Lesson 2: Multi-Agents

by Hand ✍️

Prof. Tom Yeh

University of Colorado Boulder

Hosted by

GENAI.WORKS

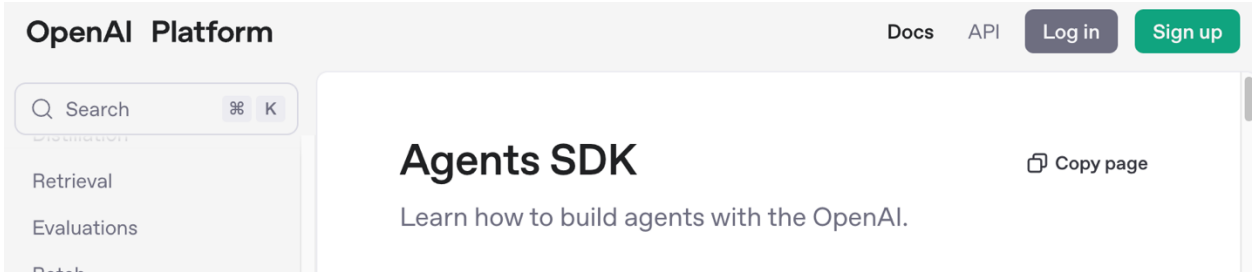
New Agent API

Lesson 2: Multi-Agents - by Hand ✍️

You think, saw, remembered, can

Creating or Cloning Agents

```python
from agents import Agent

# Basic
my_agent = Agent(
    name="my_agent",
    instructions="You are extremely friendly and helpful."
)

# Cloning with modifications
new_agent = my_agent.clone(
    name="my_new_agent",
    instructions="Now you're more formal."
)
```



https://platform.openai.com/docs/guides/agents-sdk#tools

Context

Sometimes you need to keep session data, credentials, or ephemeral state across multiple tool calls or steps. The Agents SDK allows a **context** object to be passed into the run.

- You create any Python class as context, or use a typed approach with `Agent(context_type=MyContext)`.
- Tools can then access that context, read data, and update it.

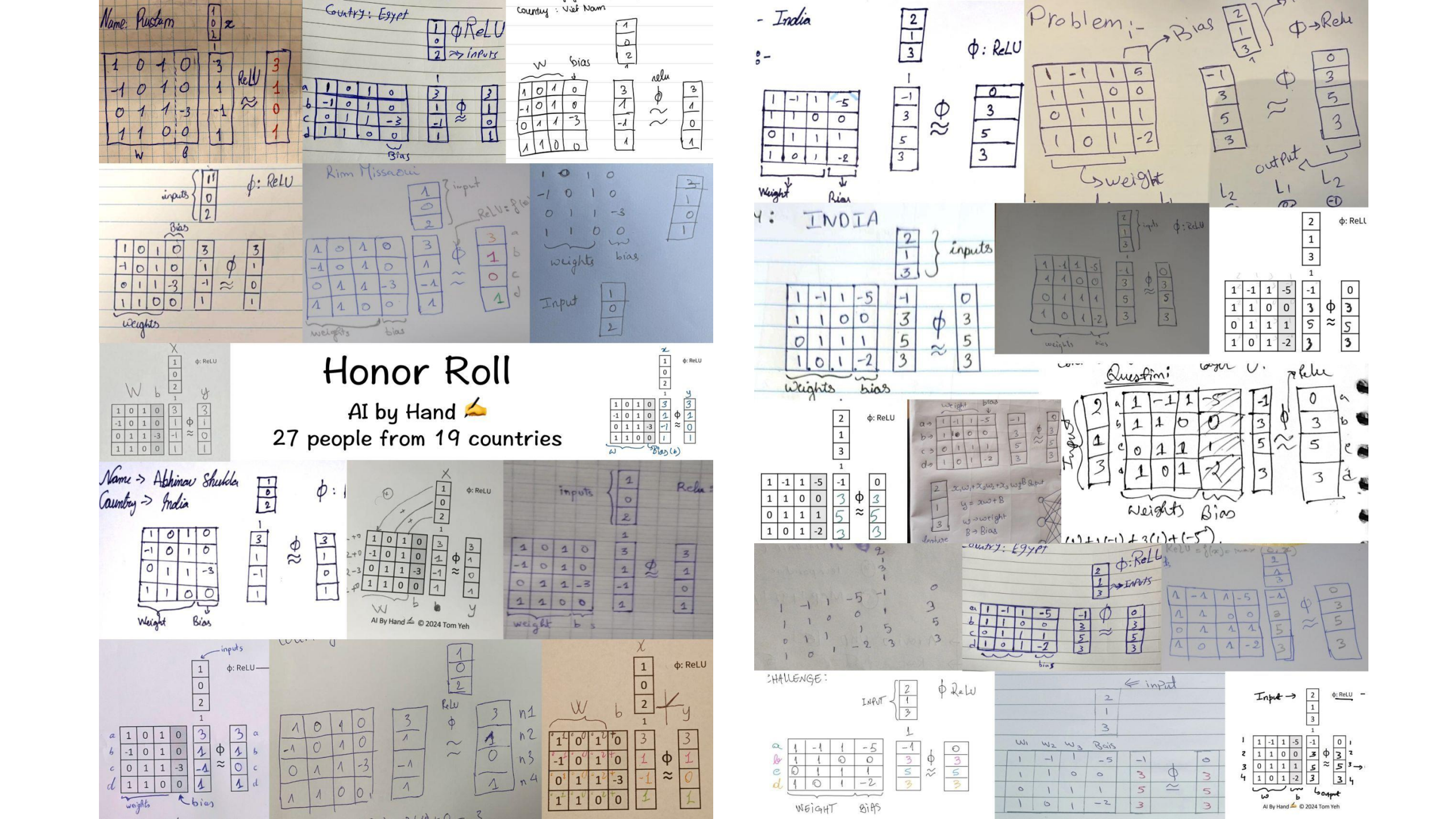
Using Context for User Sessions

```python
from agents.run_context import AgentContextWrapper
from agents.tool import function_tool

class MyContext:
  def __init__(self, user_id: str):
      self.user_id = user_id
      self.seen_messages = []

@function_tool
def greet_user(context: AgentContextWrapper[MyContext], greeting: str) -> str:
  user_id = context.agent_context.user_id
  return f"Hello {user_id}, you said: {greeting}"

agent = Agent(
  name="my_agent_with_context",
  context_type=MyContext,
  tools=[greet_user],
)

my_ctx = MyContext(user_id="alice")
result = await AgentRunner.run(
  agent,
  input=["Hi agent!"],
  context=my_ctx,
)
```

What is AI by Hand ✍️?

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder


Honor Roll
AI by Hand ✍️
27 people from 19 countries
Name: Rustam
Country: Egypt
Country : Viet Nam
- India
Problem :-
Bias
ReLU
Weight
Weights
bias
inputs
Rim Missaoui
Name -> Abhinav Shukla
Country -> India
Question:
Challenge:
Input
Output
AI By Hand ✍ © 2024 Tom Yeh

Neural Network

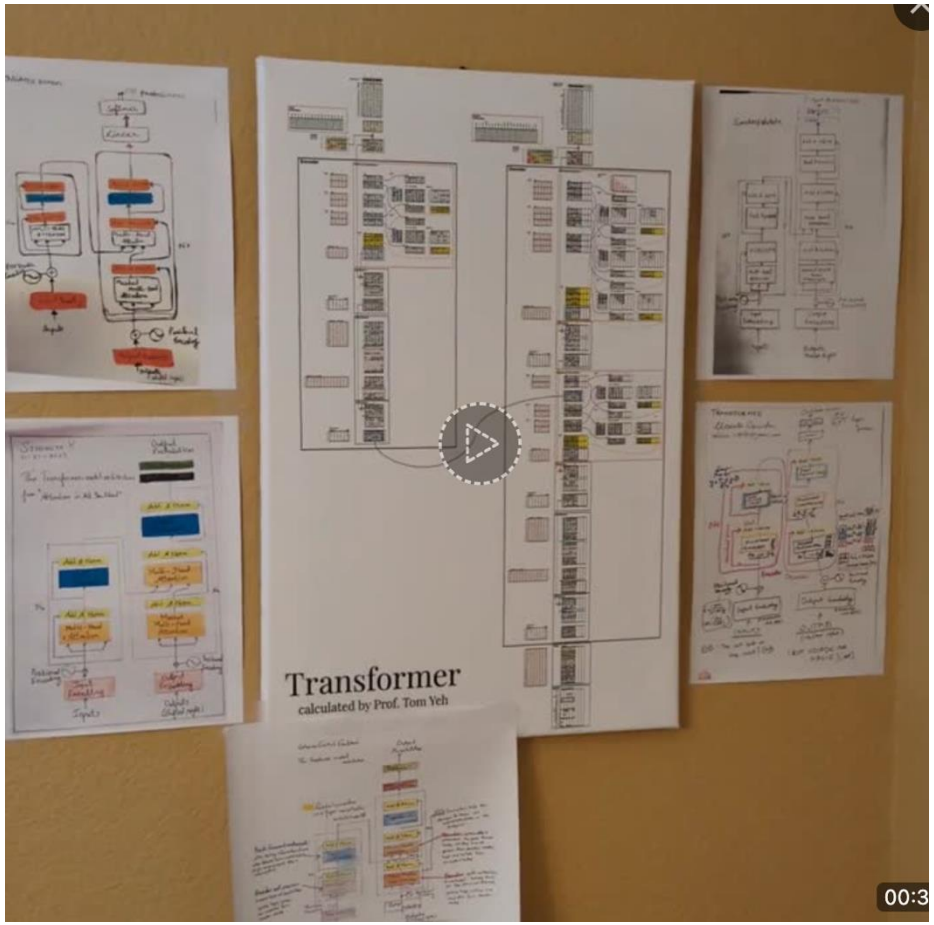
AI by Hand ✍️

Tom | Sudhin | Bhishan

Boulder USA | Pokhar Nepal | Kathmandu Nepal

Watch my lecture videos at

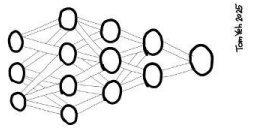
byhand.ai/youtube

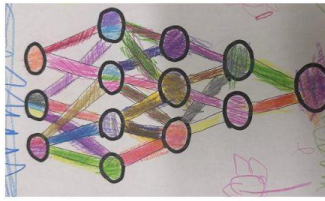


Color AI by Hand ✍️ Togeth

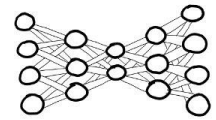
see my post to participate in th group art project

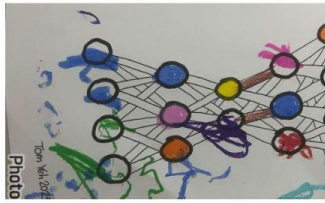
Neural Net



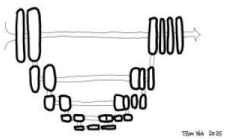


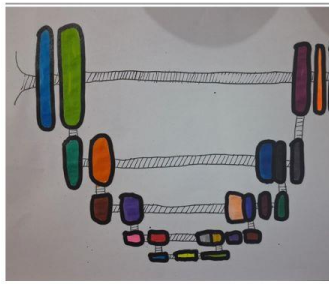
Auto-Encoder



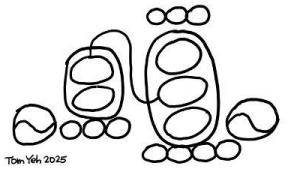


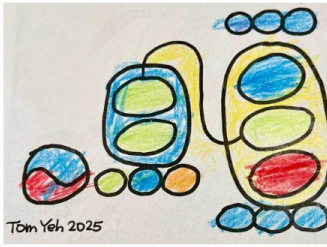
U-Net

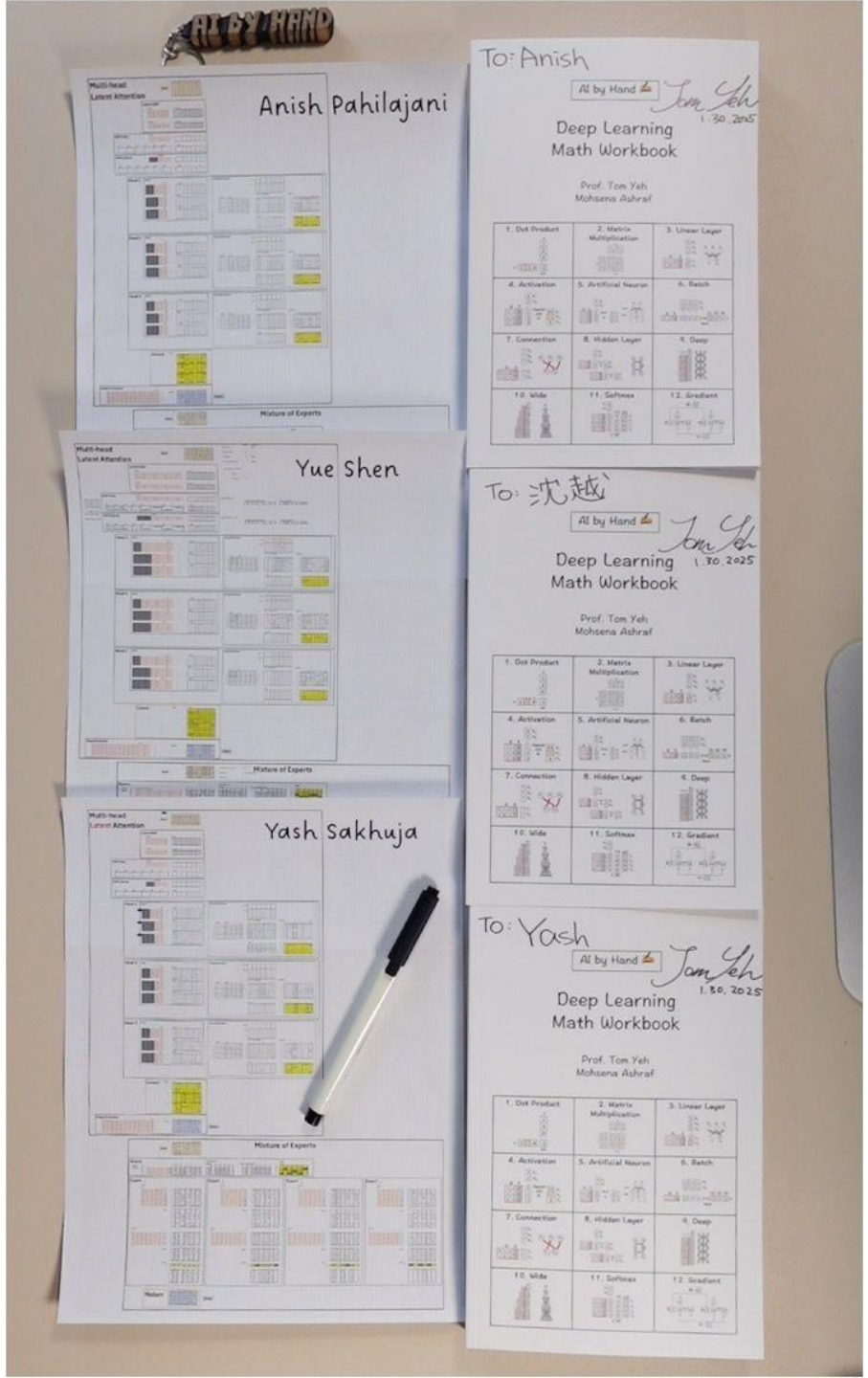




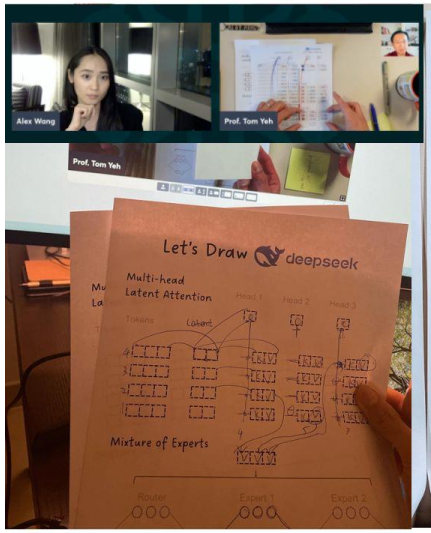
Transformer



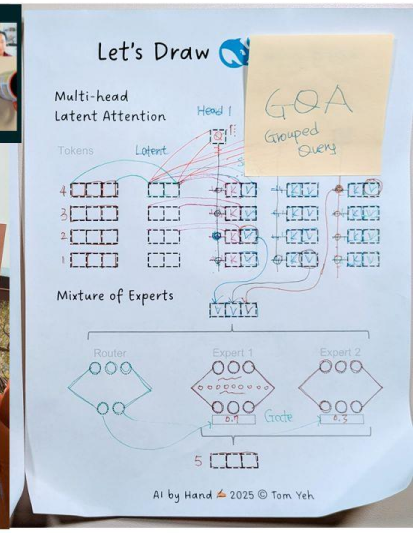




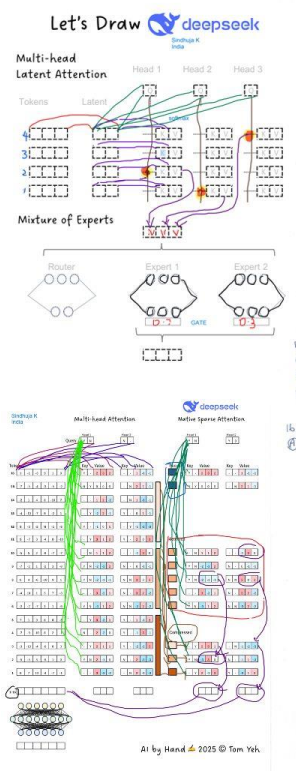
Alex Wang



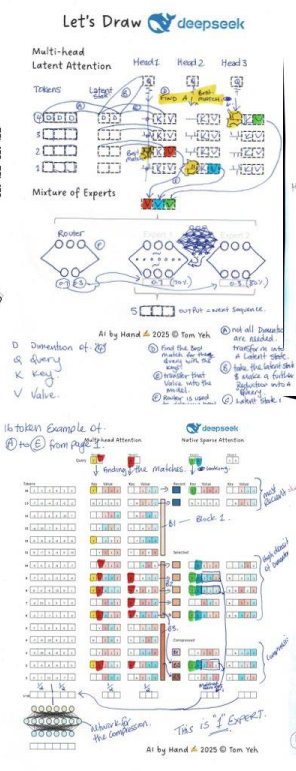
Tom Yeh



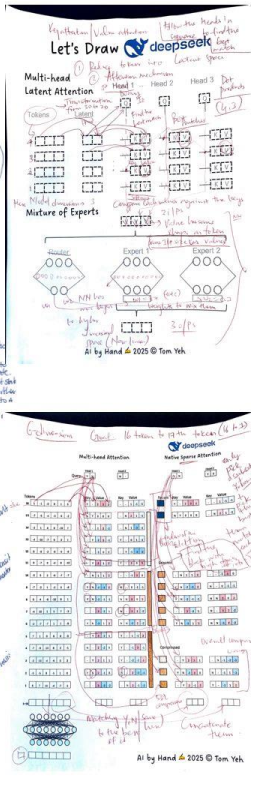
Sindhuja, India

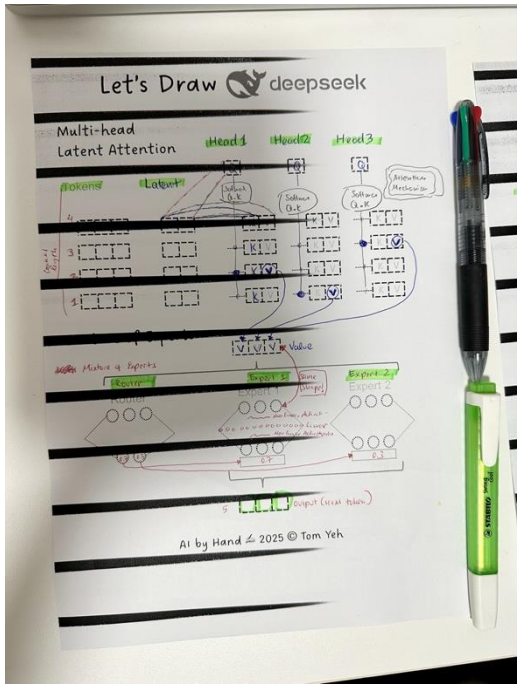


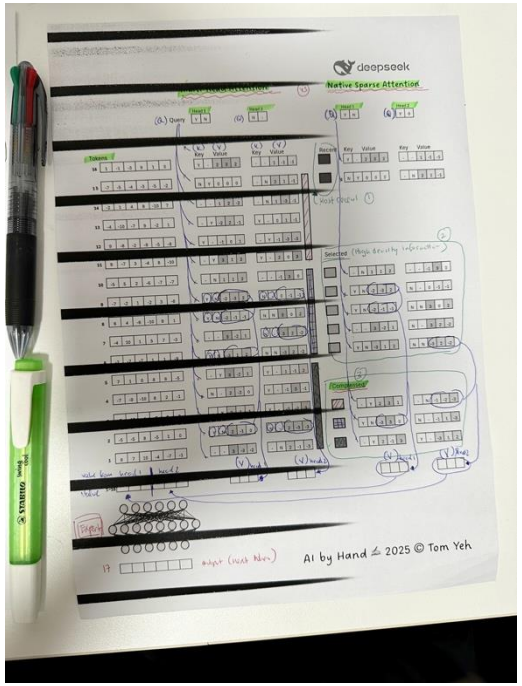
Anand, USA



Phil, Australia





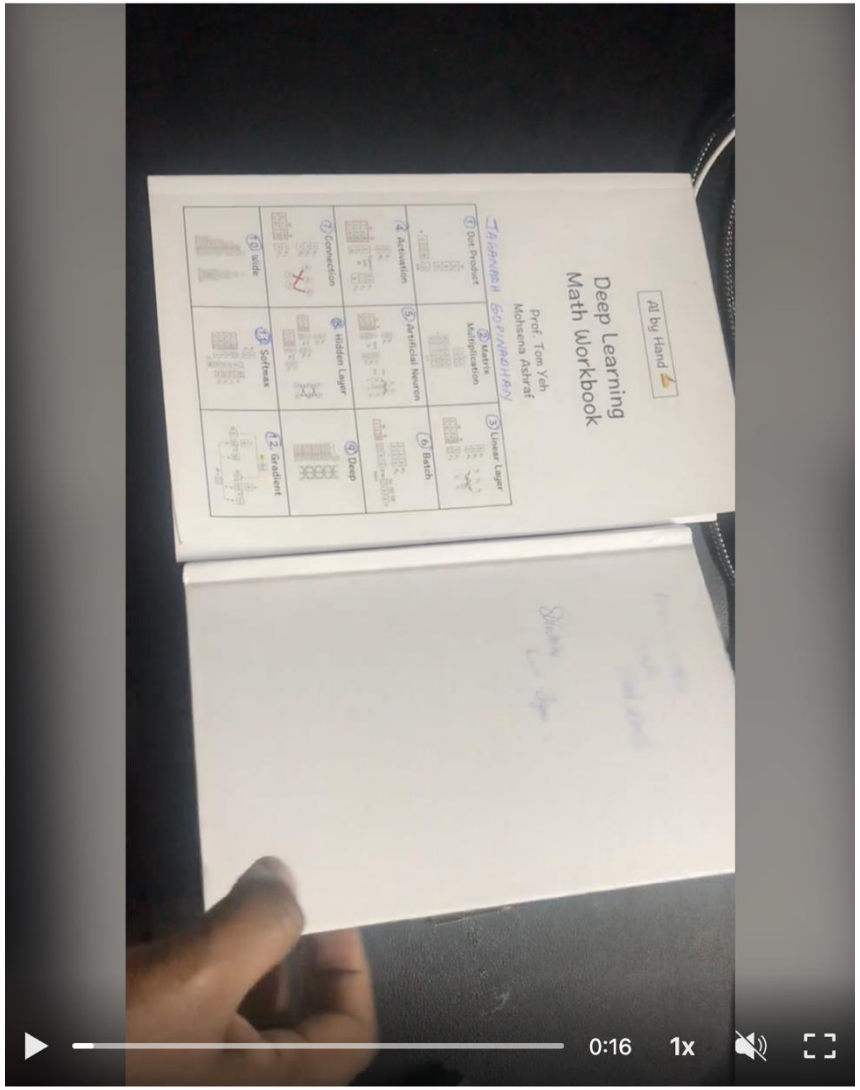


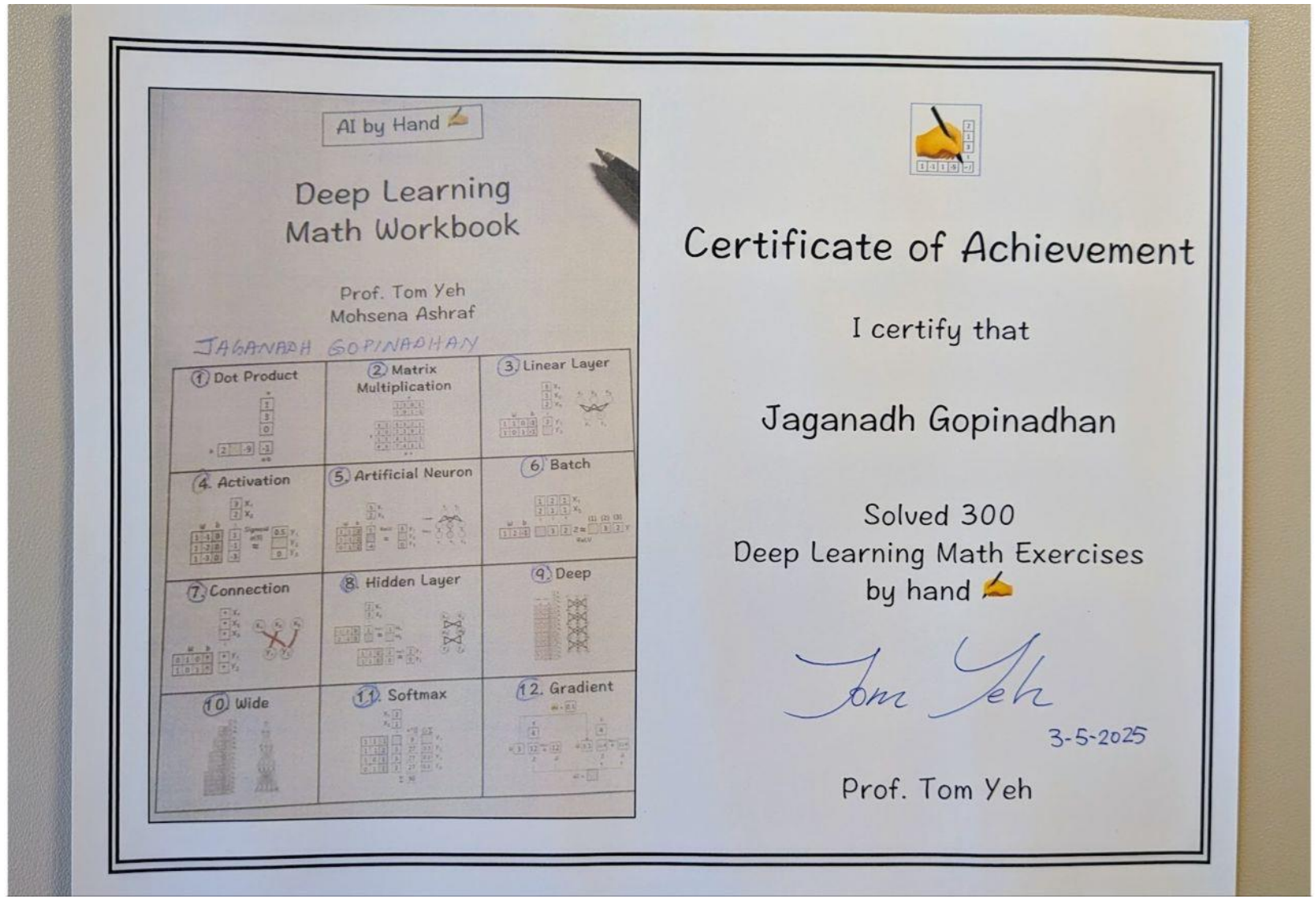
Jaganadh Gopinadhan (Jagan) · 2nd
Associate Director - Engineering @ Cognizant | Trusted AI ...
5d ·

+ Follow ···

Finally, I completed all 12 chapters from the AI by Hand Deep Learning Math Workbook. Solving each of the problems was a great experience. Thanks to Tom Yeh and Mohsena Ashraf for creating such excellent material.
I'm looking forward to other workbooks, too.

#aibyhand





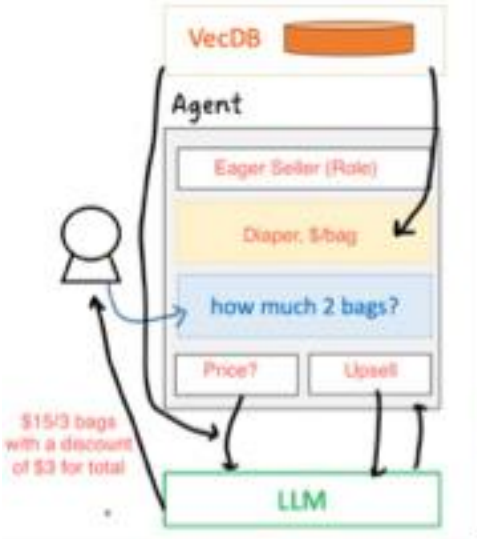
You and 62 others

3 comments · 3 reposts

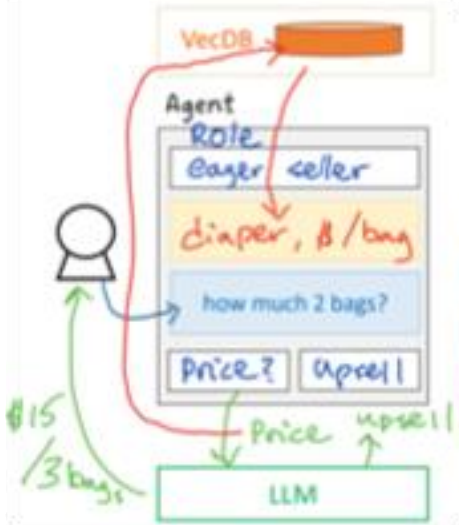
Assignment Submissions

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder

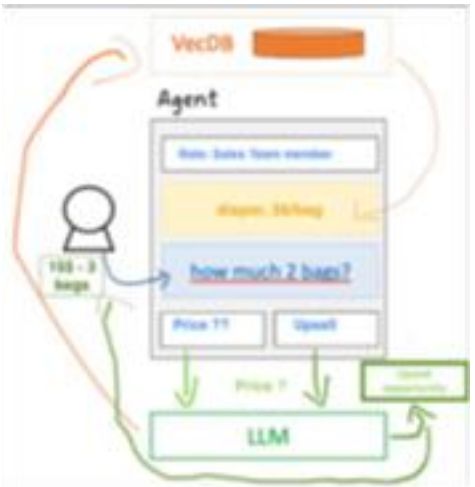


Samiul



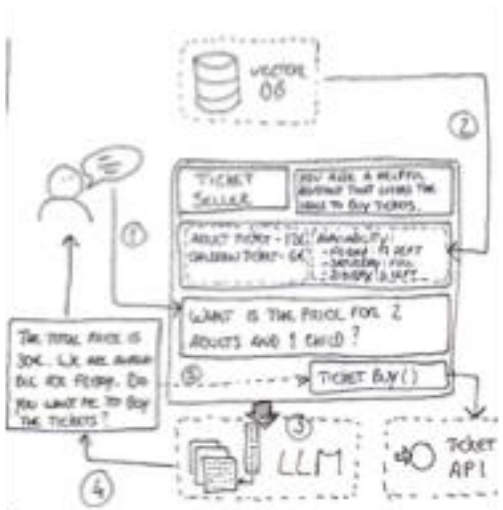
Vicent

Nishita

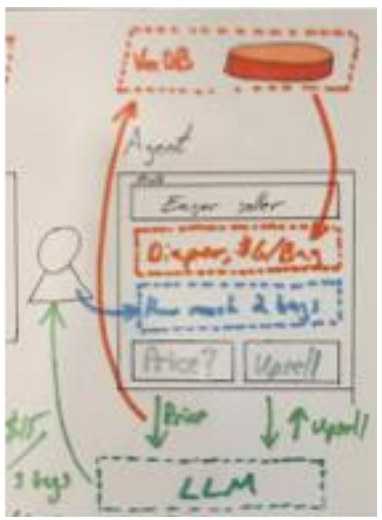


Georges

Kedar



Tiago



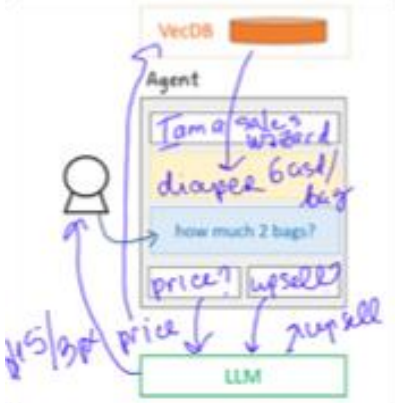
Mayer

Maria

Certificate

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder



Lesson 2

Lesson 3

Lesson 4

GENAI.WORKS

Certificate of Achievement

I certify that

Maria Petrenko

Completed

Introduction to Agentive AI

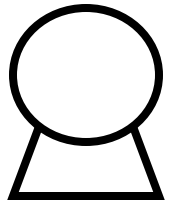
by hand ✍️

Prof. Tom Yeh

Multi-Agents

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder



Agent

helpful receptionist

→manager

who?

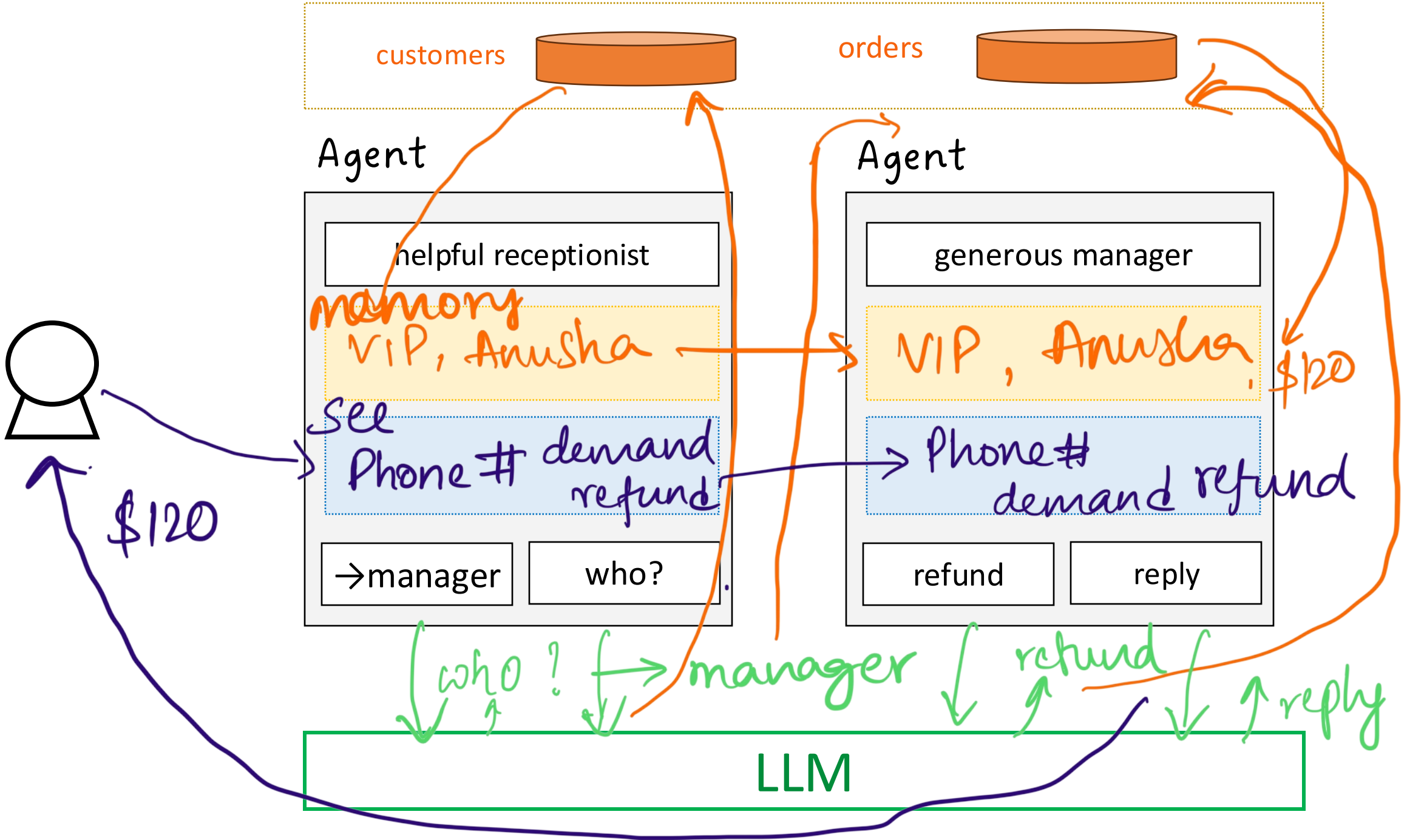
Agent

generous manager

refund

reply

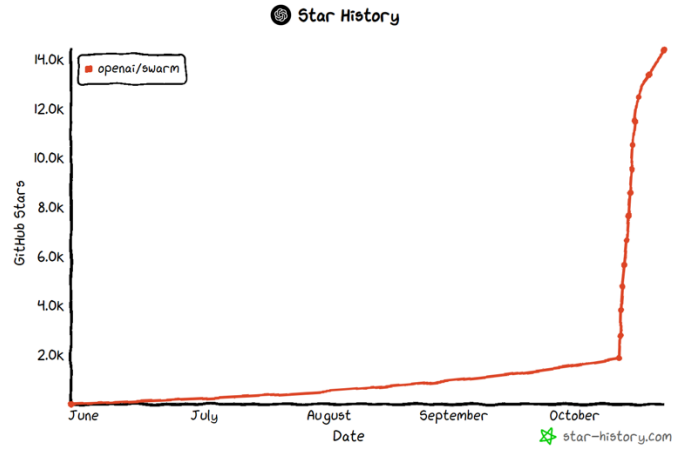
LLM


customers
orders
Agent
helpful receptionist
memory
VIP, Anusha
see
Phone # demand refund
→manager
who?
Agent
generous manager
VIP, Anusha, $120
Phone # demand refund
refund
reply
$120
who ?
→ manager
refund
reply
LLM

OpenAI Swarm

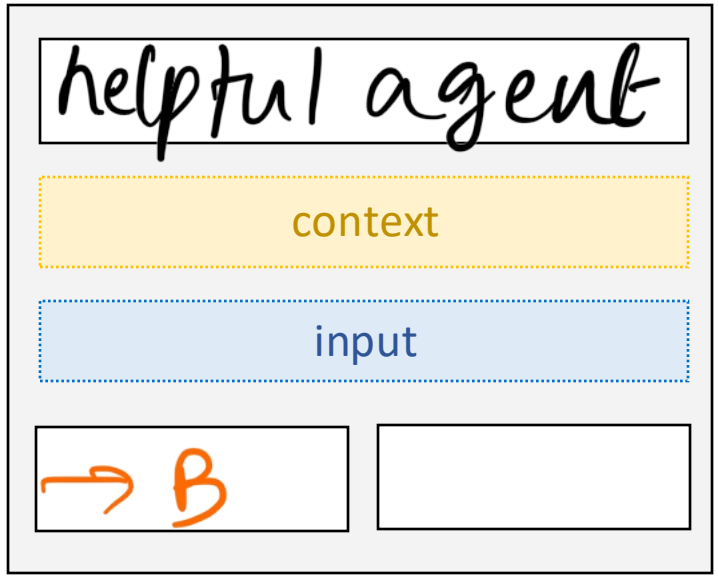
Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado Boulder

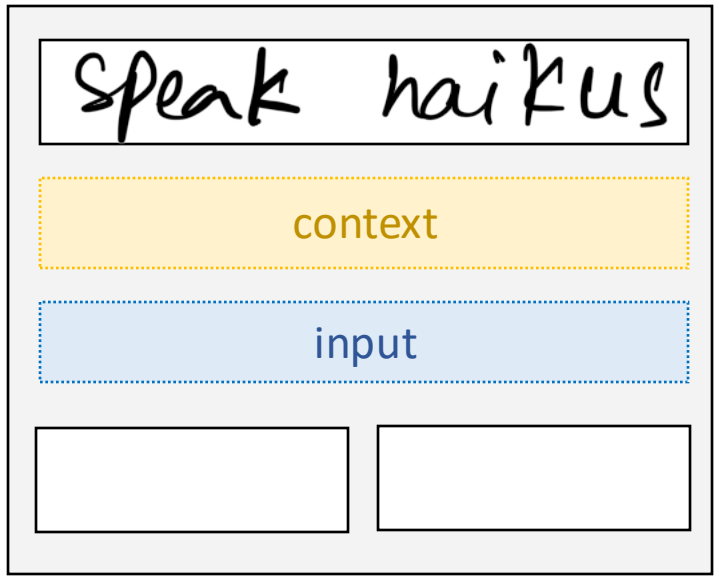


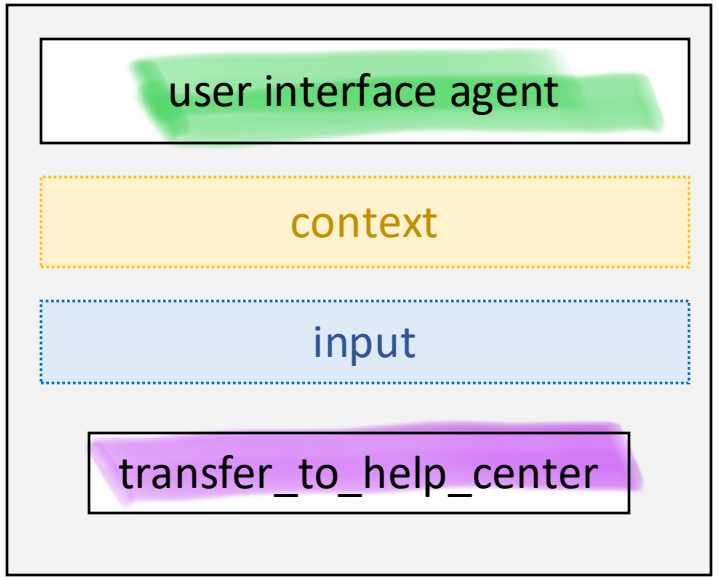
```python
from swarm import Swarm, Agent

client = Swarm()

def transfer_to_agent_b():
    return agent_b


agent_a = Agent(
    name="Agent A",
    instructions="You are a helpful agent.",
    functions=[transfer_to_agent_b],
)

agent_b = Agent(
    name="Agent B",
    instructions="Only speak in Haikus.",
)

response = client.run(
    agent=agent_a,
    messages=[{"role": "user", "content": "I want to talk to agent B."}],
)

print(response.messages[-1]["content"])
```

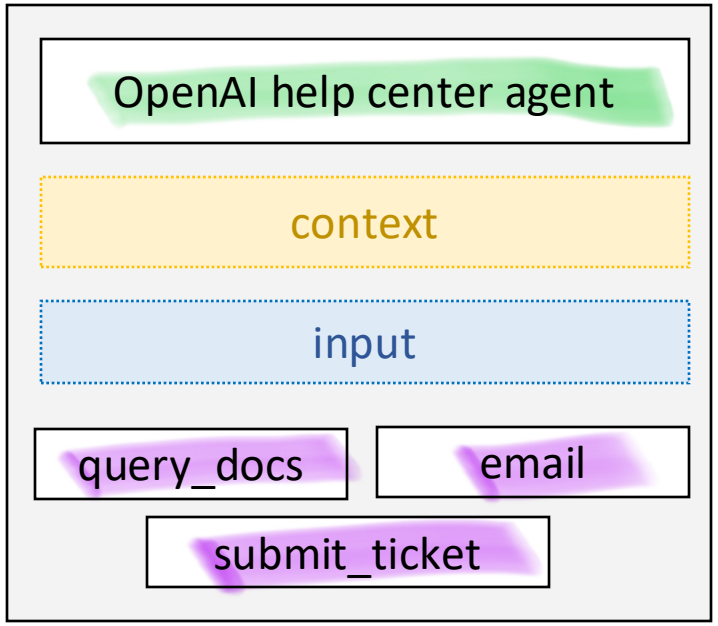
A



B

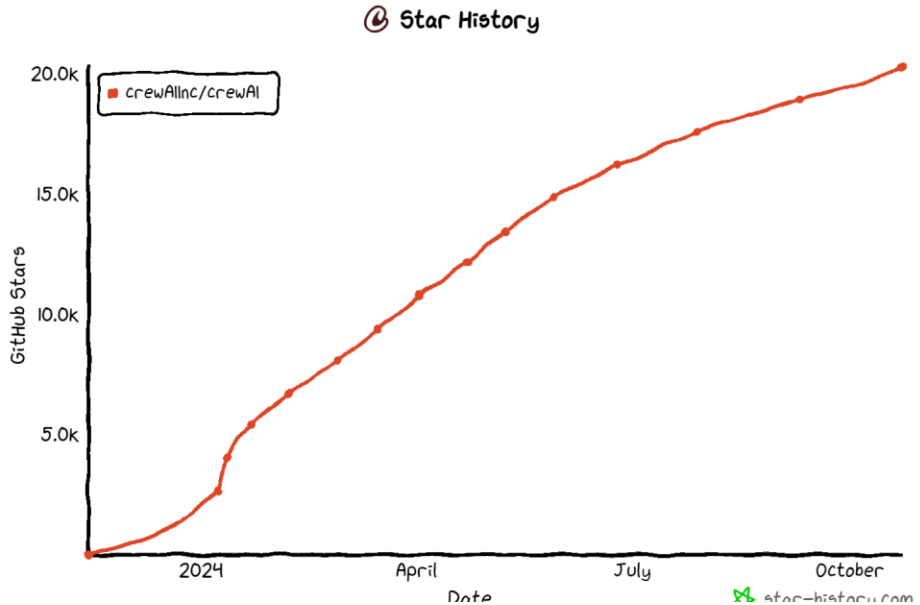






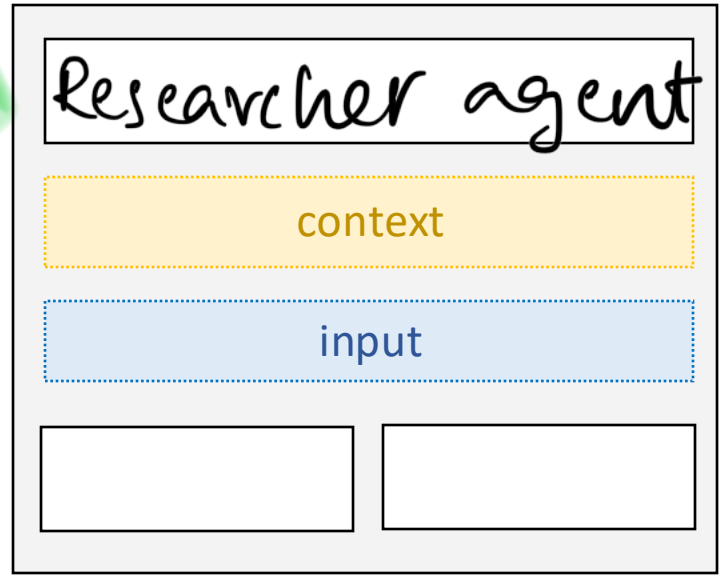

```python
user_interface_agent = Agent(
    name="User Interface Agent",
    instructions="You are a user interface agent that handles a
    functions=[transfer_to_help_center],
)

help_center_agent = Agent(
    name="Help Center Agent",
    instructions="You are an OpenAI help center agent who deals
    functions=[query_docs, submit_ticket, send_email],
)
```

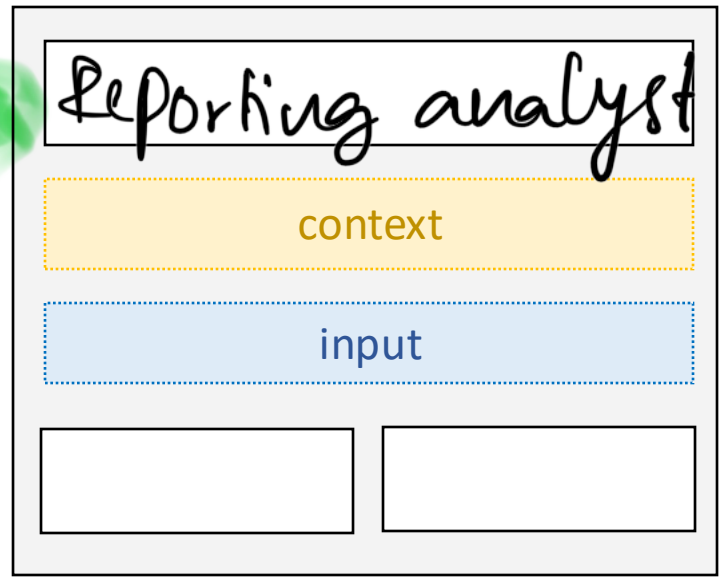
CrewAI



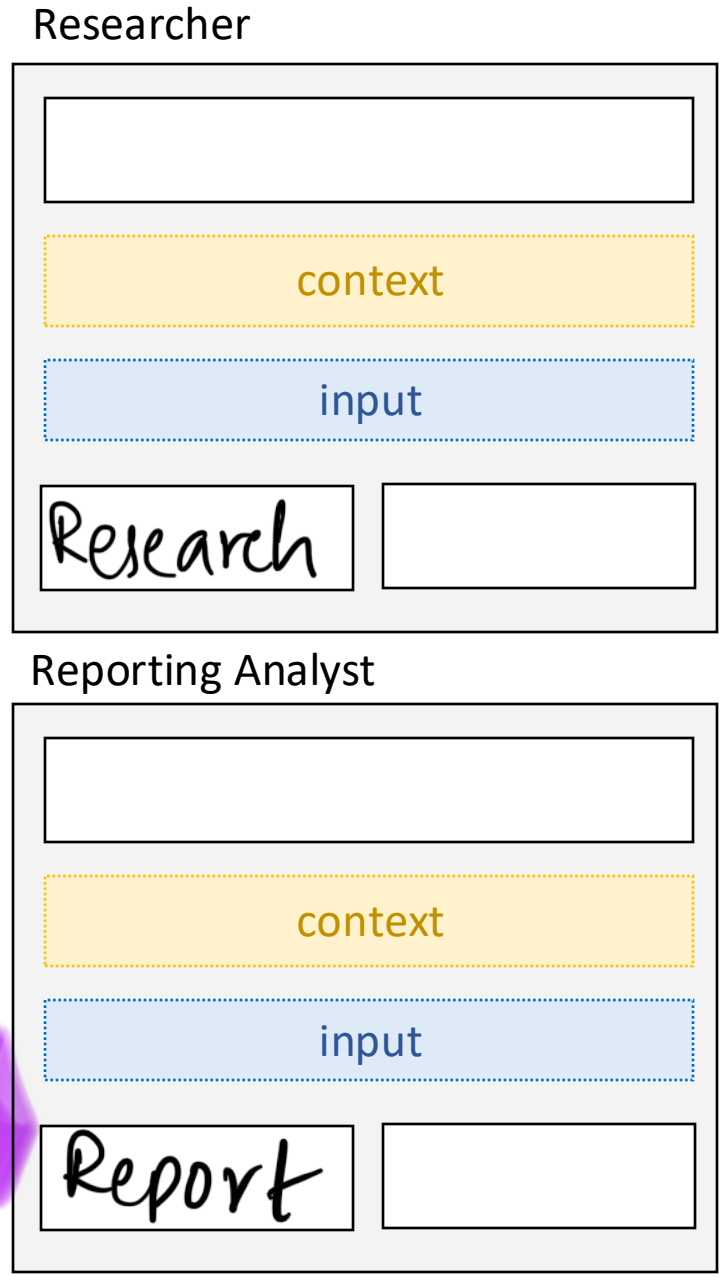
Lesson 2: Multi-Agents - by Hand ✍️

```yaml
# src/my_project/config/agents.yaml
researcher:
  role: >
    {topic} Senior Data Researcher
  goal: >
    Uncover cutting-edge developments in {topic}
  backstory: >
    You're a seasoned researcher with a knack for uncovering the lat
    developments in {topic}. Known for your ability to find the most
    information and present it in a clear and concise manner.

reporting_analyst:
  role: >
    {topic} Reporting Analyst
  goal: >
    Create detailed reports based on {topic} data analysis and resea
  backstory: >
    You're a meticulous analyst with a keen eye for detail. You're k
    your ability to turn complex data into clear and concise reports
    it easy for others to understand and act on the information you
```





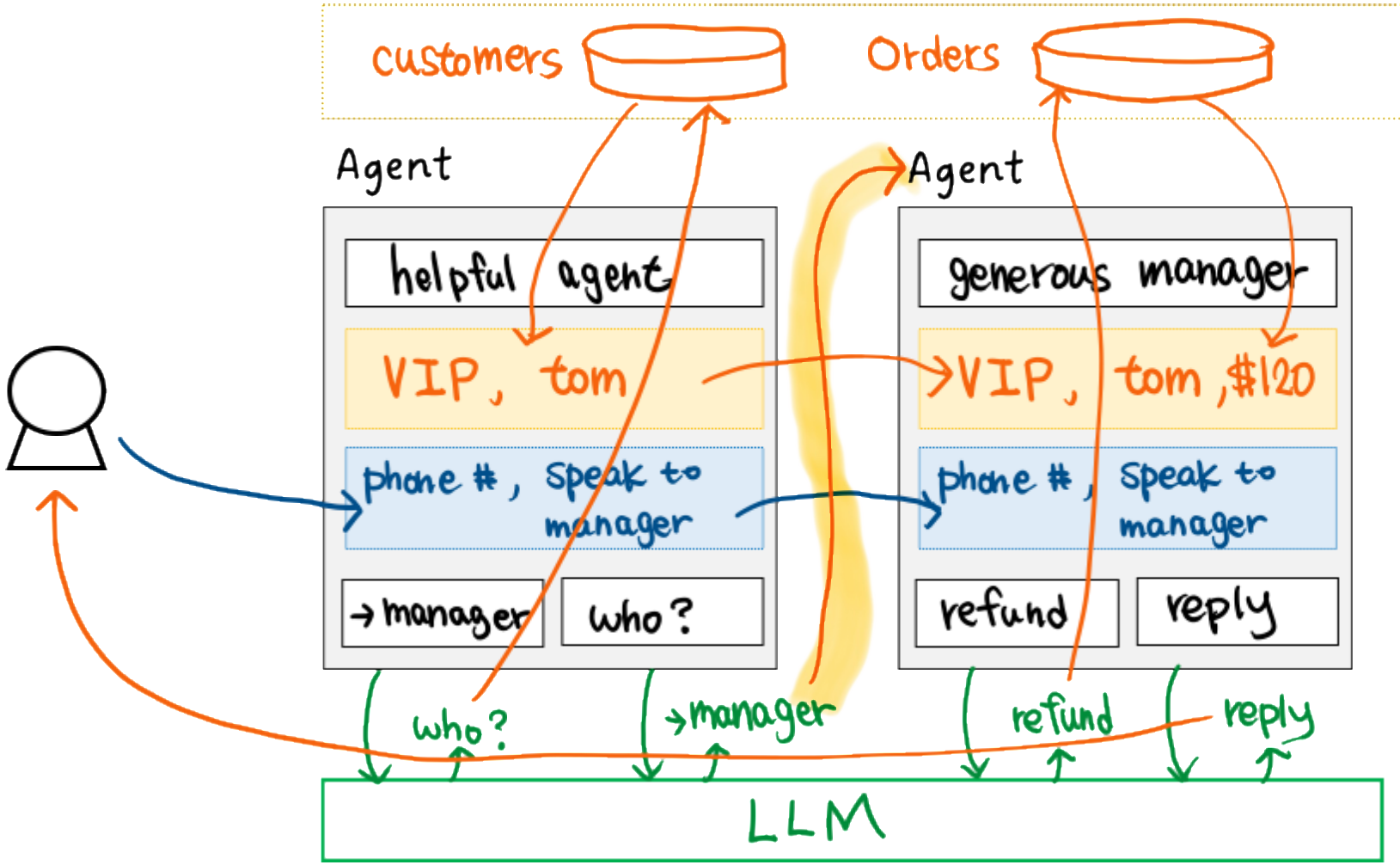
```yaml
# src/my_project/config/tasks.yaml
research_task:
  description: >
    Conduct a thorough research about {topic}
    Make sure you find any interesting and relevant information give
    the current year is 2024.
  expected_output: >
    A list with 10 bullet points of the most relevant information ab
  agent: researcher

reporting_task:
  description: >
    Review the context you got and expand each topic into a full sec
    Make sure the report is detailed and contains any and all releva
  expected_output: >
    A fully fledge reports with the mains topics, each with a full s
    Formatted as markdown without '```'
  agent: reporting_analyst
  output_file: report.md
```

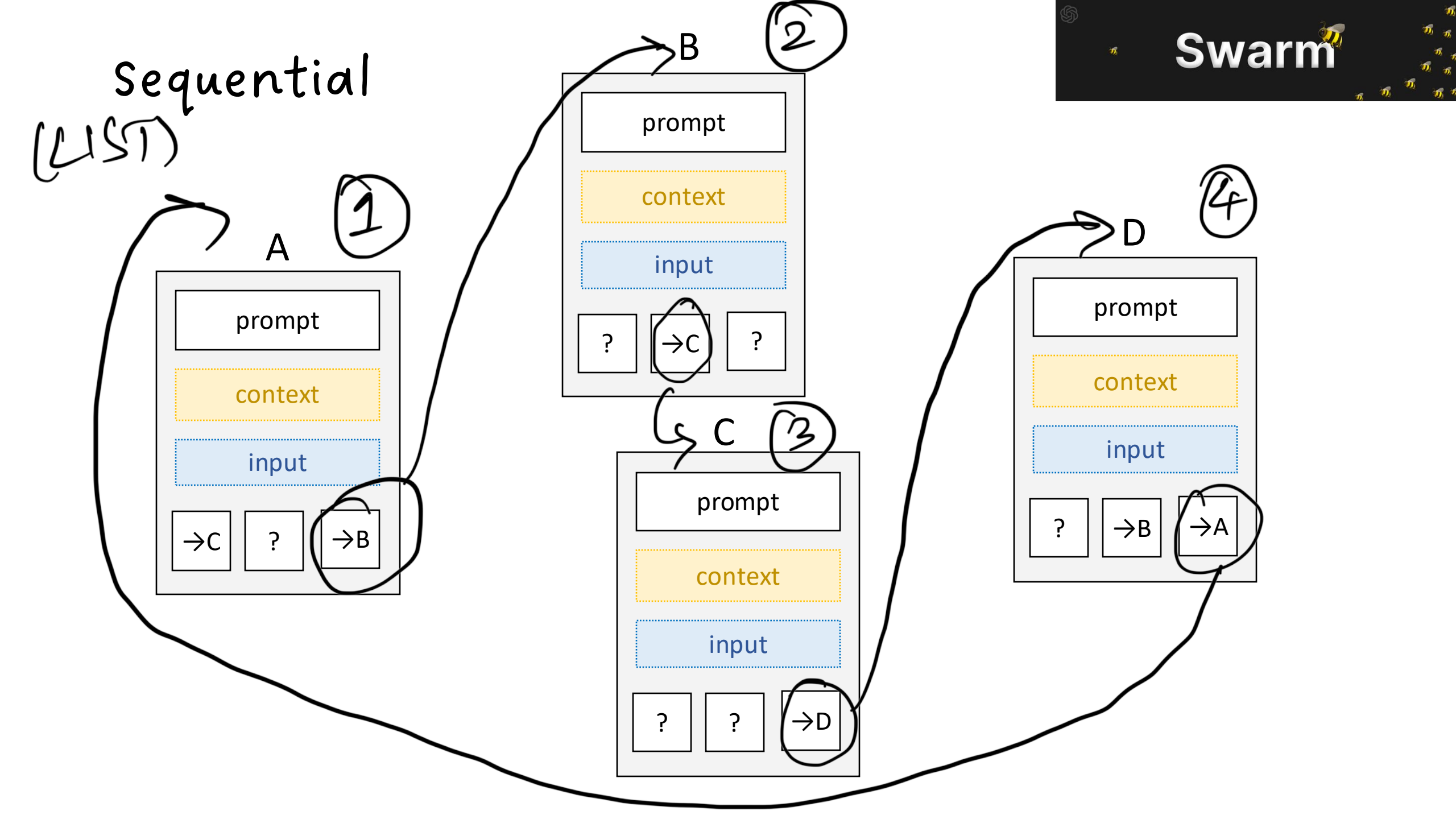


Process

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder

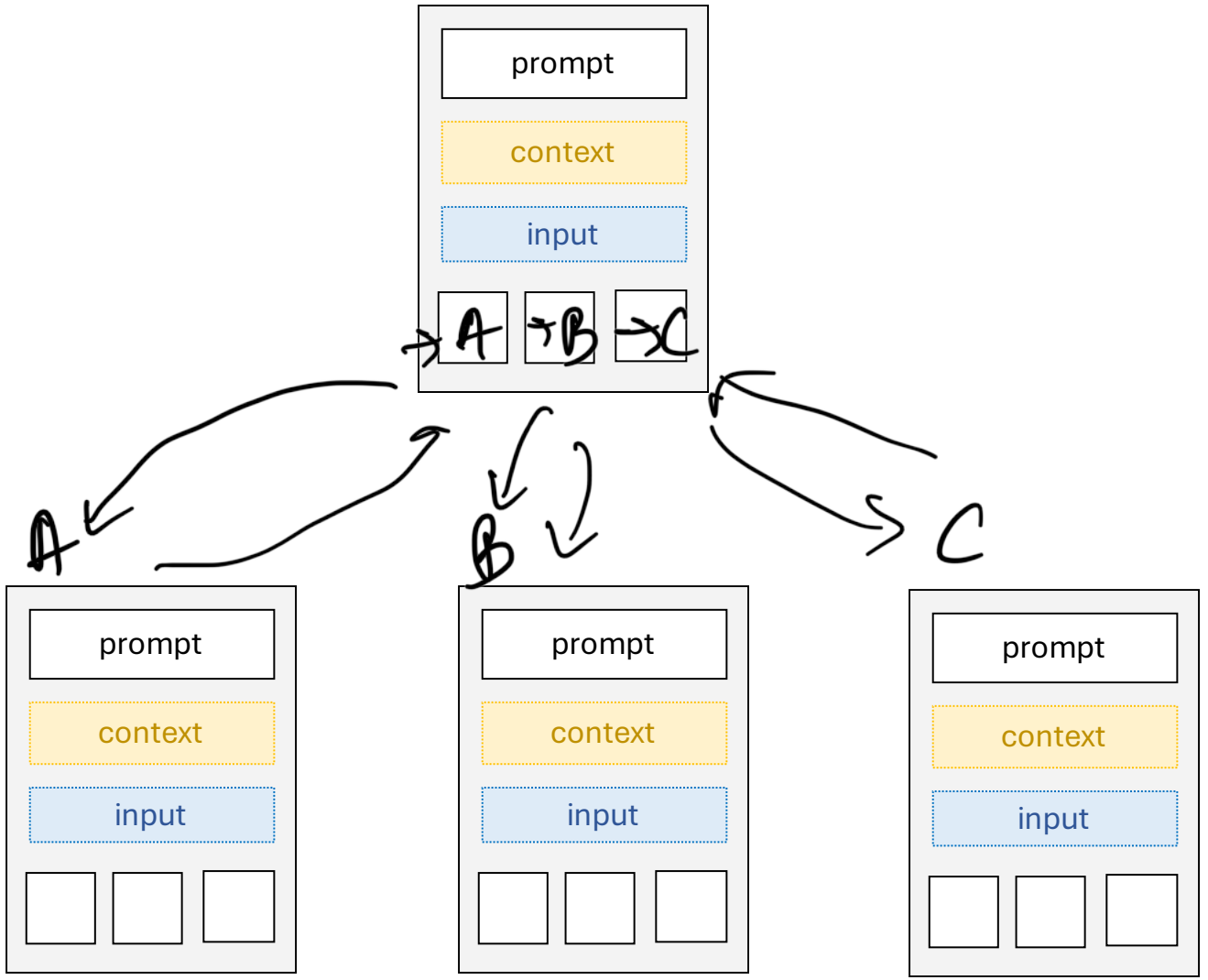

customers
Orders
Agent
Agent
helpful agent
generous manager
VIP, tom
VIP, tom, $120
phone #, Speak to manager
phone #, Speak to manager
→manager
who?
refund
reply
who?
→manager
refund
reply
LLM


Swarm
Sequential
(LIST)
A
1
prompt
context
input
→C
?
→B
B
2
prompt
context
input
?
→C
?
C
3
prompt
context
input
?
?
→D
D
4
prompt
context
input
?
→B
→A

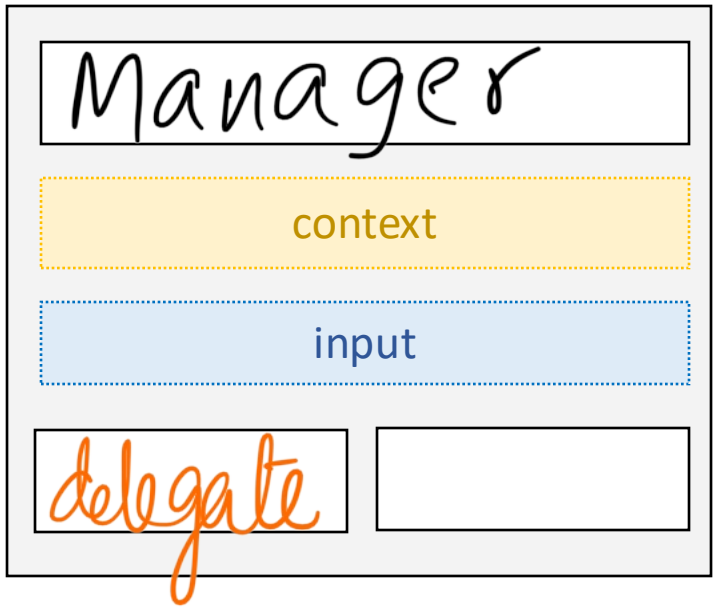
Hierarchical (GRAPH)

Manager

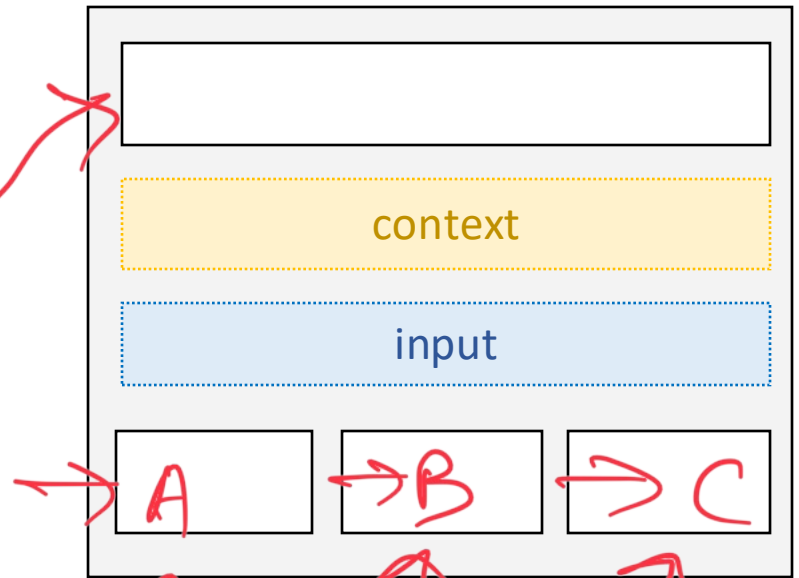
crewai



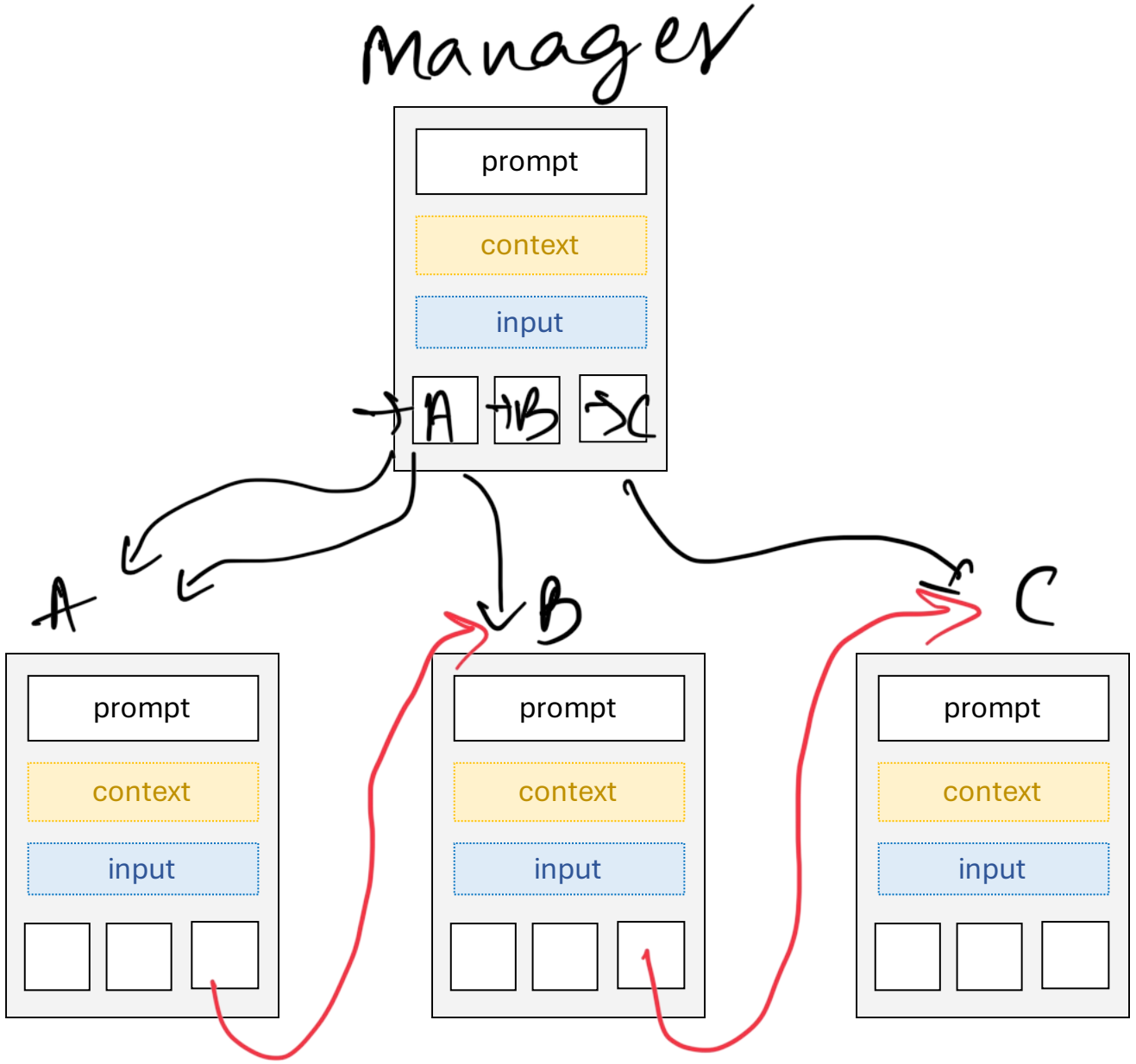
```
"hierarchical_manager_agent": {
    "role": "Crew Manager",
    "goal": "Manage the team to complete the task in the best
      way possible.",
    "backstory": "You are a seasoned manager with a knack for
      getting the best out of your team.\nYou are also known
      for your ability to delegate work to the right people,
      and to ask the right questions to get the best out of
      your team.\nEven though you don't perform tasks by
      yourself, you have a lot of experience in the field,
      which allows you to properly evaluate the work of your
      team members."
}
```



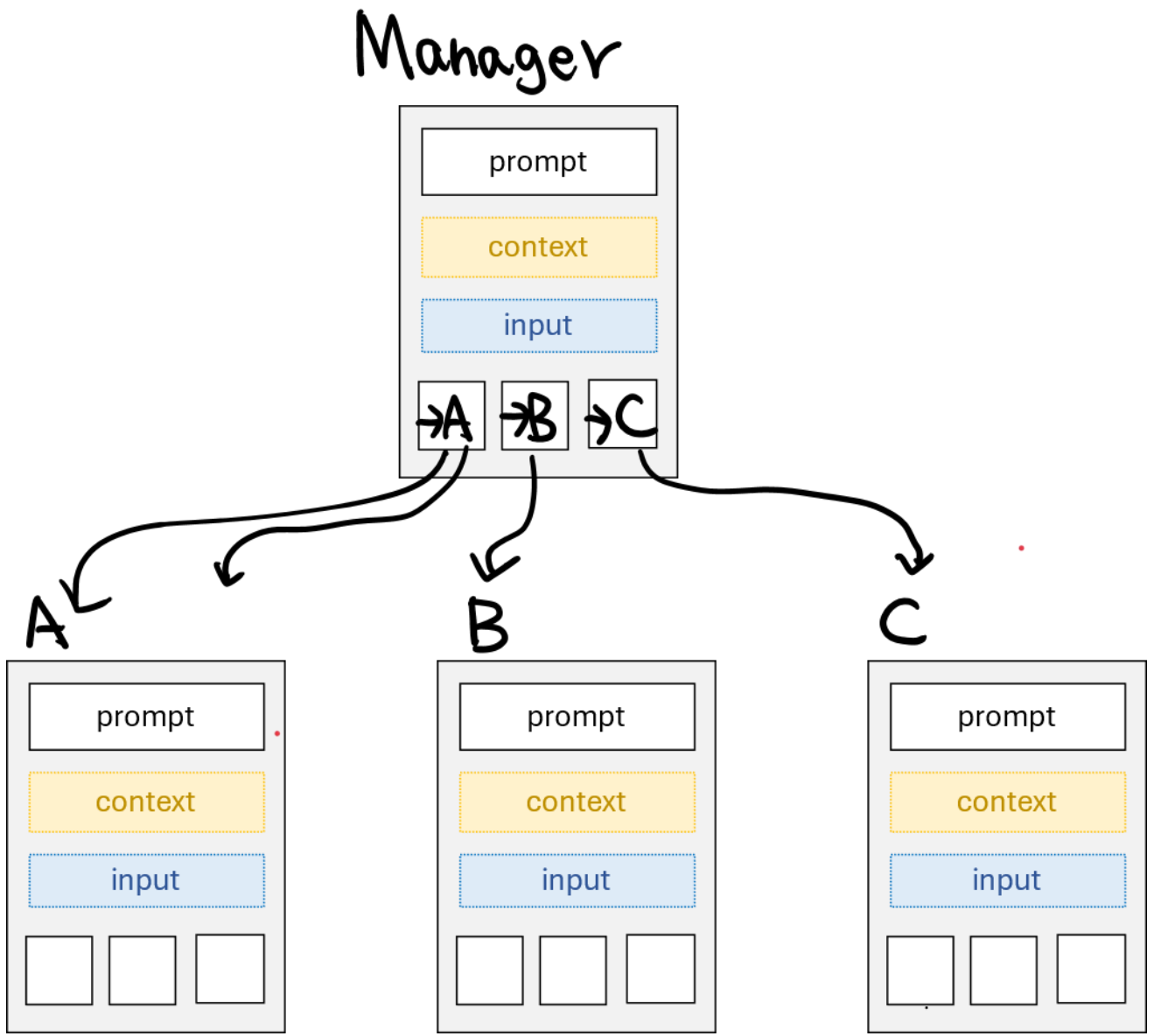
```python
def _create_manager_agent(self):
    i18n = I18N(prompt_file=self.prompt_file)
    if self.manager_agent is not None:
        self.manager_agent.allow_delegation = True
        manager = self.manager_agent
        if manager.tools is not None and len(manager.tools) > 0:
            self._logger.log(
                "warning", "Manager agent should not have tools", color="orange"
            )
            manager.tools = []
            raise Exception("Manager agent should not have tools")
        manager.tools = self.manager_agent.get_delegation_tools(self.agents)
    else:
        self.manager_llm = (
            getattr(self.manager_llm, "model_name", None)
            or getattr(self.manager_llm, "deployment_name", None)
            or self.manager_llm
        )
        manager = Agent(
            role=i18n.retrieve("hierarchical_manager_agent", "role"),
            goal=i18n.retrieve("hierarchical_manager_agent", "goal"),
            backstory=i18n.retrieve("hierarchical_manager_agent", "backstory"),
            tools=AgentTools(agents=self.agents).tools(),
            llm=self.manager_llm,
            verbose=self.verbose,
        )
        self.manager_agent = manager
    manager.crew = self
```



Graph


Manager
prompt
context
input
A
B
C
A
B
C
prompt
context
input
prompt
context
input
prompt
context
input

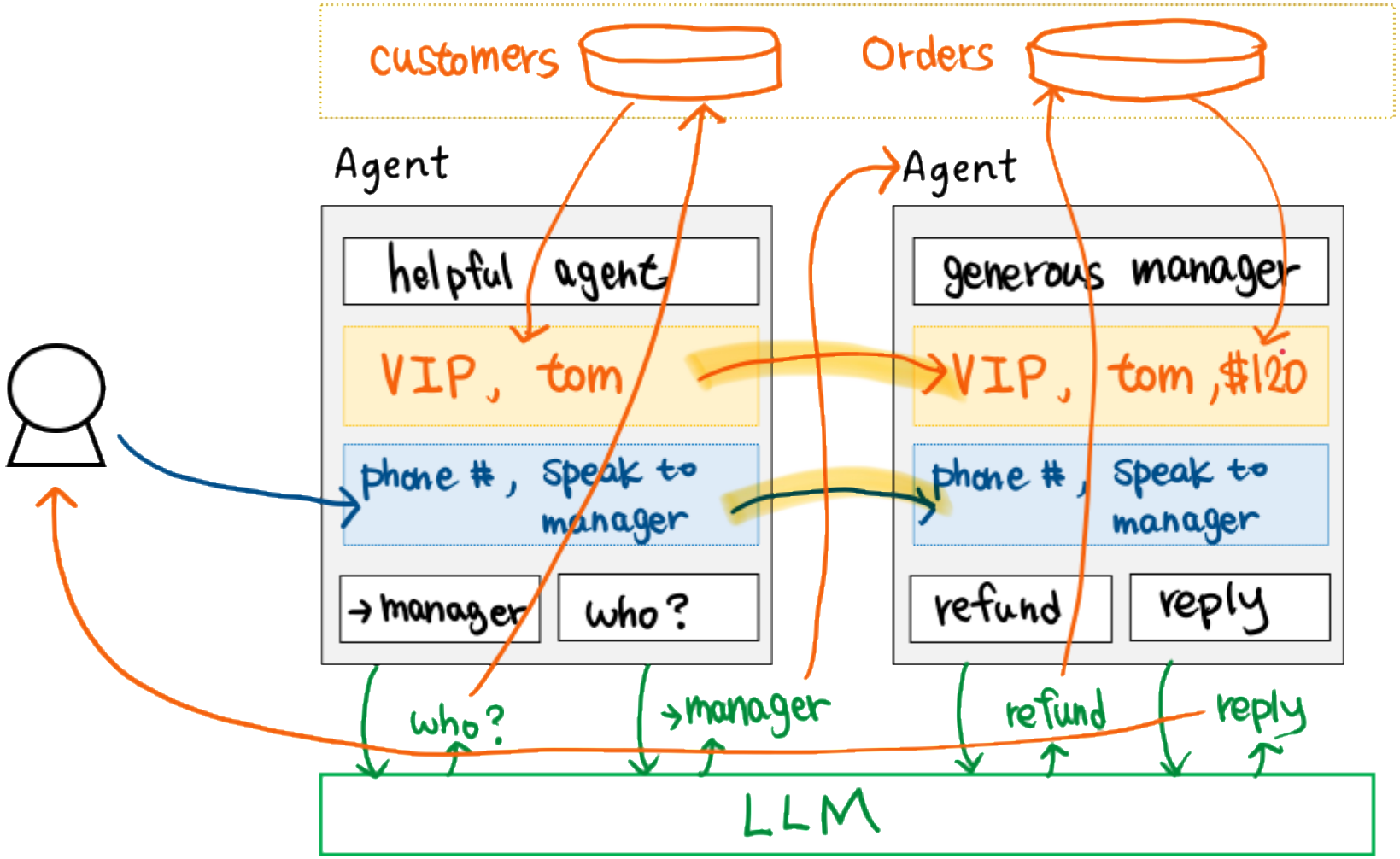
Graph


Manager
prompt
context
input
A
B
C
A
B
C
prompt
context
input
prompt
context
input
prompt
context
input

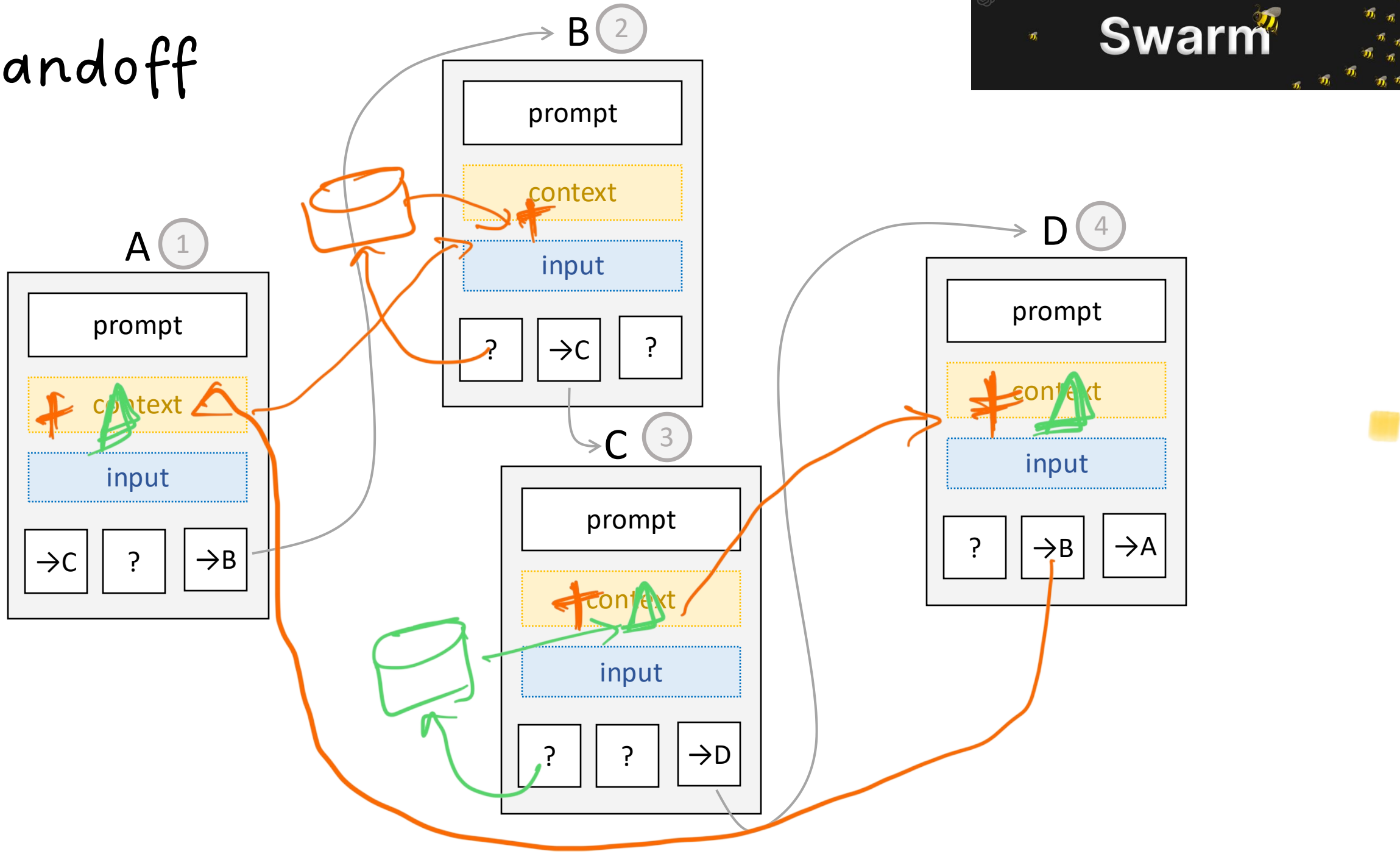
Communication

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder

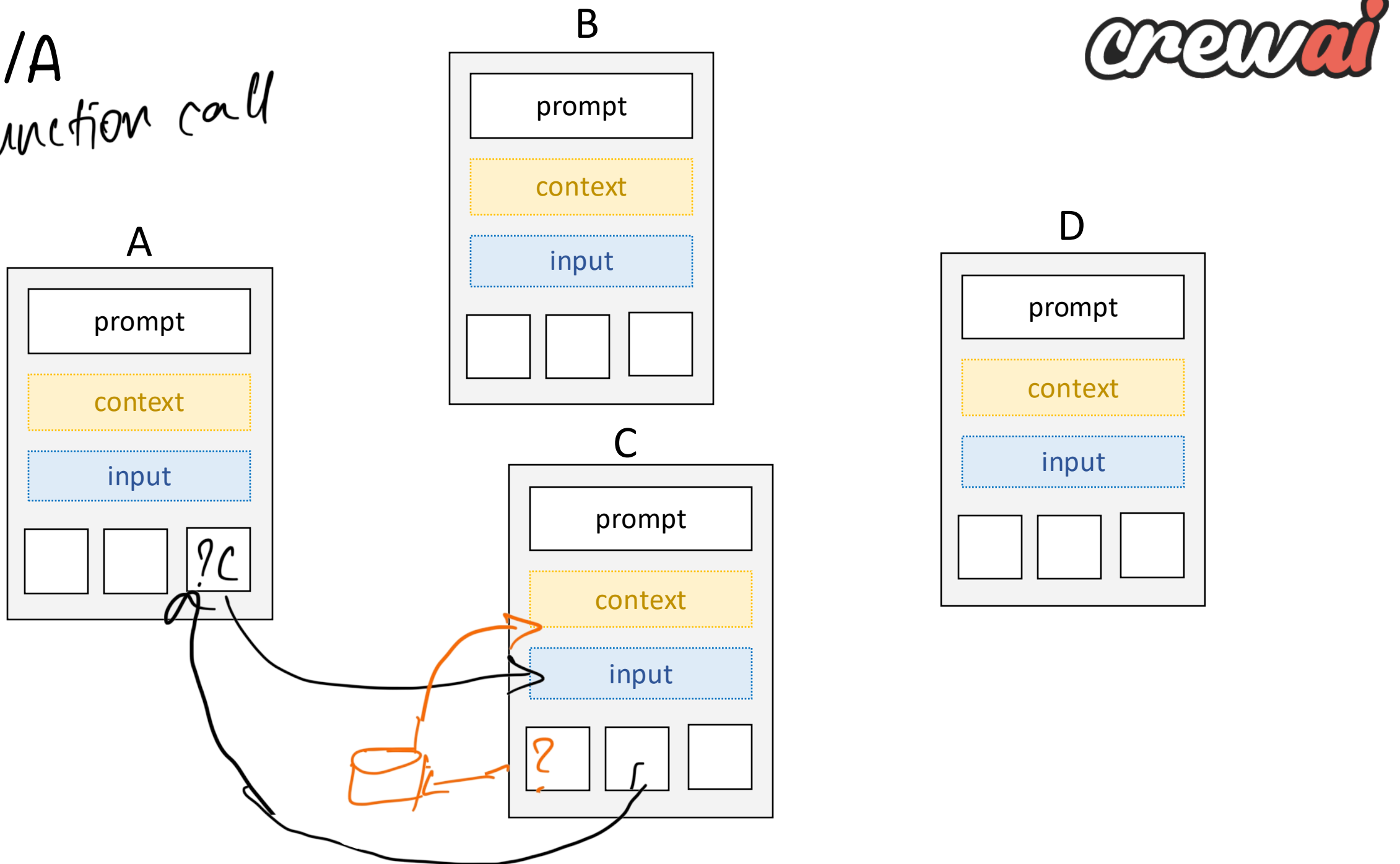

customers
Orders
Agent
Agent
helpful agent
generous manager
VIP, tom
VIP, tom, $120
phone #, Speak to manager
phone #, Speak to manager
→manager
who?
refund
reply
who?
→manager
refund
reply
LLM

Handoff

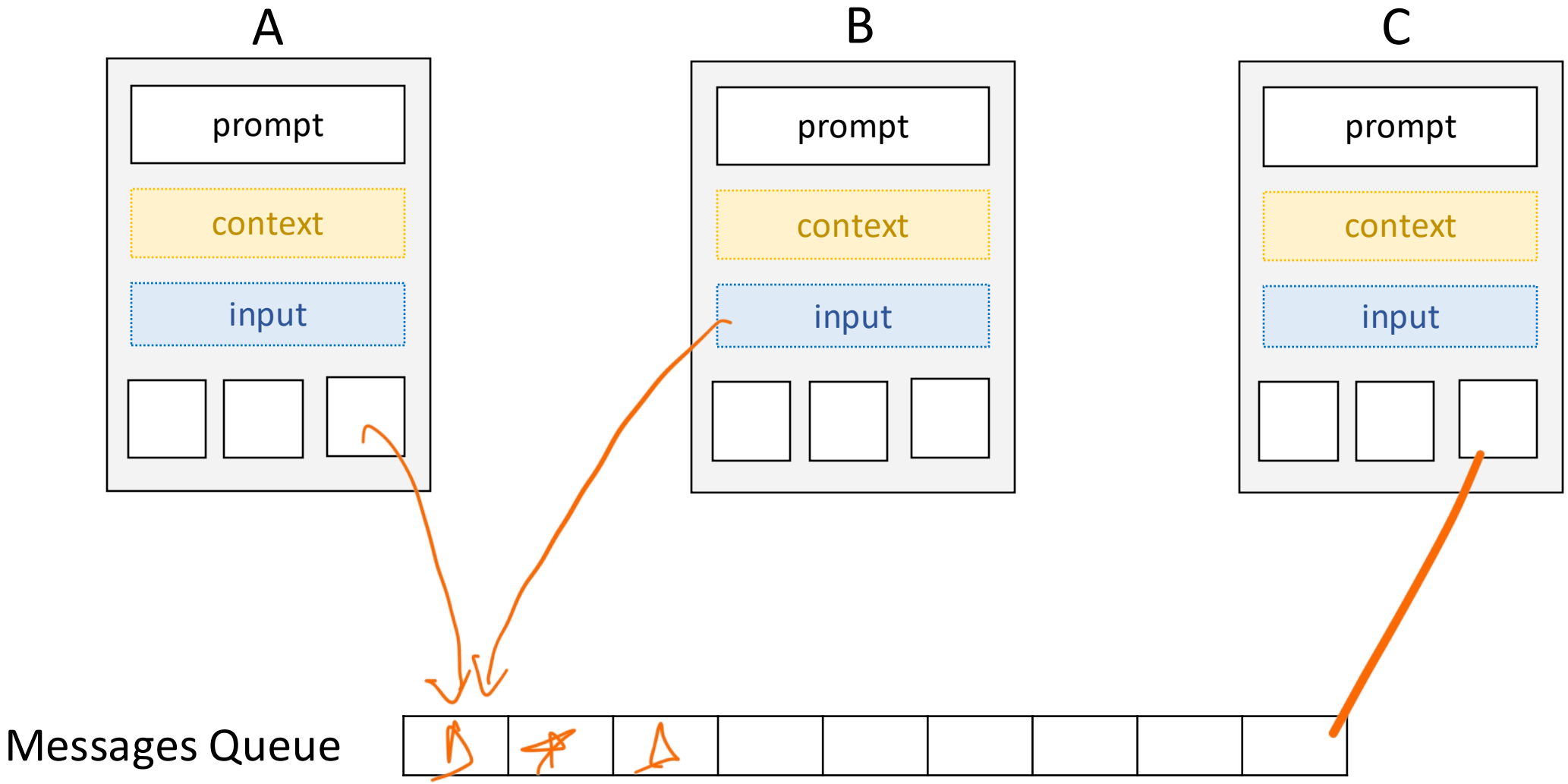

Swarm
A
1
prompt
context
input
→C
?
→B
B
2
prompt
context
input
?
→C
?
C
3
prompt
context
input
?
?
→D
D
4
prompt
context
input
?
→B
→A

Q/A

function call



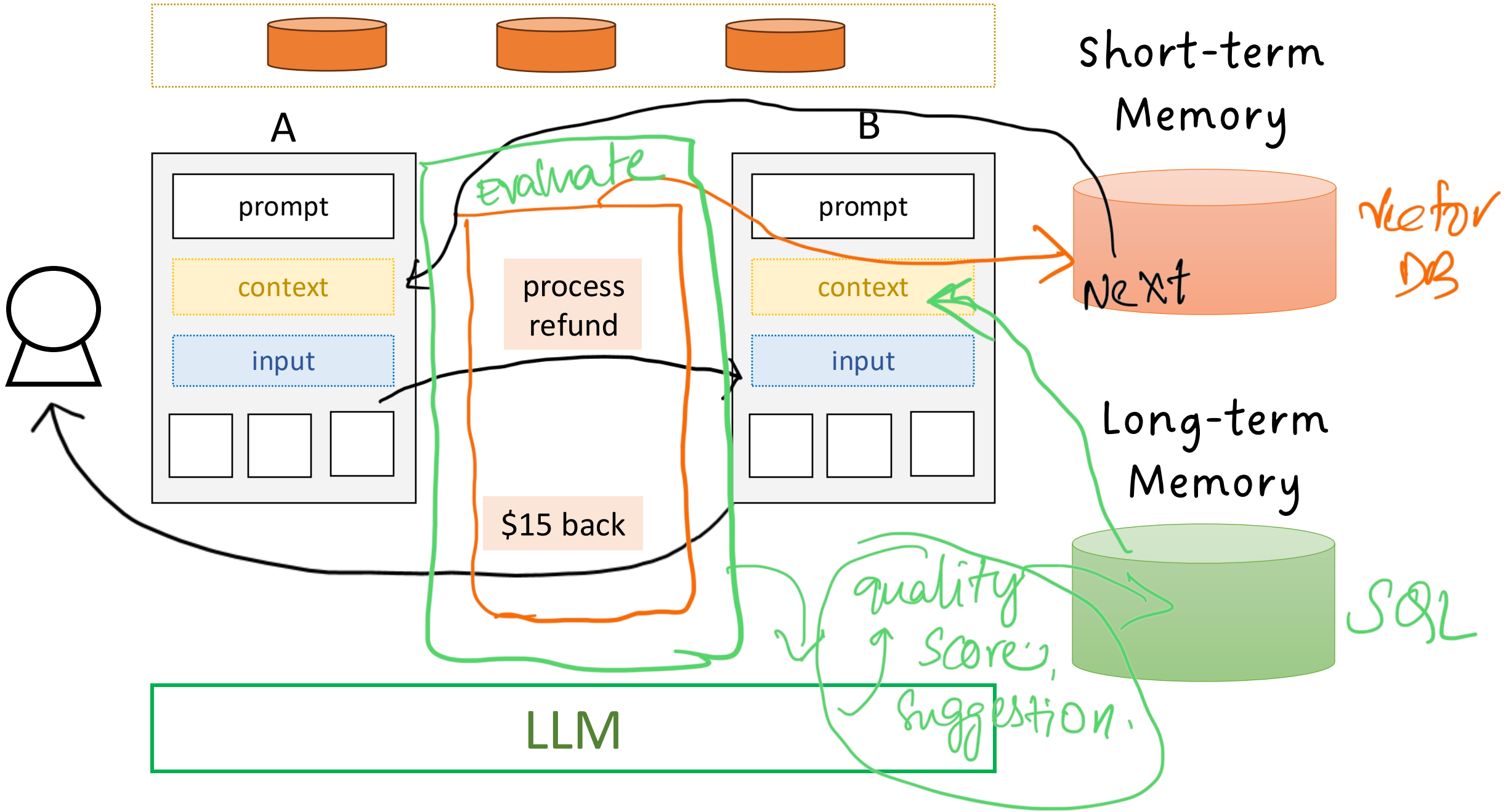
Messages


A
prompt
context
input
B
prompt
context
input
C
prompt
context
input
Messages Queue

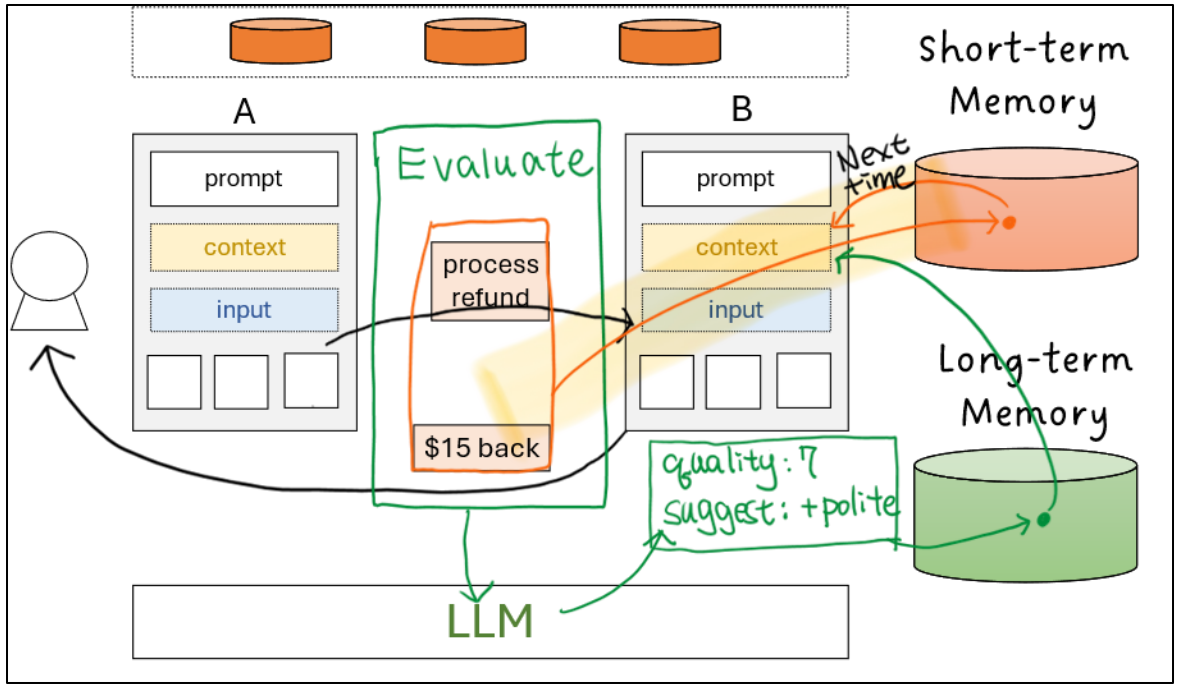
Memory

Lesson 2: Multi-Agents - by Hand ✍️

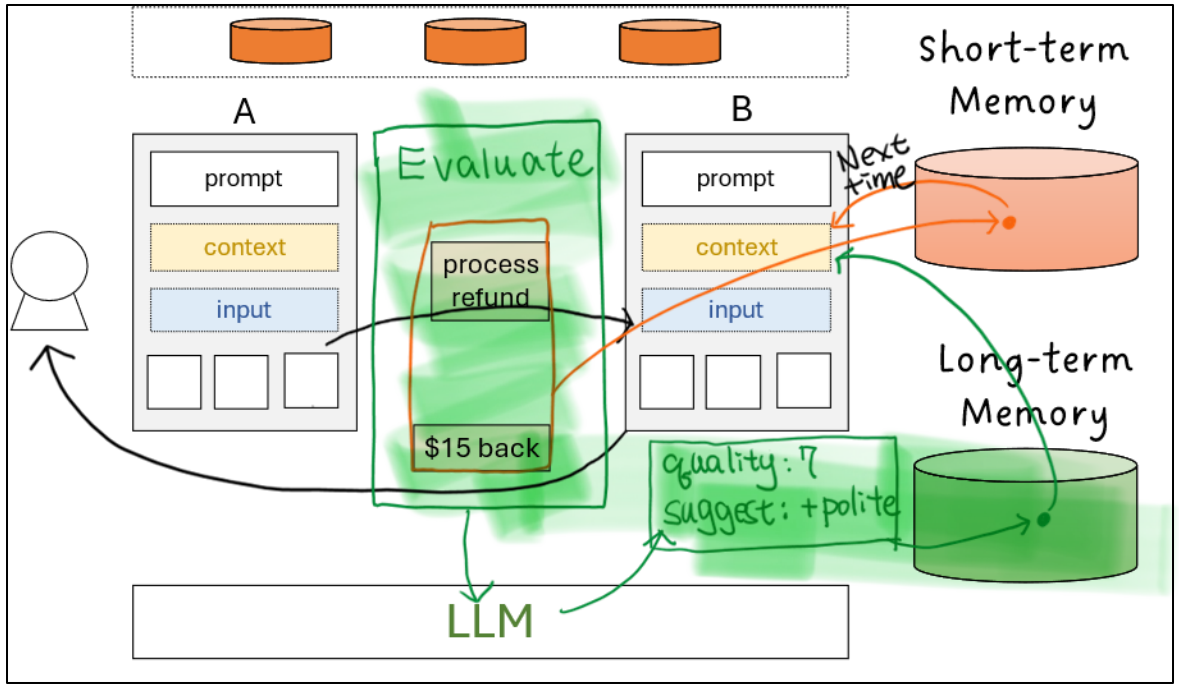
University of Colorado
Boulder


Short-term Memory
A
B
prompt
context
input
prompt
context
input
evaluate
process refund
$15 back
Next
Vector DB
Long-term Memory
SQL
quality score, suggestion.
LLM

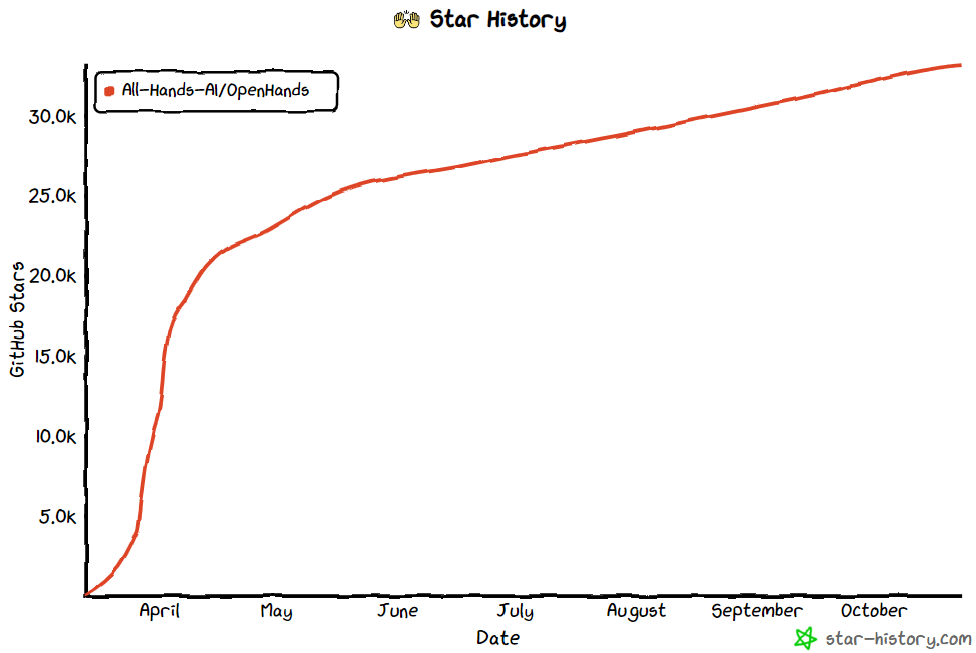
```python
def _create_short_term_memory(self, output) -> None:
    """Create and save a short-term memory item if conditions are met."""
    if (
        self.crew
        and self.agent
        and self.task
        and "Action: Delegate work to coworker" not in output.text
    ):
        try:
            if (
                hasattr(self.crew, "_short_term_memory")
                and self.crew._short_term_memory
            ):
                self.crew._short_term_memory.save(
                    value=output.text,
                    metadata={
                        "observation": self.task.description,
                    },
                    agent=self.agent.role,
                )
        except Exception as e:
            print(f"Failed to add to short term memory: {e}")
            pass
```



```python
def _create_long_term_memory(self, output) -> None:
    """Create and save long-term and entity memory items based on evaluation."""
    if (
        self.crew
        and self.crew.memory
        and self.crew._long_term_memory
        and self.crew._entity_memory
        and self.task
        and self.agent
    ):
        try:
            ltm_agent = TaskEvaluator(self.agent)
            evaluation = ltm_agent.evaluate(self.task, output.text)

            if isinstance(evaluation, ConverterError):
                return

            long_term_memory = LongTermMemoryItem(
                task=self.task.description,
                agent=self.agent.role,
                quality=evaluation.quality,
                datetime=str(time.time()),
                expected_output=self.task.expected_output,
                metadata={
                    "suggestions": evaluation.suggestions,
                    "quality": evaluation.quality,
                },
            )
            self.crew._long_term_memory.save(long_term_memory)
```

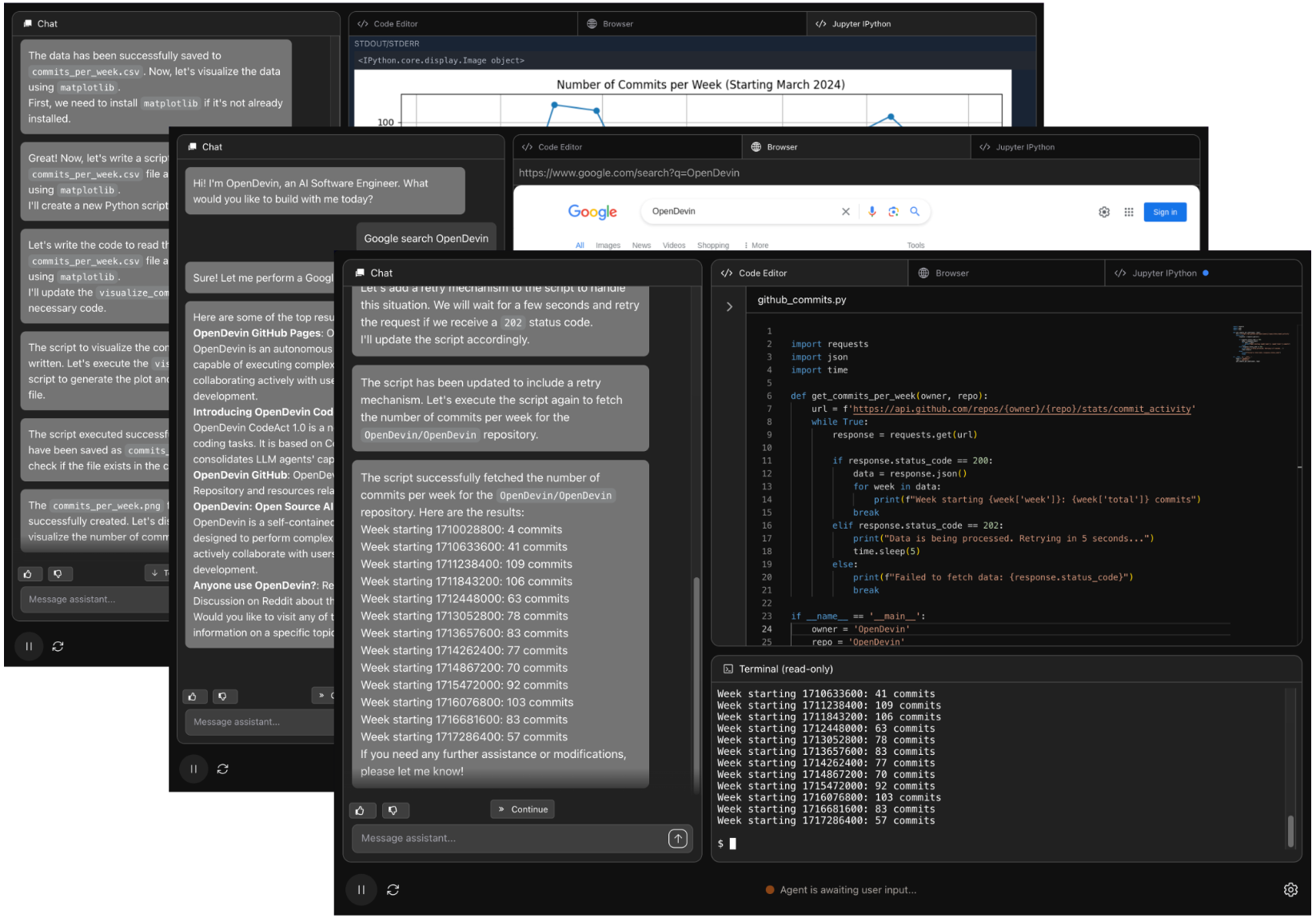


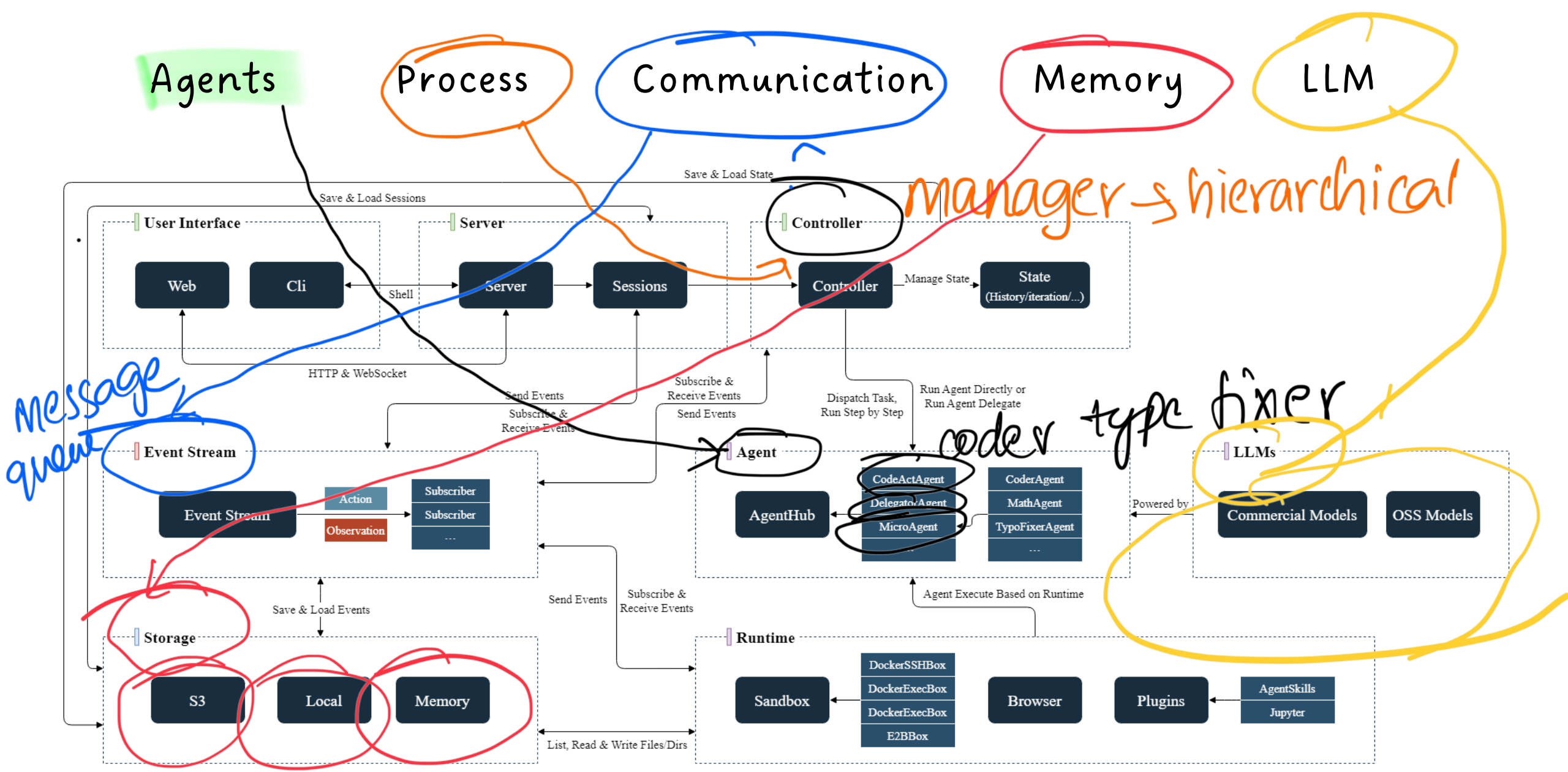
OpenHands



Lesson 2: Multi-Agents - by Hand

University of Colorado Boulder


Chat
The data has been successfully saved to commits_per_week.csv. Now, let's visualize the data using matplotlib.
First, we need to install matplotlib if it's not already installed.
Message assistant...
Code Editor
Browser
Jupyter IPython
STDOUT/STDERR
<IPython.core.display.Image object>
Number of Commits per Week (Starting March 2024)
100
Chat
Hi! I'm OpenDevin, an AI Software Engineer. What would you like to build with me today?
Google search OpenDevin
Sure! Let me perform a Googl
Here are some of the top resu
OpenDevin GitHub Pages: O
OpenDevin is an autonomous
capable of executing complex
collaborating actively with use
development.
Introducing OpenDevin Cod
OpenDevin CodeAct 1.0 is a n
coding tasks. It is based on C
consolidates LLM agents' cap
OpenDevin GitHub: OpenDev
Repository and resources rela
OpenDevin: Open Source AI
OpenDevin is a self-containec
designed to perform complex
actively collaborate with users
development.
Anyone use OpenDevin?: Re
Discussion on Reddit about th
Would you like to visit any of t
information on a specific topic
Message assistant...
Code Editor
Browser
Jupyter IPython
https://www.google.com/search?q=OpenDevin
Google
OpenDevin
Sign in
All
Images
News
Videos
Shopping
More
Tools
Chat
this situation. We will wait for a few seconds and retry the request if we receive a 202 status code.
I'll update the script accordingly.
The script has been updated to include a retry mechanism. Let's execute the script again to fetch the number of commits per week for the OpenDevin/OpenDevin repository.
The script successfully fetched the number of commits per week for the OpenDevin/OpenDevin repository. Here are the results:
Week starting 1710028800: 4 commits
Week starting 1710633600: 41 commits
Week starting 1711238400: 109 commits
Week starting 1711843200: 106 commits
Week starting 1712448000: 63 commits
Week starting 1713052800: 78 commits
Week starting 1713657600: 83 commits
Week starting 1714262400: 77 commits
Week starting 1714867200: 70 commits
Week starting 1715472000: 92 commits
Week starting 1716076800: 103 commits
Week starting 1716681600: 83 commits
Week starting 1717286400: 57 commits
If you need any further assistance or modifications, please let me know!
Continue
Message assistant...
Code Editor
Browser
Jupyter IPython
github_commits.py
import requests
import json
import time
def get_commits_per_week(owner, repo):
url = f'https://api.github.com/repos/{owner}/{repo}/stats/commit_activity'
while True:
response = requests.get(url)
if response.status_code == 200:
data = response.json()
for week in data:
print(f"Week starting {week['week']}: {week['total']} commits")
break
elif response.status_code == 202:
print("Data is being processed. Retrying in 5 seconds...")
time.sleep(5)
else:
print(f"Failed to fetch data: {response.status_code}")
break
if __name__ == '__main__':
owner = 'OpenDevin'
repo = 'OpenDevin'
Terminal (read-only)
Week starting 1710633600: 41 commits
Week starting 1711238400: 109 commits
Week starting 1711843200: 106 commits
Week starting 1712448000: 63 commits
Week starting 1713052800: 78 commits
Week starting 1713657600: 83 commits
Week starting 1714262400: 77 commits
Week starting 1714867200: 70 commits
Week starting 1715472000: 92 commits
Week starting 1716076800: 103 commits
Week starting 1716681600: 83 commits
Week starting 1717286400: 57 commits
$
Agent is awaiting user input...


Agents
Process
Communication
Memory
LLM
manager → hierarchical
message queue
coder
typo fixer
Save & Load State
Save & Load Sessions
User Interface
Server
Controller
Web
Cli
Shell
Server
Sessions
Controller
Manage State
State
(History/iteration/...)
HTTP & WebSocket
Send Events
Subscribe & Receive Events
Subscribe & Receive Events
Send Events
Dispatch Task, Run Step by Step
Run Agent Directly or Run Agent Delegate
Event Stream
Agent
LLMs
Event Stream
Action
Observation
Subscriber
Subscriber
...
AgentHub
CodeActAgent
DelegatorAgent
MicroAgent
...
CoderAgent
MathAgent
TypoFixerAgent
...
Powered by
Commercial Models
OSS Models
Save & Load Events
Send Events
Subscribe & Receive Events
Agent Execute Based on Runtime
Storage
Runtime
S3
Local
Memory
Sandbox
DockerSSHBox
DockerExecBox
DockerExecBox
E2BBox
Browser
Plugins
AgentSkills
Jupyter
List, Read & Write Files/Dirs

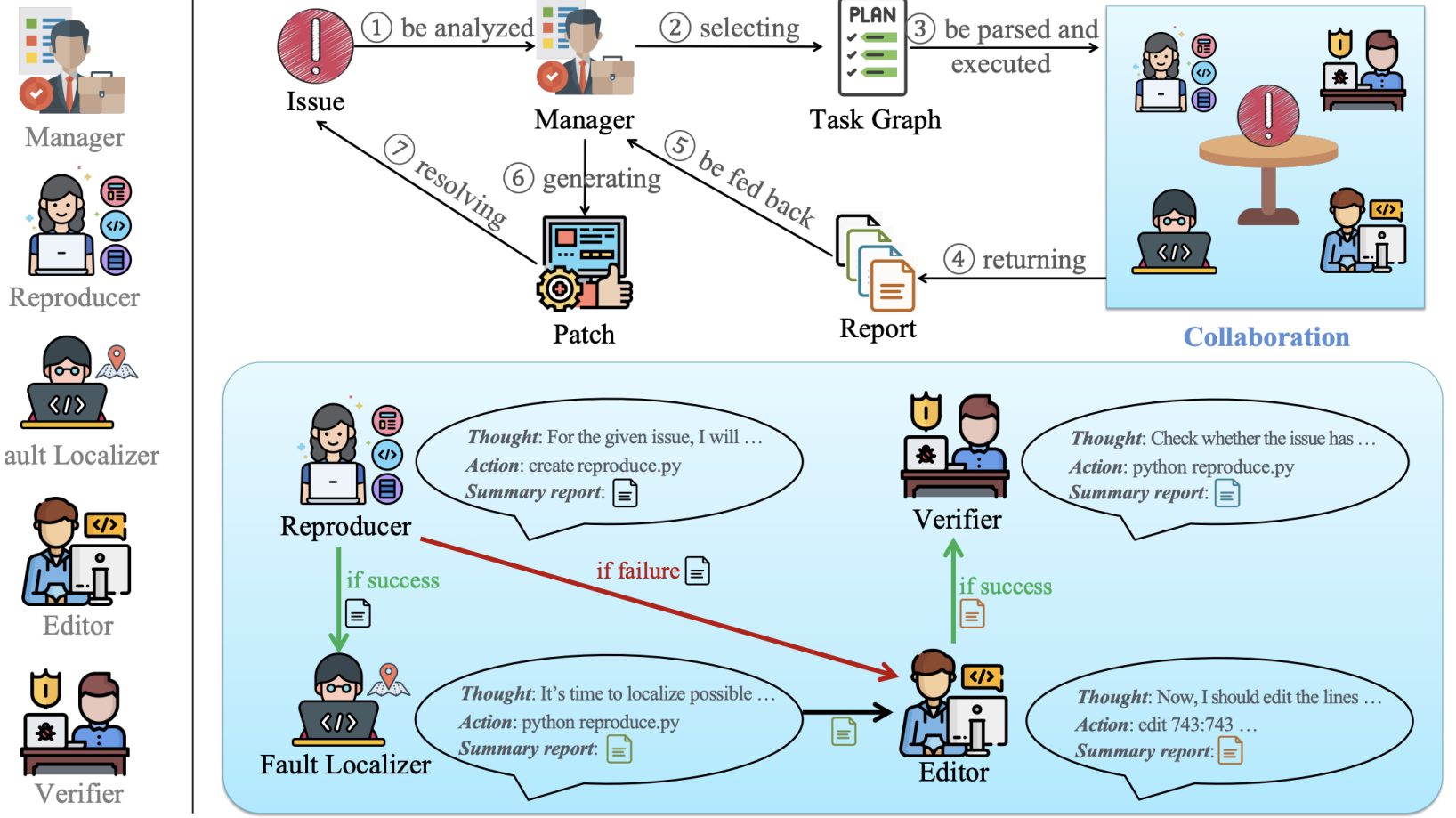
Jul 4, 2024

CodeR

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder

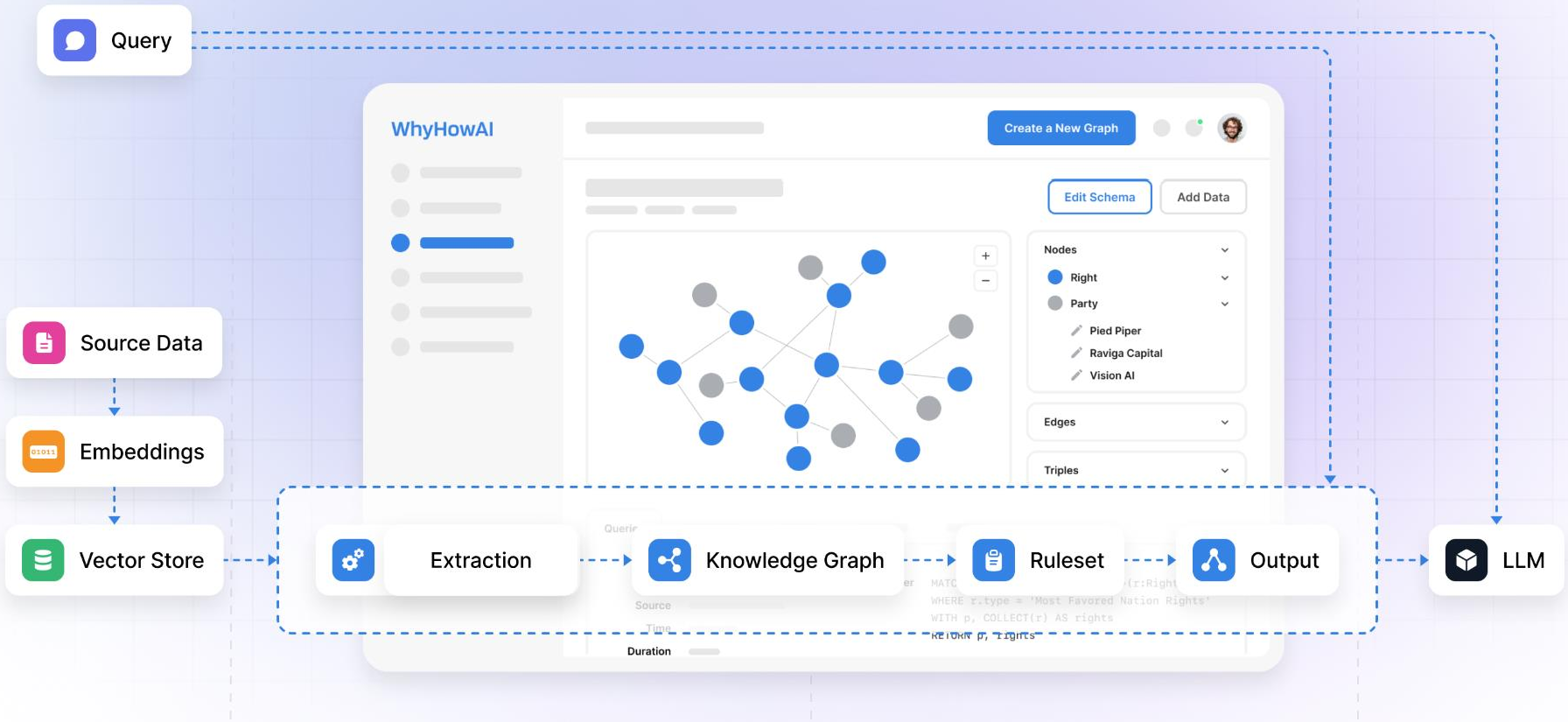
Agents Process Communication Memory LLM



Multi-Agents + Graphs WhyHow.ai

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado Boulder


Query
WhyHowAI
Create a New Graph
Edit Schema
Add Data
Nodes
Right
Party
Pied Piper
Raviga Capital
Vision AI
Edges
Triples
Source Data
Embeddings
Vector Store
Extraction
Knowledge Graph
Ruleset
Output
LLM
Duration

Legal

The management of control functions by the Board and the Chief Compliance Officer (CCO) involves several key responsibilities as outlined in the compliance document:

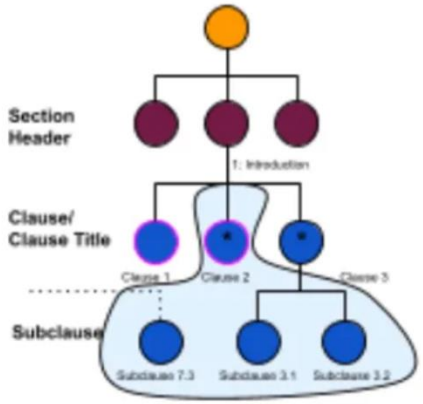
Responsibilities of the Board:
1. Oversight of Compliance Risk Management: The Board is responsible for overseeing the management of compliance risk within the financial institution. T
2. Approval and Support for the Compliance Function: The Board must ensure that the compliance function and the CCO have the appropriate standing, authority, and independence to carry out their duties effectively. The Board must also ensure that the compliance function is adequately resourced.
3.Engagement with the CCO: The Board should engage with the CCO regularly to discuss issues faced by the compliance function and ensure the CCO has direct access to the Board.
4. Annual Evaluation: The Board must evaluate the effectiveness of the institution's management of compliance risk at least annually, considering assessments from senior management, internal audit, and the CCO.

Responsibilities of the Chief Compliance Officer (CCO):
1. Coordination and Management of Compliance Risk: The CCO is responsible for coordinating the identification, management, and monitoring of compliance risk across the institution. This includes ensuring compliance monitoring and testing are consistent across the organization.
2. Reporting and Advising: The CCO must report regularly to senior management on the findings and analysis of compliance risks and must ensure that reports are readily available to internal audit and regulatory authorities. The CCO also advises the Board and senior management on legal and regulatory requirements, keeping them informed of developments and their implications.
3. Independence: The CCO must maintain independence from business lines to effectively carry out the role of a control function. This includes ensuring that there is no conflict of interest in their responsibilities, reporting lines, or remuneration.
4. Resources and Training: The CCO must ensure that the compliance function is sufficiently resourced with officers who have the necessary qualifications and experience. The CCO is also responsible for ensuring that adequate training is provided to officers on relevant legal and regulatory requirements.

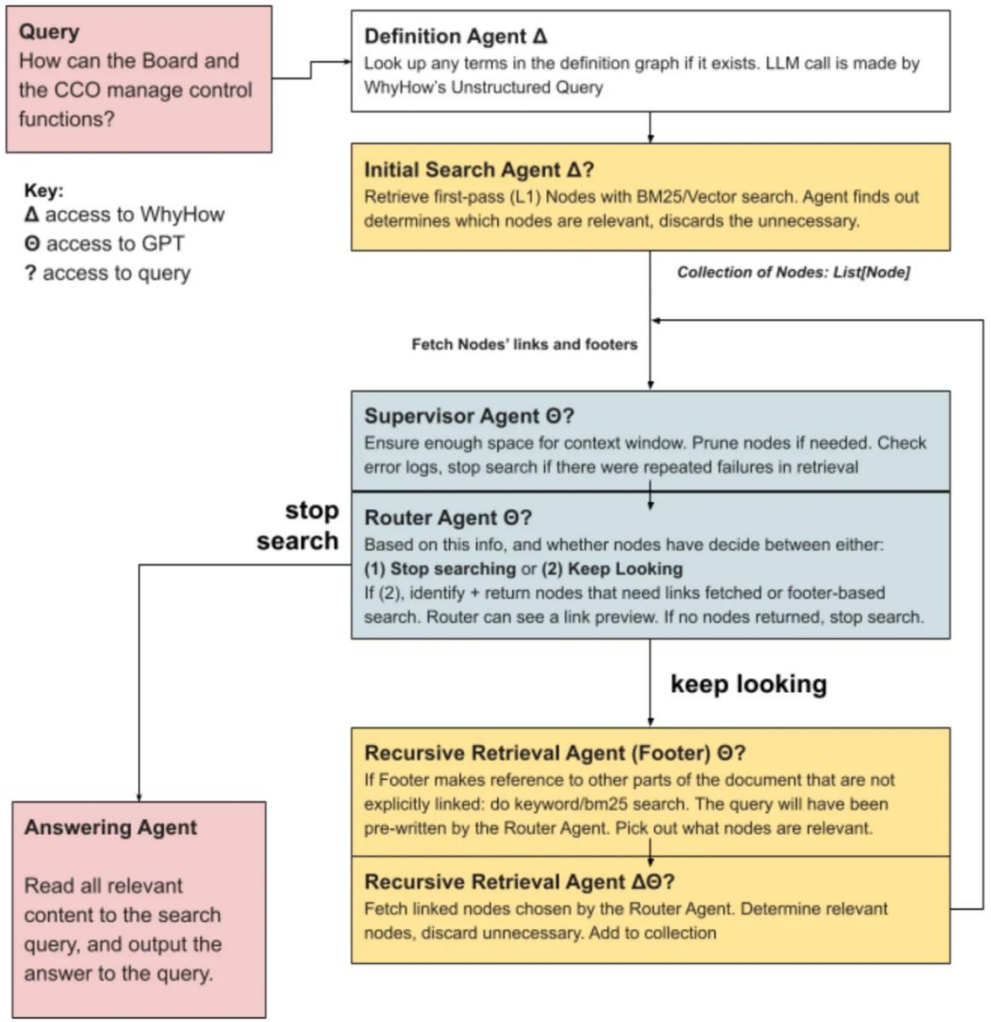
Interaction with Other Control Functions:
The compliance function should be organized in a way that allows effective management of compliance risk, considering the size, nature, and complexity of the financial institution's operations. Where responsibilities are shared between a dedicated compliance unit and other control functions, clear documentation and coordination are required to ensure that the CCO can perform their duties effectively without impairing independence or focus.
By fulfilling these responsibilities, the Board and the CCO can ensure that the institution's compliance risks are managed effectively, supporting the overall safety and soundness of the financial institution.

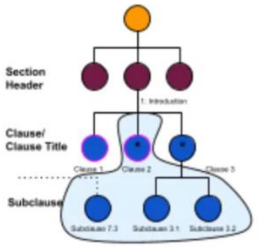
Query

How can the board and the CCO manage control functions?



Agents Process Communication Memory LLM





Collections of retrieved subgraphs are passed through the pipeline (cyan highlight)

Nodes marked with * have extra footer info

Nodes with pink outline a link to another link

As the collection passes through the pipeline, they can be 'marked', either for redundancy, added extra context e.g. what section the clause is in

Some agents/tools can add to the collection

Node:
- id : str
- type : ElementType
- content : str
- context: Dict[str, Any]
- children: Dict[str, Node]

For those looking to understand what each agent does, we include code snippets for each agent in the Appendix

https://medium.com/enterprise-rag/legal-document-rag-multi-graph-multi-agent-recursive-retrieval-through-legal-clauses-c90e073e0052

Assignment

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder

Prizes

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder

Q/A

Lesson 2: Multi-Agents - by Hand ✍️

University of Colorado
Boulder