Once that all the tools and sub-tasks were done, the LLM will interpret the end result, and will deliver a plain-english **ANSWER** to the user

## Coding in langgraph



- Agent State is accessible to all parts of the graph
- It is local to the graph
- Can be stored in a persistence layer

The **AgentState** let us access all the components of the agent at any specific moment

Takes the result of the LLM call And decides what to do next

This calls a tool, and the retrieves the result To the LLM