



MultAgent

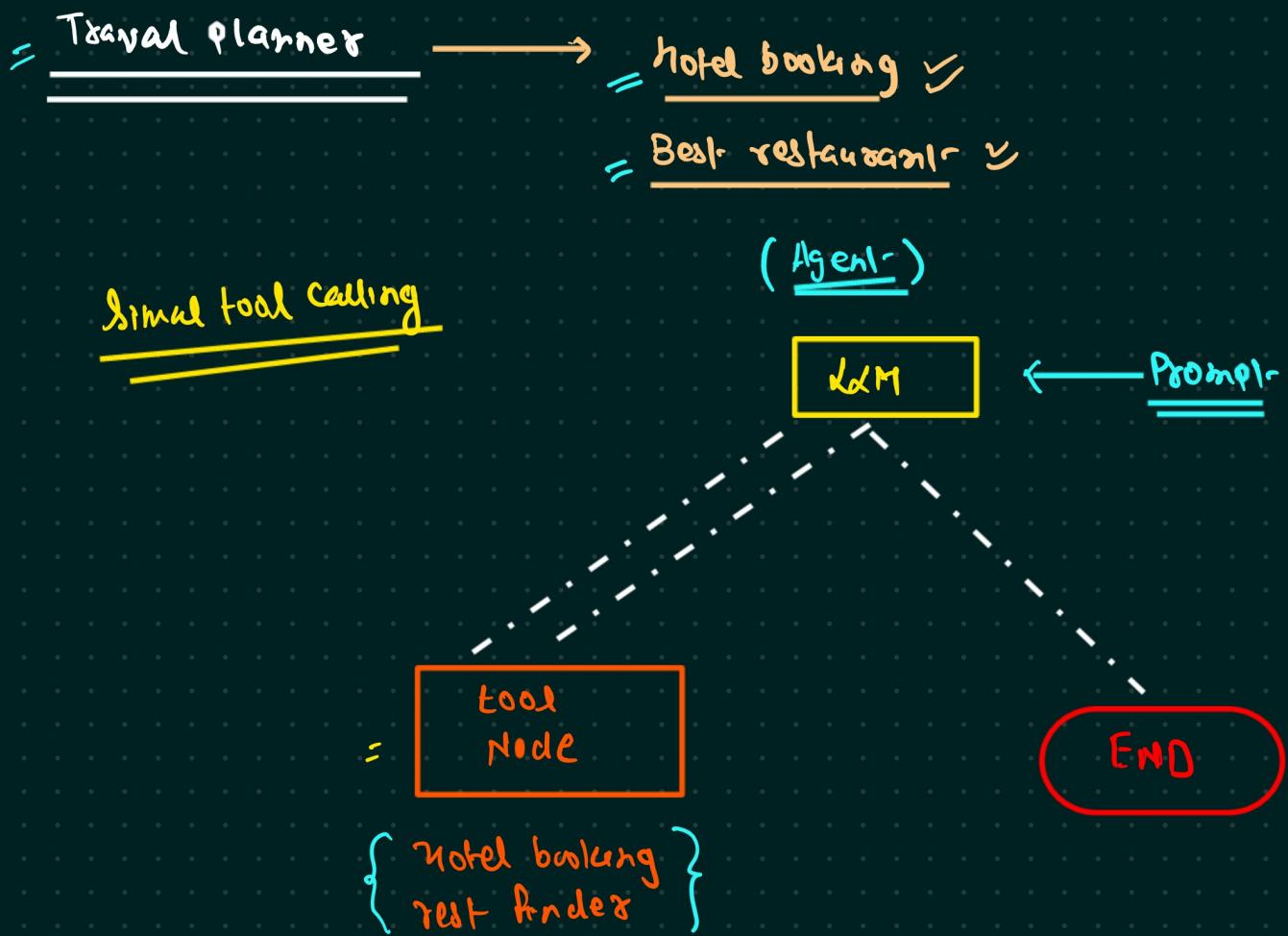
Multi + Agent

many

→ More than one Agent

→ Multiple Agent is talking to each other

- {
 - ① Network
 - ② Supervisor
- Hierarchical pattern

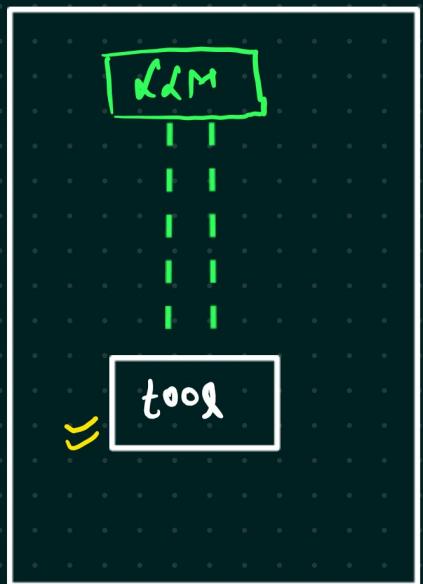


MultiAgent \Rightarrow Travel Planner

- ① Hotel booking
- ② restaurant finder

2 Agent \Rightarrow N no. of agent

↳ Hotel booking Agent



↳ Restaurant Finder Agent

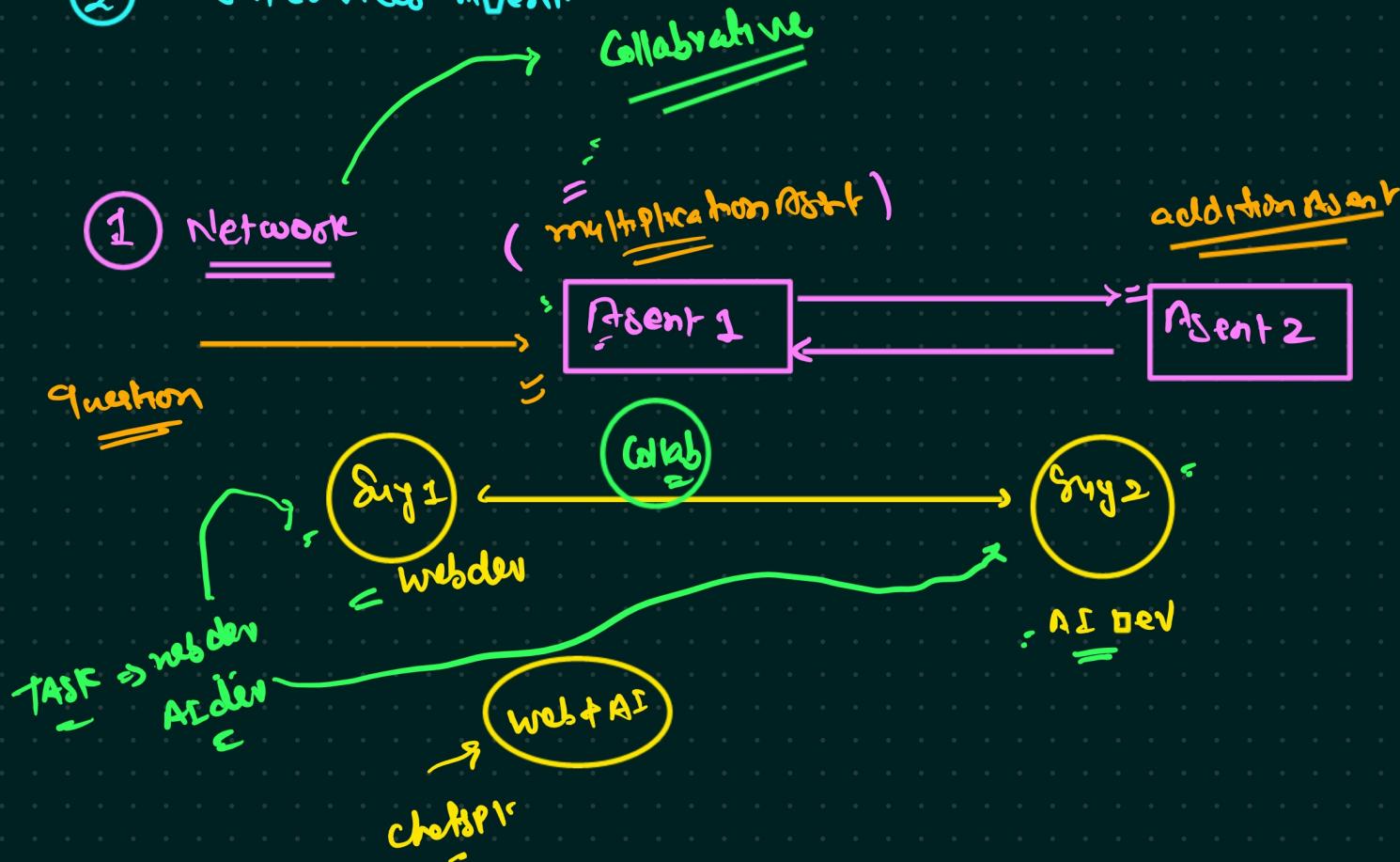


MultiAgent \Rightarrow (2 Agent)

- 1
- 2

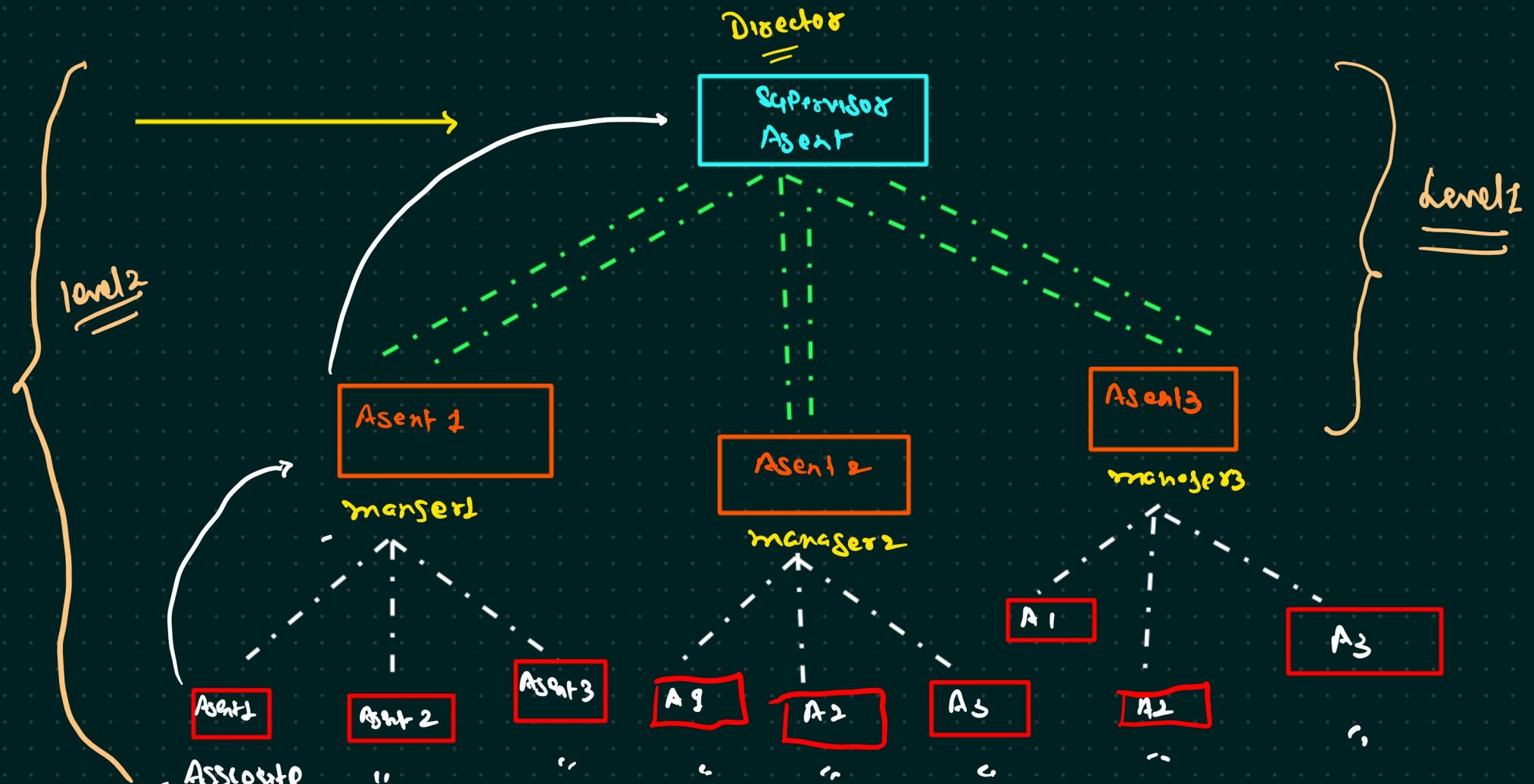
Network multiagent ✓

Supervisor agent ✓



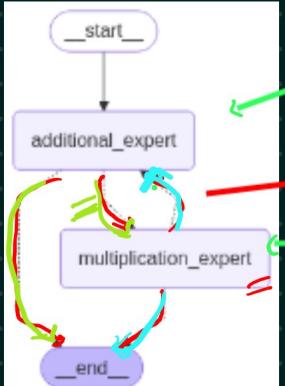
2

Supervisor multilevel.

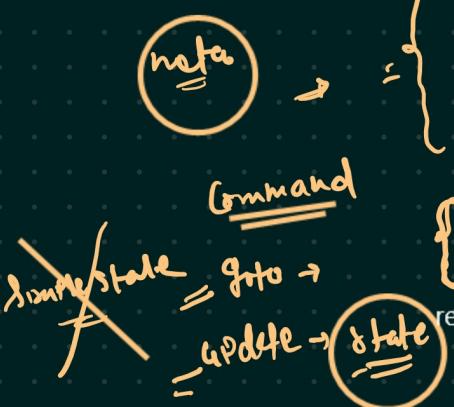


3

Hierarchical Agent → extension of supervisor Agent



= OIP

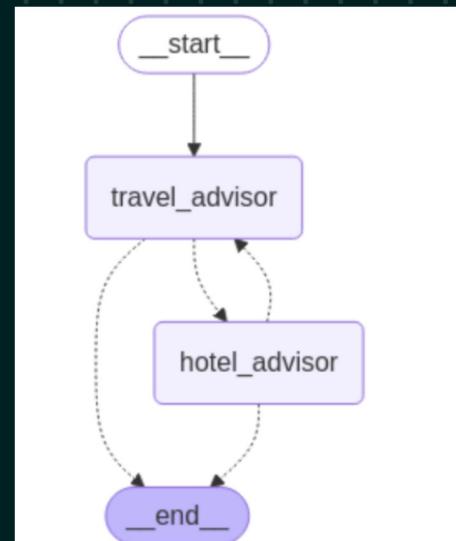
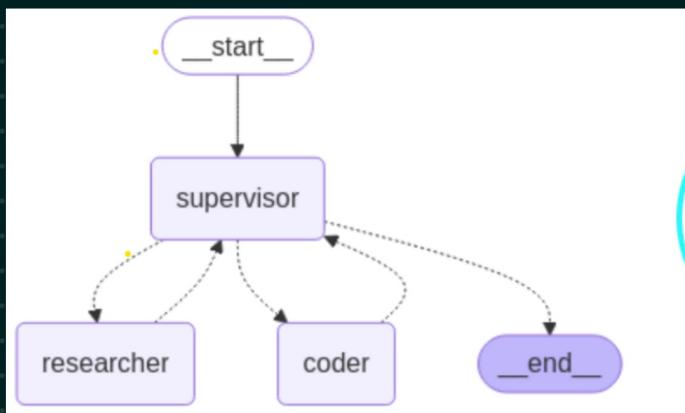


```

def additional_expert(state:MessagesState)-> Command[Literal["multiplication_expert", "__end__"]]:
    system_prompt = (
        "You are an addition expert, you can ask the multiplication expert for help with multiplication."
        "Always do your portion of calculation before the handoff." ← transfer
    )
    messages = [{"role": "system", "content": system_prompt}] + state["messages"]
    ai_msg = llm.bind_tools([transfer_to_multiplication_expert]).invoke(messages)

    if len(ai_msg.tool_calls) > 0:
        tool_call_id = ai_msg.tool_calls[-1]["id"]
        tool_msg = {
            "role": "tool",
            "content": "Successfully transferred",
            "tool_call_id": tool_call_id,
        }
        return Command(
            goto="multiplication_expert", update={"messages": [ai_msg, tool_msg]}
        )
    return {"messages": [ai_msg]}

```



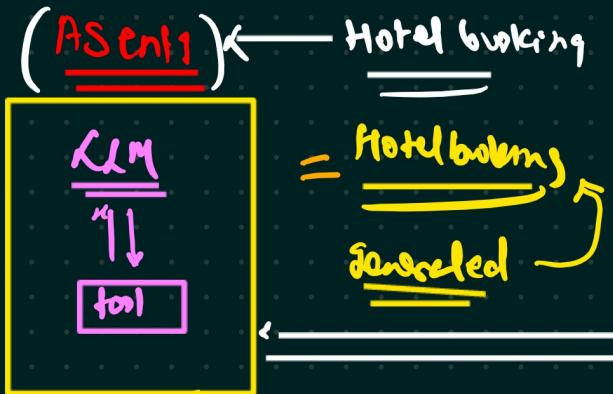
Revision of Network Agent

Problem \Rightarrow AI travel planner

Network | Collaborative Agent \leftarrow multiple Agent. Connected to each other

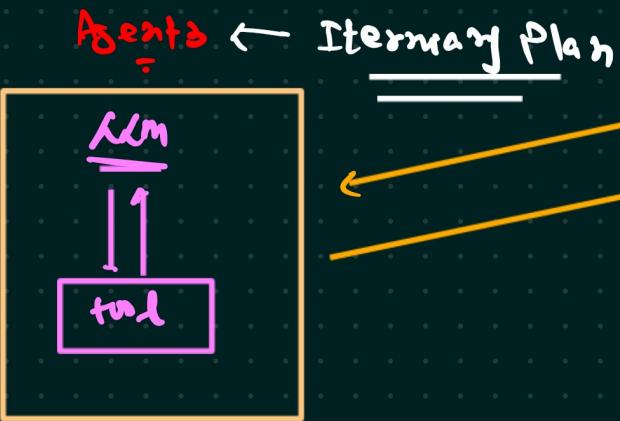
Question

I want to travel USA
For BM
can you book
a cheaper flight
With 3 star hotel
and save me 3 day -
Itinerary Plan -



(Flight booking) \rightarrow Agent 2

Flight booking =



```

##Agent2
def multiplication_expert(state:MessagesState) -> Command[Literal["additional_expert", "end"]]:
    system_prompt = (
        "You are a multiplication expert, you can ask an addition expert for help with addition."
        "Always do your portion of calculation before the handoff."
    )
    messages = [{"role": "system", "content": system_prompt}] + state["messages"]
    ai_msg = llm.bind_tools([transfer_to_addition_expert]).invoke(messages)
    if len(ai_msg.tool_calls) > 0:
        tool_call_id = ai_msg.tool_calls[-1]["id"]
        tool_msg = {
            "role": "tool",
            "content": "Successfully transferred",
            "tool_call_id": tool_call_id,
        }
    return Command(goto="additional_expert", update={"messages": [ai_msg, tool_msg]})
    return {"messages": [ai_msg]}

```

{ addition_tool → description }
 { multiply_tool → description }

real time tool → calculation

meta data