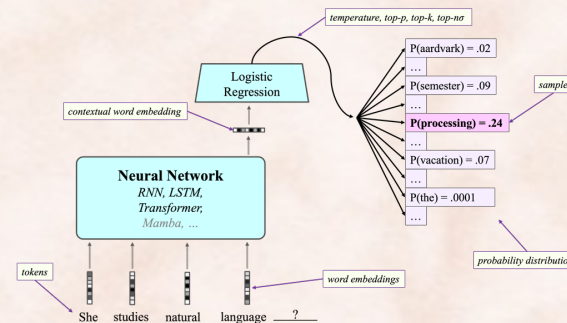


ITCS 4101: Introduction to NLP

LLMs and Agentic Workflows



Razvan C. Bunescu

Department of Computer Science @ CCI

rbunescu@charlotte.edu

Agentic AI

- Slides extracted from the wonderful [Agentic AI](#) course taught by Andrew Ng on the [DeepLearning.AI](#) platform.

Module 1: Introduction to Agentic Workflows



Module 2: Reflection Design Pattern



Module 3: Tool use



Module 4: Practical Tips for Building Agentic AI



Module 5: Patterns for Highly Autonomous Agents



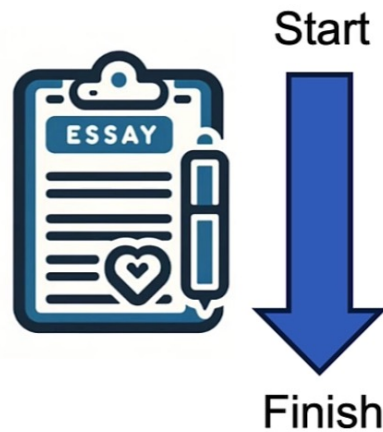
- In this lecture, we focus mostly on material from Module 1 and Module 5.

What is Agentic AI

Agentic AI

Non-agentic workflow (zero-shot):

Please type out an essay on topic X from start to finish in one go, without using backspace.



Agentic workflow:

Write an essay outline on topic X

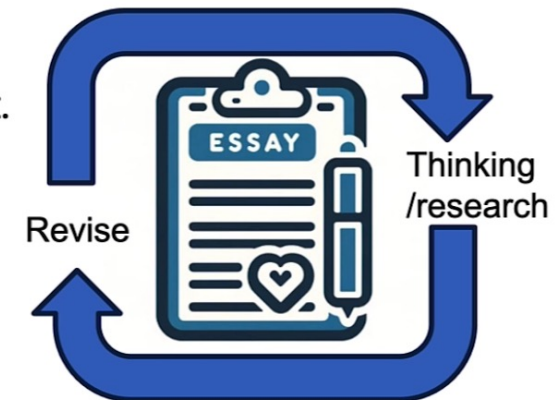
Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

Revise your draft.

....



Agentic AI Workflows

Agentic AI workflows

An agentic AI workflow is a process where an LLM-based app executes multiple steps to complete a task.

Essay-writing example:

Write an essay outline on topic X

LLM

Do you need any web research?

LLM

+

web search

Write a first draft.

LLM

Consider what parts need revision or more research.

LLM

+

request
human review

Revise your draft.

LLM

Degrees of Autonomy



Andrew Ng
AI Fund

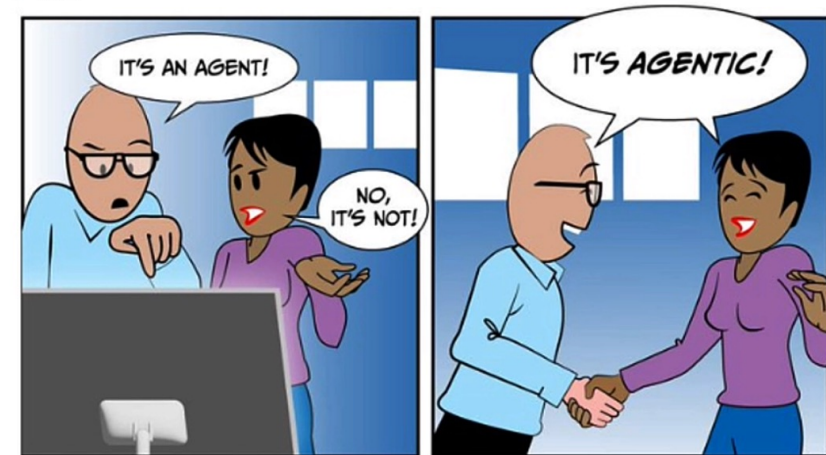
Sequoia Ascent, March 2024



....

Rather than arguing over which work to include or exclude as being a true agent, we can acknowledge that there are different degrees to which systems can be agentic.

....



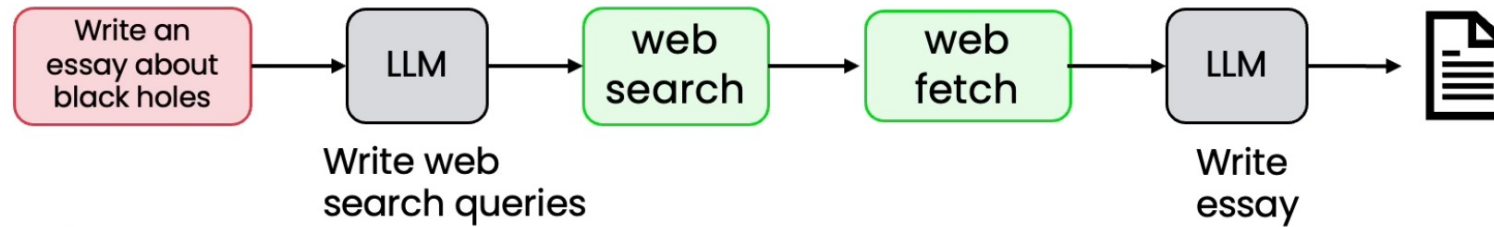
X (twitter) post , June 2024

Andrew Ng

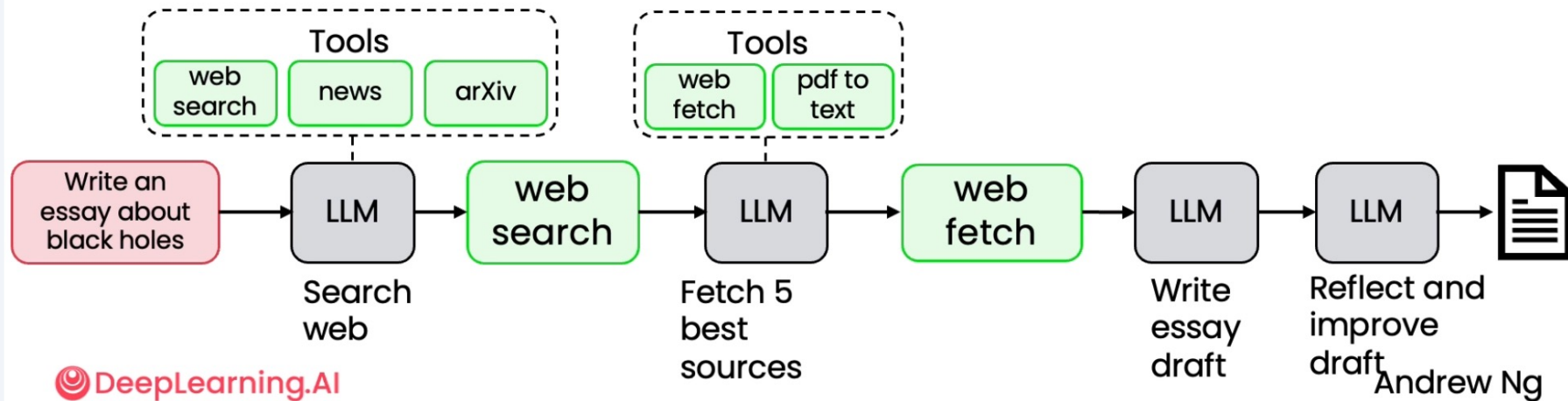
Degrees of Autonomy

Degrees of autonomy

Less autonomous



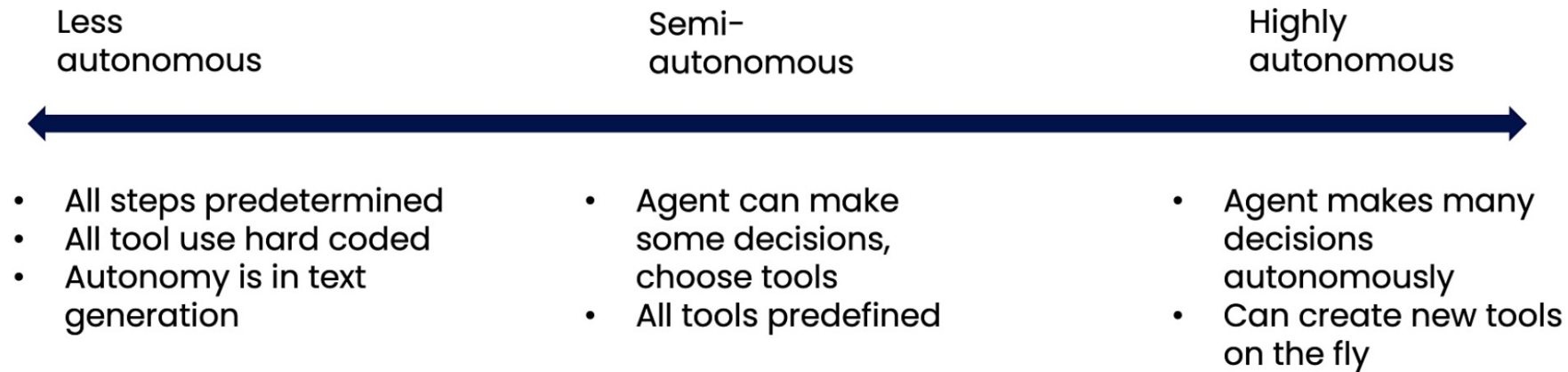
More autonomous



Degrees of Autonomy

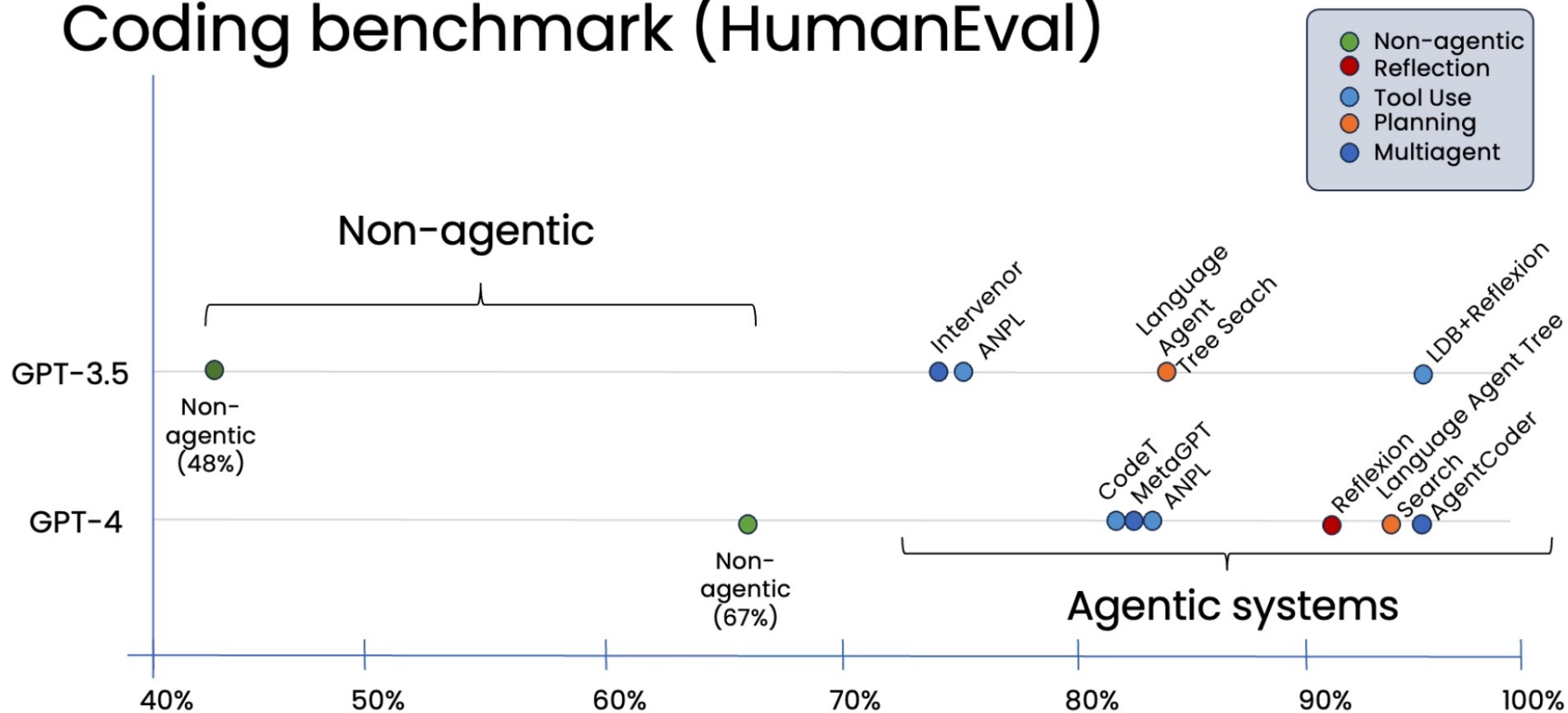
Degrees of autonomy

Agentic AI can be less or more autonomous



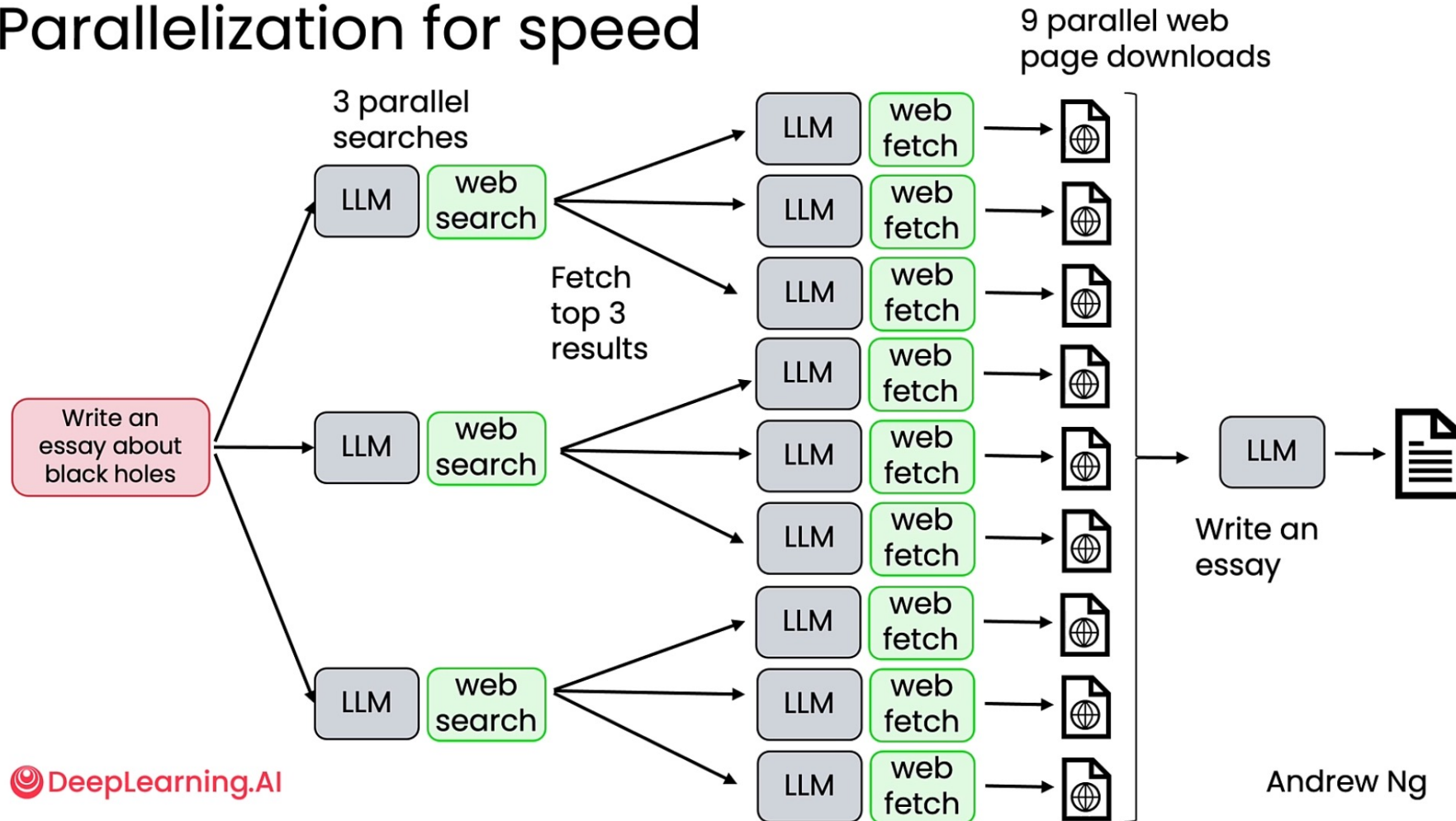
Agentic AI Improves Coding Performance

Coding benchmark (HumanEval)



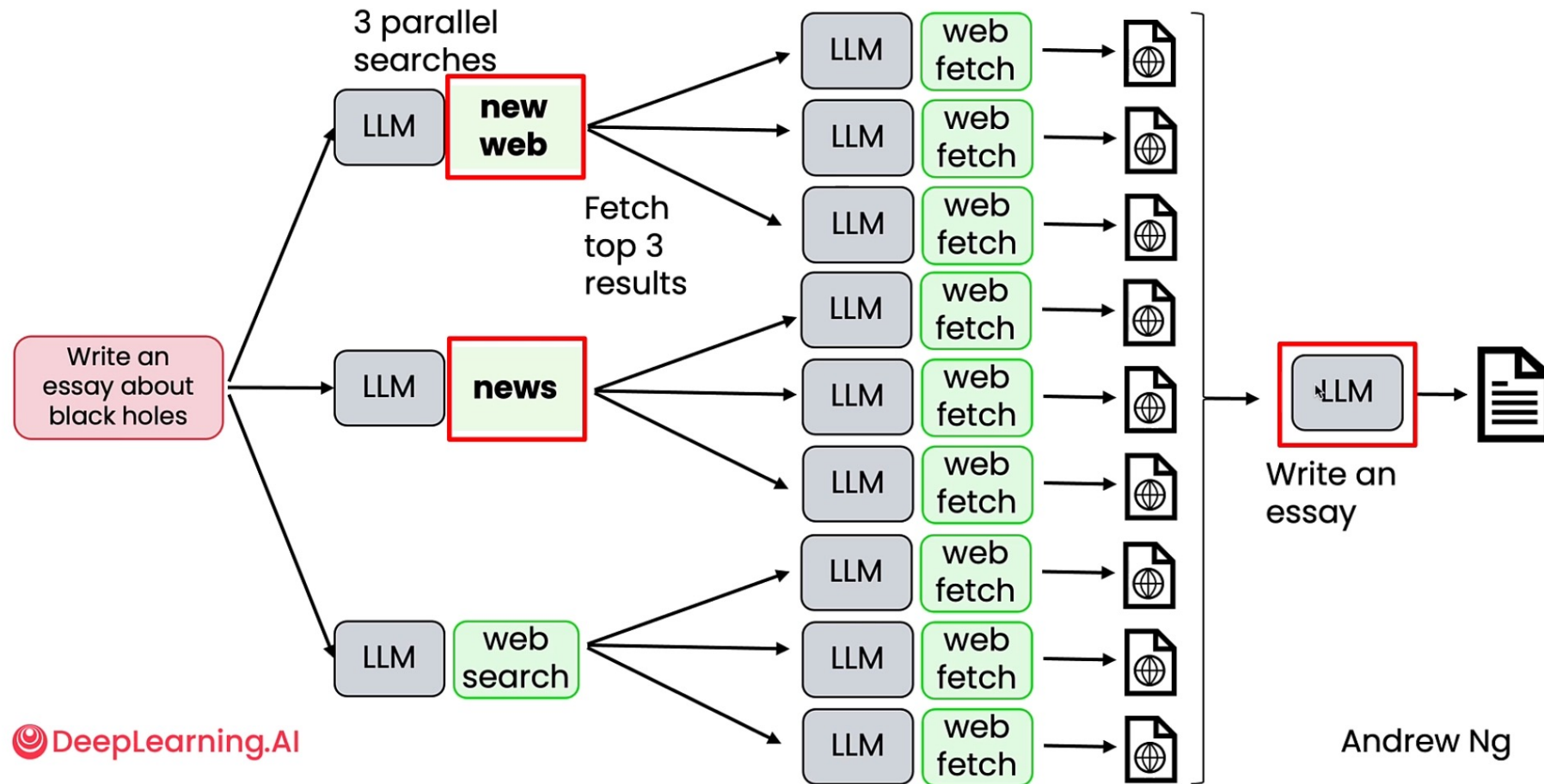
Agentic AI is Parallelizable

Parallelization for speed



Agentic AI is Modular

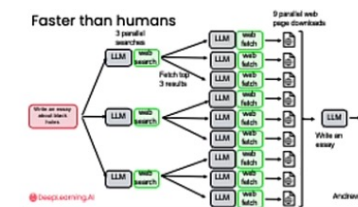
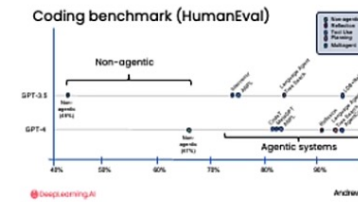
Modular: Add/swap out components



Key Benefits of Agentic AI

Key benefits of agentic workflows

- Much better performance
- Faster than humans because of parallelization
- Modular: can add or update tools, swap out models



Agentic AI Design Patterns

Agentic Design Patterns

1. Reflection
2. Tool use
3. Planning
4. Multi-agent collaboration

Reflection Design Pattern

Reflection



Please write code for {task}

`def do_task(x): ...`

`def do_task_v2(x):`

`def do_task_v3(x):`



Coder
Agent
(LLM)

Here's code intended for {task}:

```
def do_task (x):  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...

Reflection Design Pattern

Reflection



Please write code for {task}

`def do_task(x): ...`

`def do_task_v2(x):`

`def do_task_v3(x):`



Coder
Agent
(LLM)

Here's code intended for {task}:

```
def do_task (x):  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...

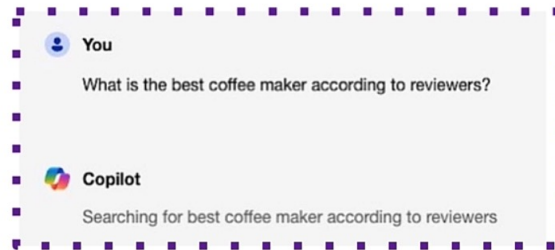


Critic
Agent
(LLM)

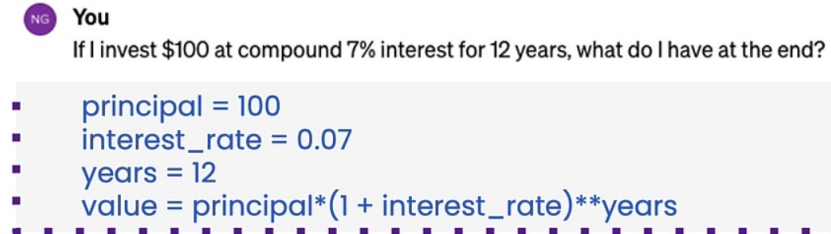
Tool Use Design Pattern

Tool use

Web search tool



Code execution tool



Analysis

- Code Execution
- Wolfram Alpha
- Bearly Code Interpreter

Information gathering

- Web search
- Wikipedia
- Database access

Productivity

- Email
- Calendar
- Messaging

Images

- Image generation
- Image captioning
- OCR

Planning Design Pattern

Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



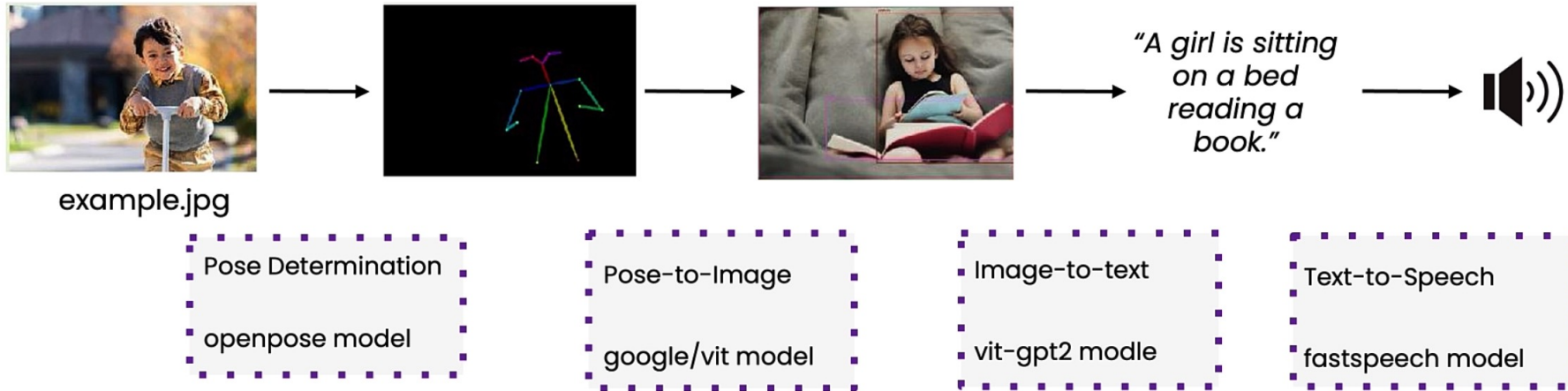
example.jpg

[Example adapted from HuggingGPT
paper]

Planning Design Pattern

Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



[Example adapted from HuggingGPT paper]

Multi-Agent Design Pattern

Multi-agent workflows



Multiagent Debate

Task	Single agent	Multi-agent
Biographies	66.0%	73.8%
MMLU	63.9%	71.1%
Chess move	29.3%	45.2%

(Du et al., 2023)

Multiple agents collaborate on a complex software development task.

Planning Design Pattern

Planning example: Customer service agent

Inventory Database

id	name	description	price	stock
1001	Aviator	Timeless pilot style for any occasion, metal frame	80	12
1002	Catseye	Glamorous 1950s profile, plastic frame	60	28
1003	Moon	Oversized round style, plastic frame	120	15
1004	Classic	Classic round profile, gold frame	60	9

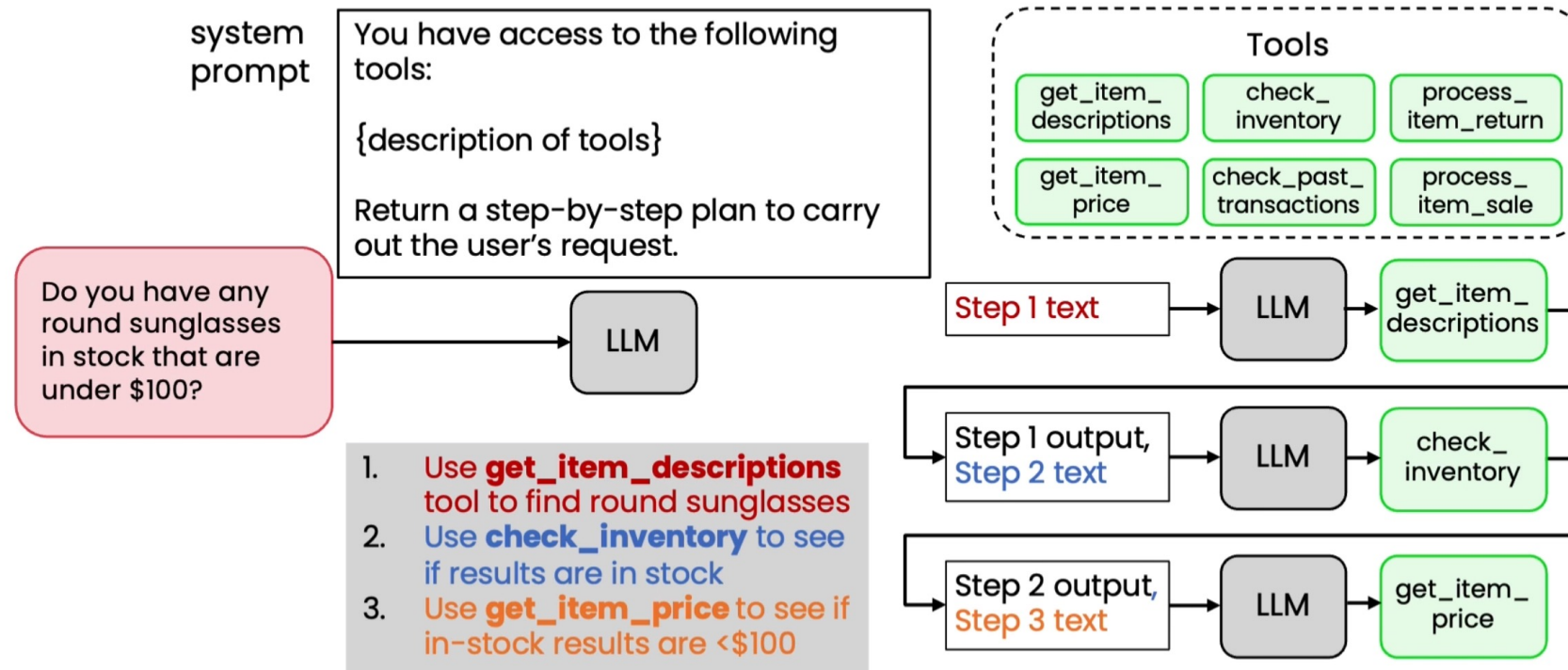
Customer query:

Do you have any **round** sunglasses in stock that are under \$100?

Yes, we have our **Classic** sunglasses, which are a classic round metal frame and cost \$60

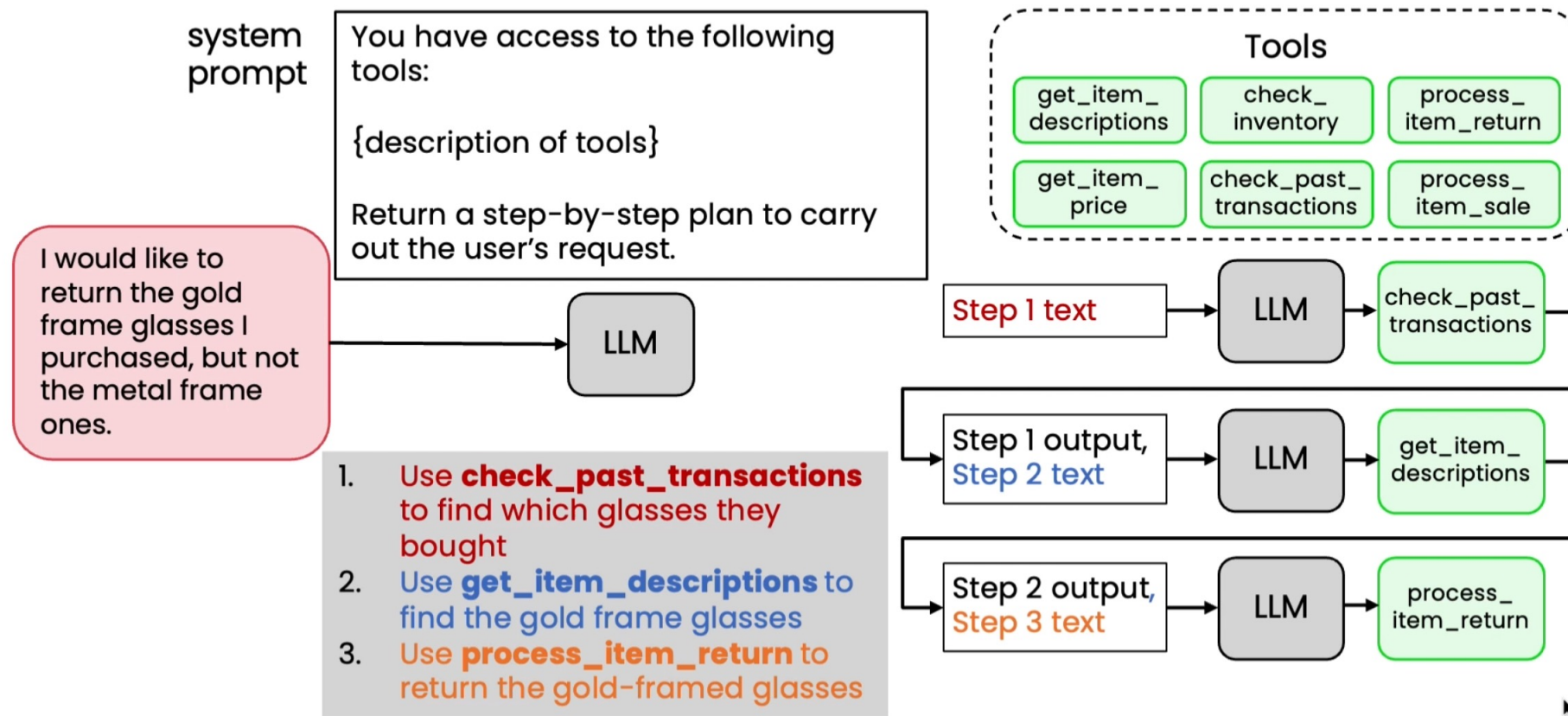
Planning with Tools

Planning example: Customer service agent



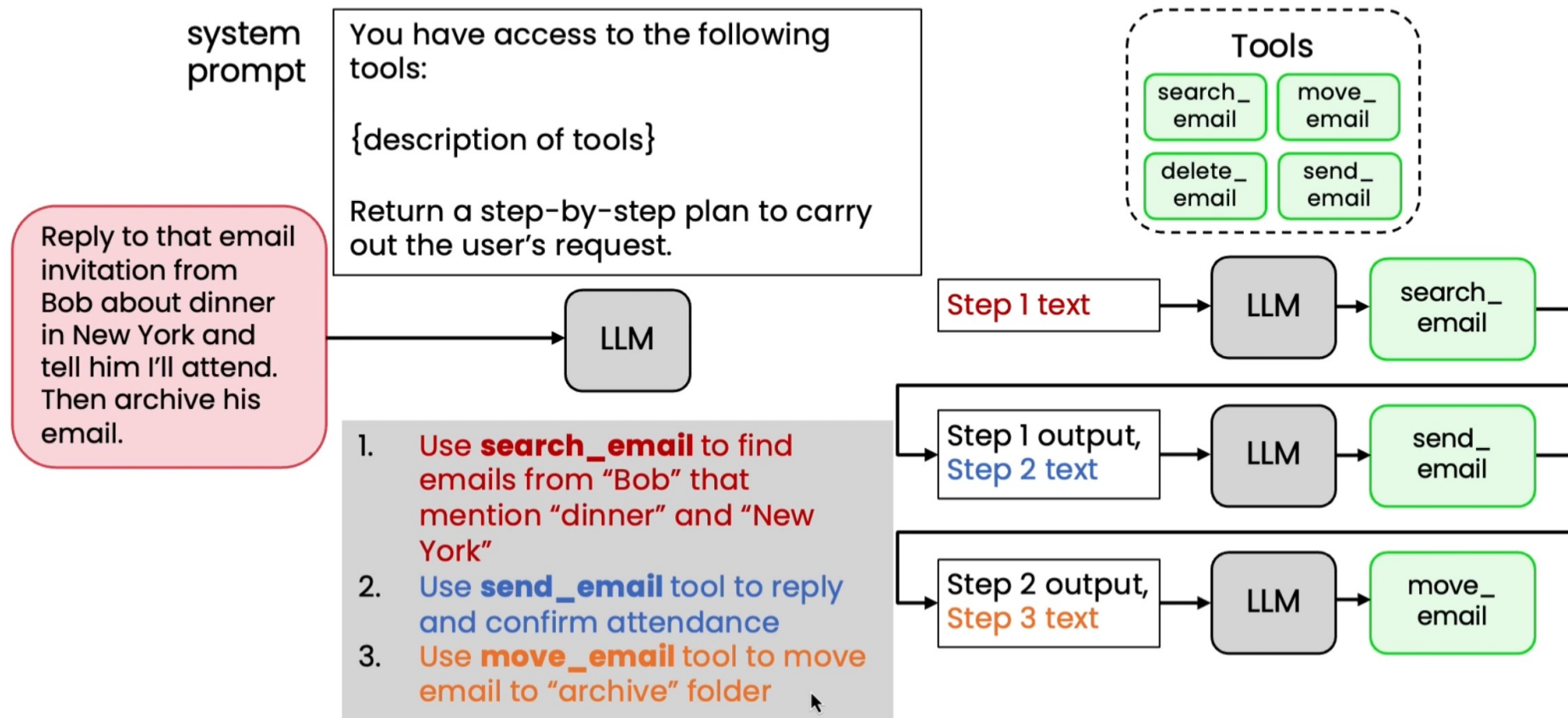
Planning with Tools

Planning example: Customer service agent



Planning with Tools

Planning example: Email assistant



Creating and Executing Plans

Formatting plan as JSON

Updated system prompt

You have access to the following tools:

{description of tools}

Create a step-by-step plan in JSON format.

Each step should have the following items:
step number, description, tool name, and
args.

Do you have any
round sunglasses
in stock that are
under \$100?

LLM

```
{
  "plan": [
    {
      "step": 1,
      "description": "Find round sunglasses",
      "tool": "get_item_descriptions",
      "args": {"query": "round sunglasses"}
    },
    {
      "step": 2,
      "description": "Check available stock",
      "tool": "check_inventory",
      "args": {"items": "results from step 1"}
    },
    ...
  ]
}
```

Limitations of Planning with Tools

The challenge of planning with tools



Which month had the highest sales of hot chocolate?

LLM

Tools

get_column_max

get_column_mean

filter_rows

get_column_min

get_column_median

sum_rows

date	price	coffee_name	Size
2024-01-28	3.87	Hot Chocolate	M
2024-03-01	2.89	Cappuccino	S
2024-03-04	3.87	Latte	M
...	...	...	...
2025-03-23	4.57	Latte	L

coffee_sales.csv

1. Use the **filter_rows** tool to extract transactions in January with coffee_name "Hot Chocolate"
2. Use the **get_column_mean** to find the average amount
3. Use the **filter_rows** tool to extract transactions in February with coffee_name "Hot Chocolate"
4. Use the **get_column_mean** to find the average amount
5. Repeat for March, April, May, ..., December
6. Determine the month with highest average using results of previous steps

Limitations of Planning with Tools

The challenge of planning with tools



Which month had the highest sales of hot chocolate?

LLM

Tools

get_column_
max

get_column_
mean

filter_rows

get_column_
min

get_column_
median

sum_rows

date	price	coffee_name	Size
2024-01-28	3.87	Hot Chocolate	M
2024-03-01	2.89	Cappuccino	S
2024-03-04	3.87	Latte	M
...	...	...	...
2025-03-23	4.57	Latte	L

How many unique transactions last week?

get_unique_
entries

What were the amounts of the last 5 transactions?

get_last_N_
values

...

...

- Brittle
- Inefficient
- Continuously dealing with edge cases

Planning: From Tools to Code

Planning with code execution

system
prompt

Write code to solve the user's query. Return your answer as python code delimited with `<execute_python>` and `</execute_python>` tags.

What were the amounts of the last 5 transactions?

LLM

```
<execute_python>
```

```
import pandas as pd
```

```
# Load CSV
```

```
df = pd.read_csv("transactions.csv")
```

```
# Ensure date column is parsed as datetime
```

```
df["date"] = pd.to_datetime(df["date"])
```

```
# Sort by date to get most recent transactions  
df_sorted = df.sort_values(by="date",  
                           ascending=False)
```

```
# Select the last 5 transactions
```

```
last_5 = df_sorted.head(5)
```

```
# Show just the price column (amounts)
```

```
print("Last 5 transaction amounts:")
```

```
print(last_5["price"].to_list())
```

```
</execute_python>
```

Planning: From Tools to Code

Planning with code execution

system prompt

Write code to solve the user's query. Return your answer as python code delimited with `<execute_python>` and `</execute_python>` tags.

How many unique transactions last week?

LLM

```
<execute_python>

import pandas as pd

# Read CSV and parse the "date" column as datetime
df = pd.read_csv("transactions.csv",
parse_dates=["date"])

# Define time window
today = pd.Timestamp.today()
week_ago = today - pd.Timedelta(days=7)

# Filter rows where date is within last week
last_week = df[df["date"].between(week_ago, today)]

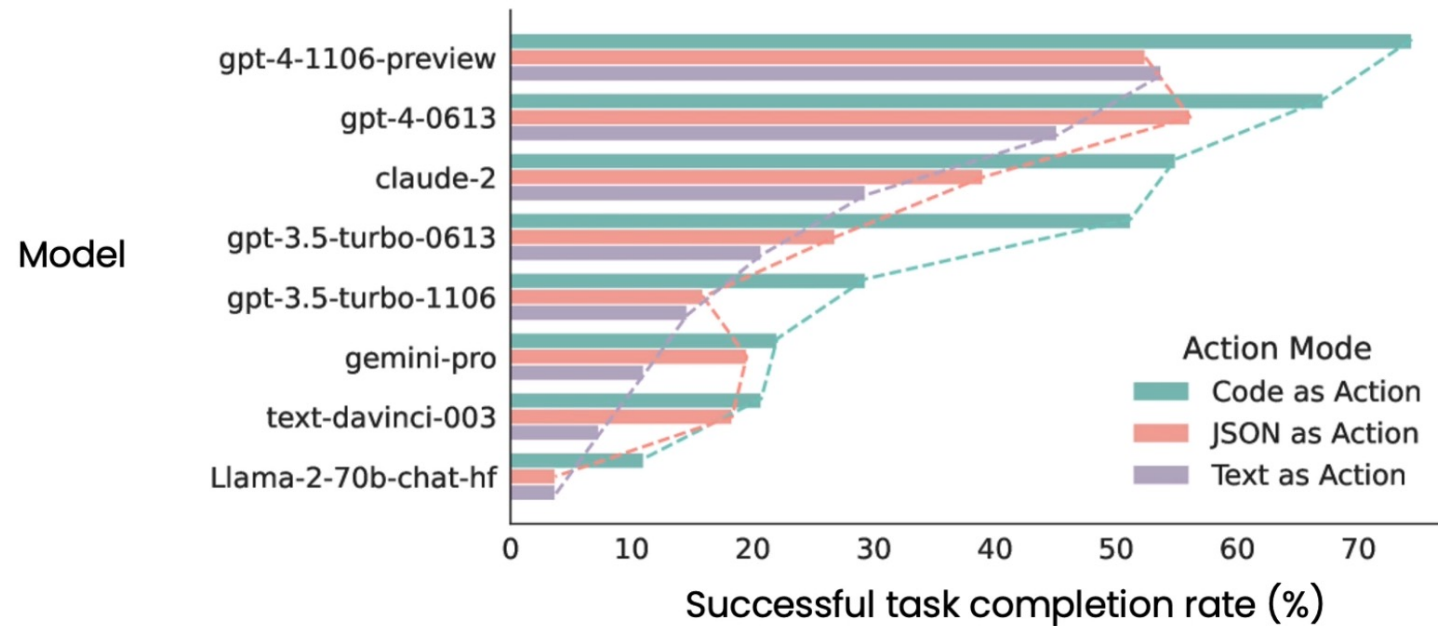
# Drop duplicate rows and count
print(last_week.drop_duplicates().shape[0])

</execute_python>
```

Homework: Can we instruct to plan with code that also uses tools?

Planning with Code

Planning with code improves performance



[Adapted from "Executable Code actions Elicit Better LLM Agents", Wang et al. 2024]

Multi-agentic Workflows

Some tasks require more than 1 person!

Task	Team
Create marketing assets	Researcher Graphic Designer Writer
Writing a research article	Researcher Statistician Lead writer Editor
Preparing a legal case	Associate Paralegal Investigator

Multi-agentic Example

Example: Marketing team

Researcher

Tasks

- Analyze market trends
- Research competitors

Tools

- Web search

researcher

Graphic designer

Tasks

- Create data visualizations
- Create artwork

Tools

- Image generation, manipulation
- Code execution for chart generation

graphic designer

Writer

Tasks

- Transform research into report text and marketing copy

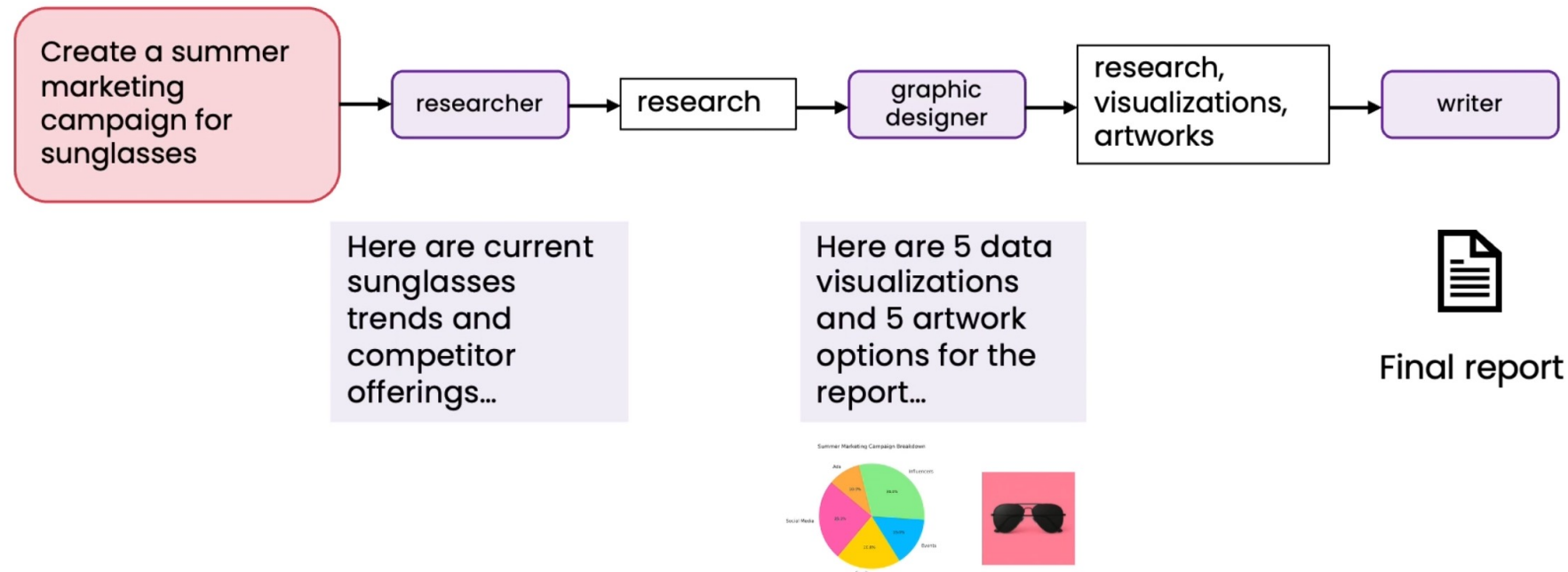
Tools

- (None)

writer

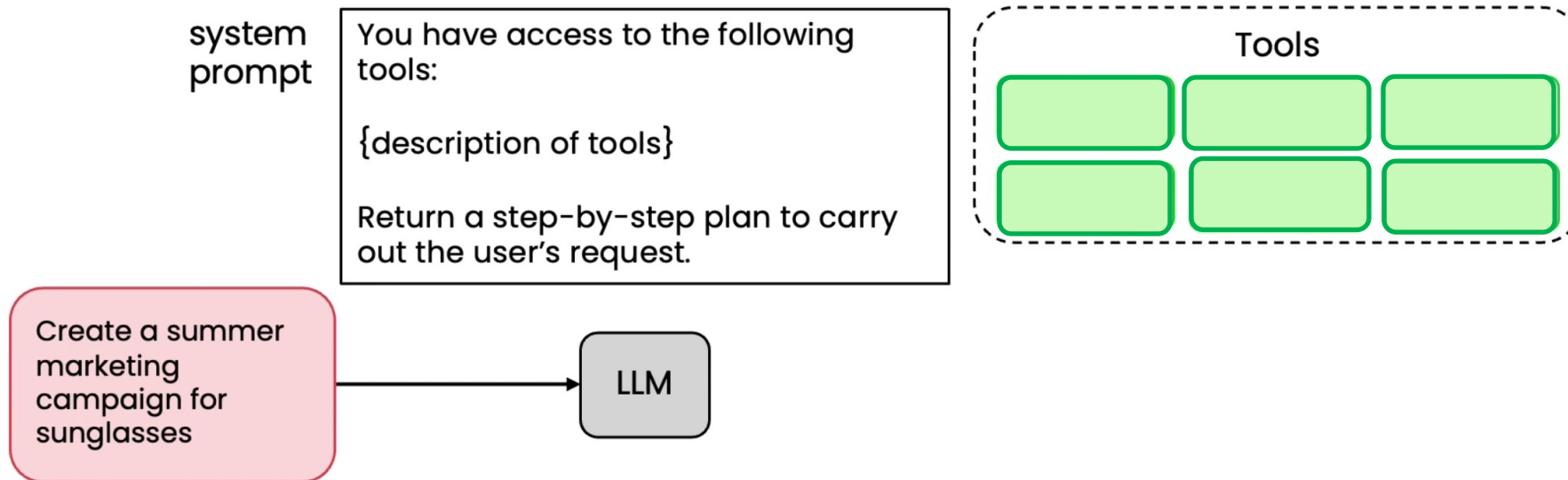
Multi-agentic Example

Example: Marketing team with linear plan



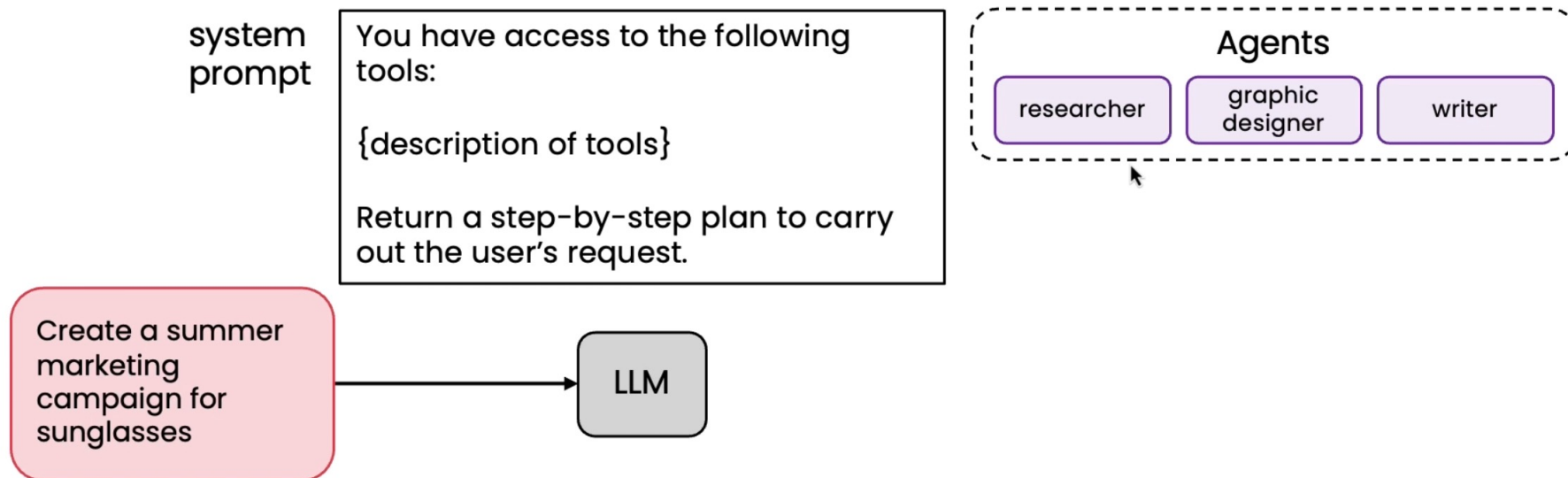
Non-Linear Planning with Multiple Agents

Example: Planning with multiple agents



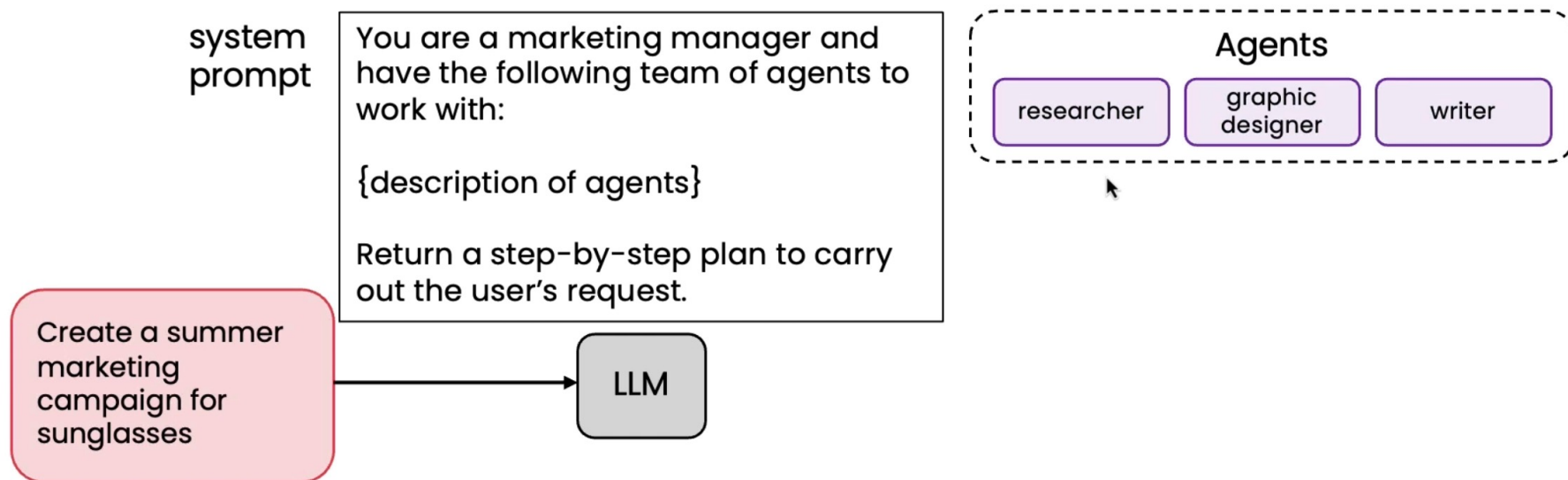
Non-Linear Planning with Multiple Agents

Example: Planning with multiple agents



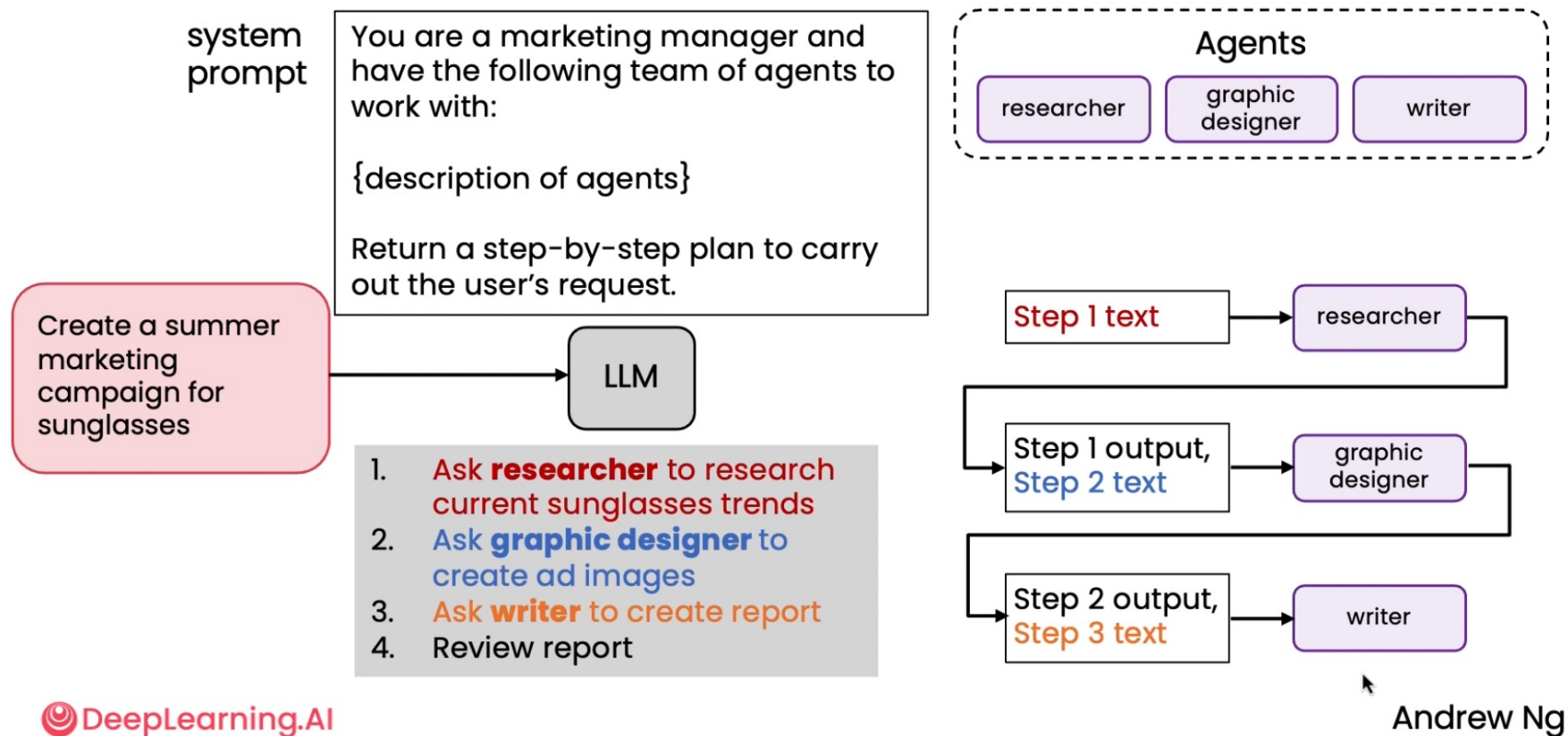
Non-Linear Planning with Multiple Agents

Example: Planning with multiple agents



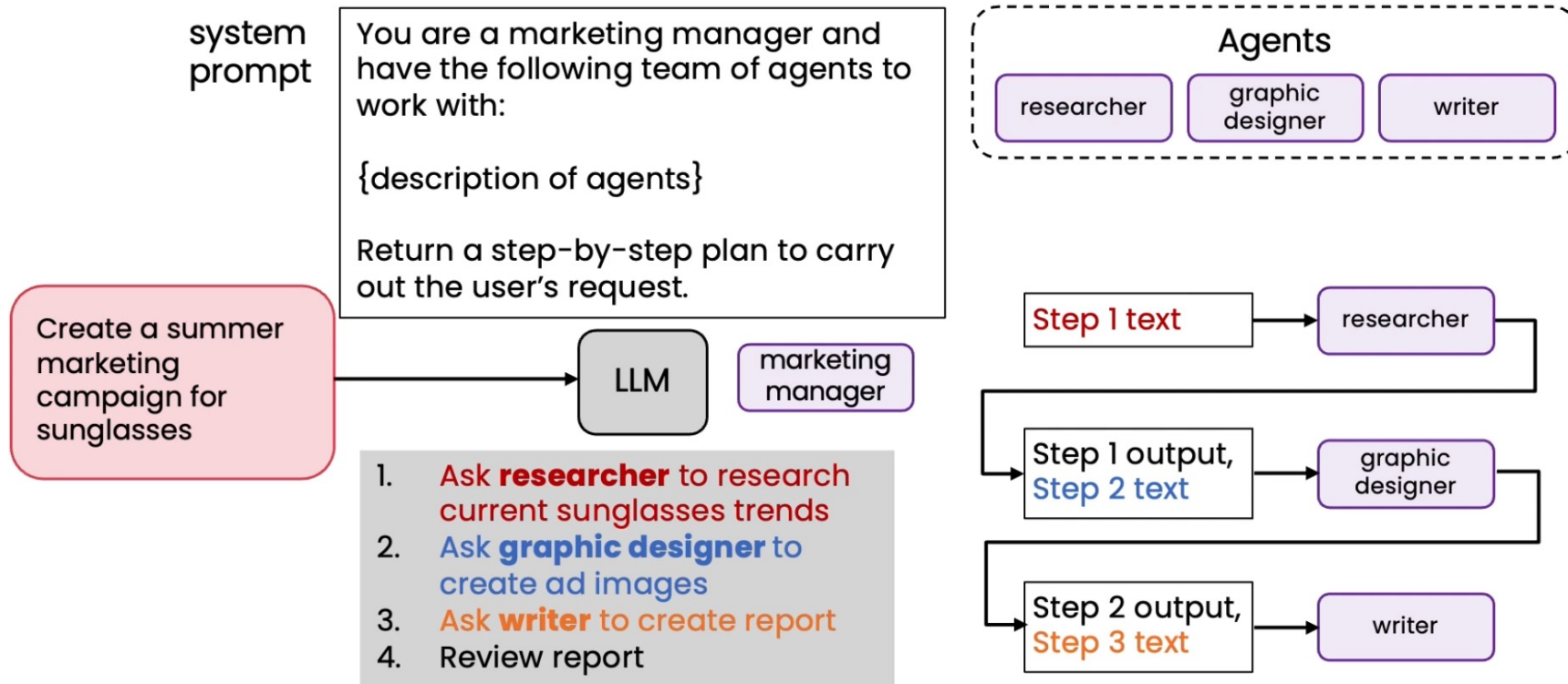
Non-Linear Planning with Multiple Agents

Example: Planning with multiple agents



Hierarchical Planning

Example: Planning with multiple agents

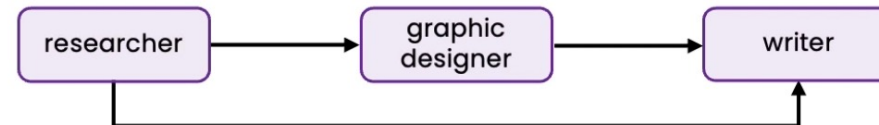


Linear Plan

Example: Marketing team with linear plan

Create a summary
marketing
campaign for
sunglasses

Communication Pattern



writer

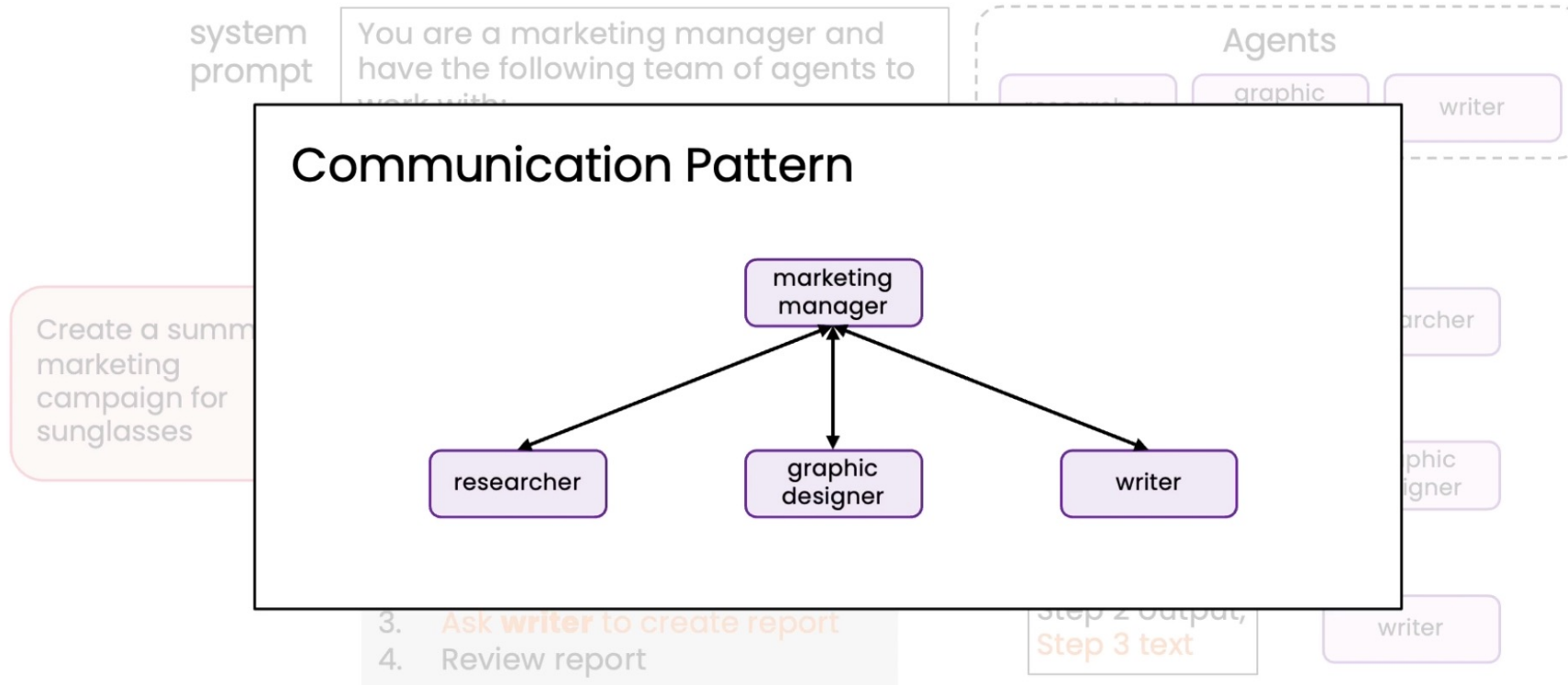


Final report



Hierarchical Plan

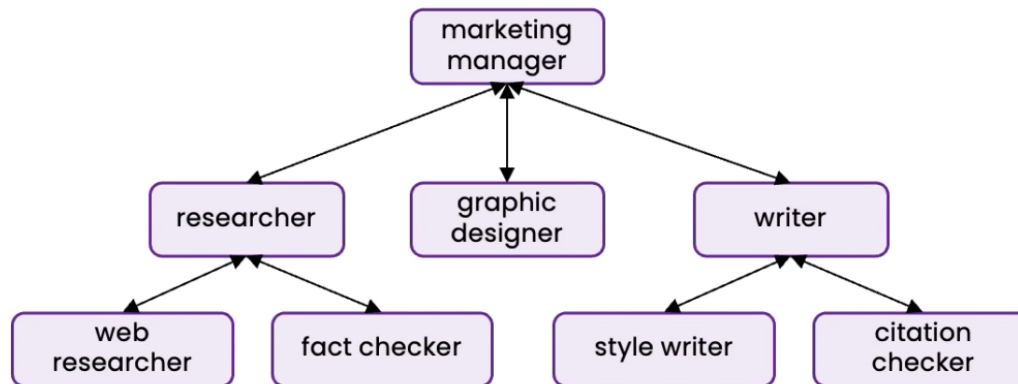
Example: Planning with multiple agents



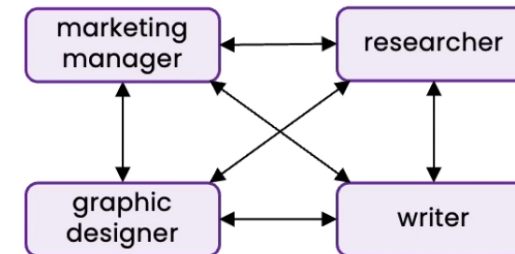
Deep Hierarchy and All-to-All

Other communication patterns

Deeper hierarchy



All-to-all



Tasks Suitable for Agentic AI

What tasks is agentic AI suited to?

Easier

Harder



Clear, step-by-step process

Steps not known ahead of time

Standard procedures to follow

Plan/solve as you go

Text assets only

Multimodal (sound, vision)

Clear Steps, One LLM Call

Example: Invoice processing workflow

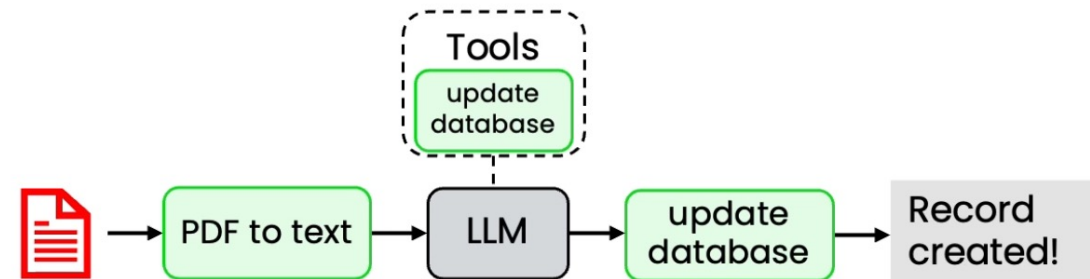
TechFlow Solutions LLC 890 Juniper Drive San Mateo, CA 94401 Phone: (415) 555-7890 Email: billing@techflowsol.com			
Due Date: August 20, 2025		Invoice Date: August 6, 2025	
Description	Qty	Unit Price	Line Total
Consulting - Systems Integration (hrs)	20	\$150.00	\$3,000.00
Total Due:			\$3,000.00

4 required fields:

Billers
Billers address
Amount due
Due date

Steps:

1. Identify required fields
2. Record in database



Clear Steps, Multiple LLM Calls

Example: Responding to customer email

From: Susan Jones
Subject: Wrong item shipped

I ordered a blue KitchenPro blender (Order #8847) but received a red toaster instead.

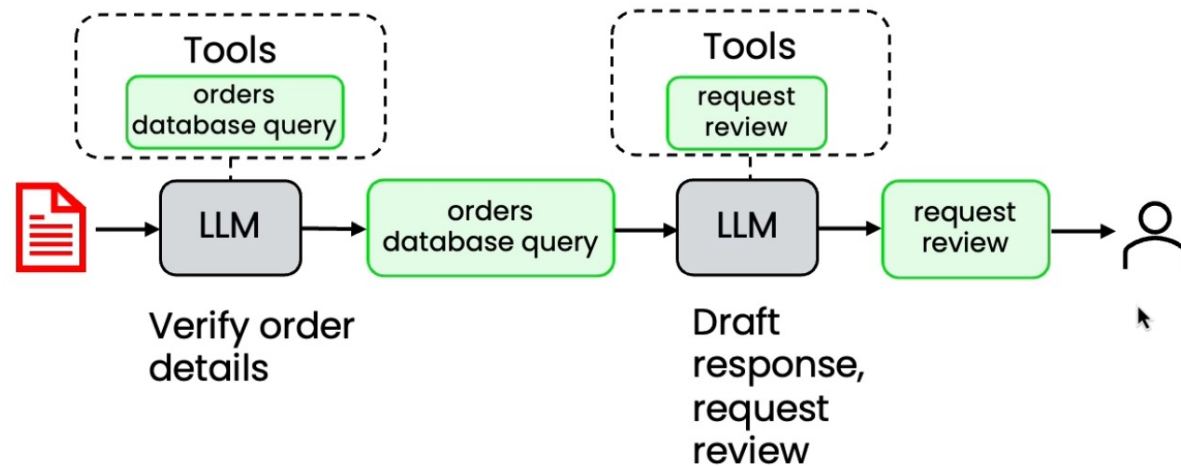
I need the blender for my daughter's birthday party this weekend. Can you help?

Susan

 DeepLearning.AI

Steps:

1. Extract key information
2. Find relevant customer records
3. Draft response for human review



Andrew Ng

Query-Dependent Steps

More challenging: Customer service agent

Do you have any black jeans or blue jeans?

1. Check inventory for black jeans
2. Check inventory for blue jeans
3. Respond to customer

I'd like to return the beach towel I bought

1. Verify customer purchase
2. Check return policies
3. If return allowed = "yes", then:
 - a. Issue return packing slip
 - b. Set database record to "return pending"

Required steps
not known ahead
of time

Complex Steps

Difficult: Visual computer use

The screenshot shows the United.com website interface. At the top, there's a search bar with fields for 'From', 'To', 'Dates' (set to Jul 31), 'Travelers' (1 Adult), and 'Show price in: Money'. An 'Update' button and an 'Advanced Search' link are also present. Below this is the 'Traveler Info' section, which includes a sign-in prompt for MileagePlus and a form for 'Traveler 1:'. The form has fields for 'First name (required)', 'Last name (required)', 'Middle name/initial (if shown on ID)', 'Date of birth (required)' (with month, day, and year dropdowns), 'Gender (required)' (with a radio button), and 'Suffix'. There's also a 'Frequent flyer program' section. To the right, the 'ONEWAY (1 TRAVELER)' section shows flight details: 'San Francisco SFO to Washington DCA', 'Jul 31 • 1:23 pm to 9:27 pm • Nonstop', and a price breakdown: 'Fare \$449.47', '1 adult 18+ \$449.47/person', 'Taxes and fees \$49.01', and a total of '\$498.48'. A chatbot overlay at the bottom center says: 'I am waiting for the content to load on the United website. Once it's fully loaded, I can proceed with the next steps based on what information or actions are required.'

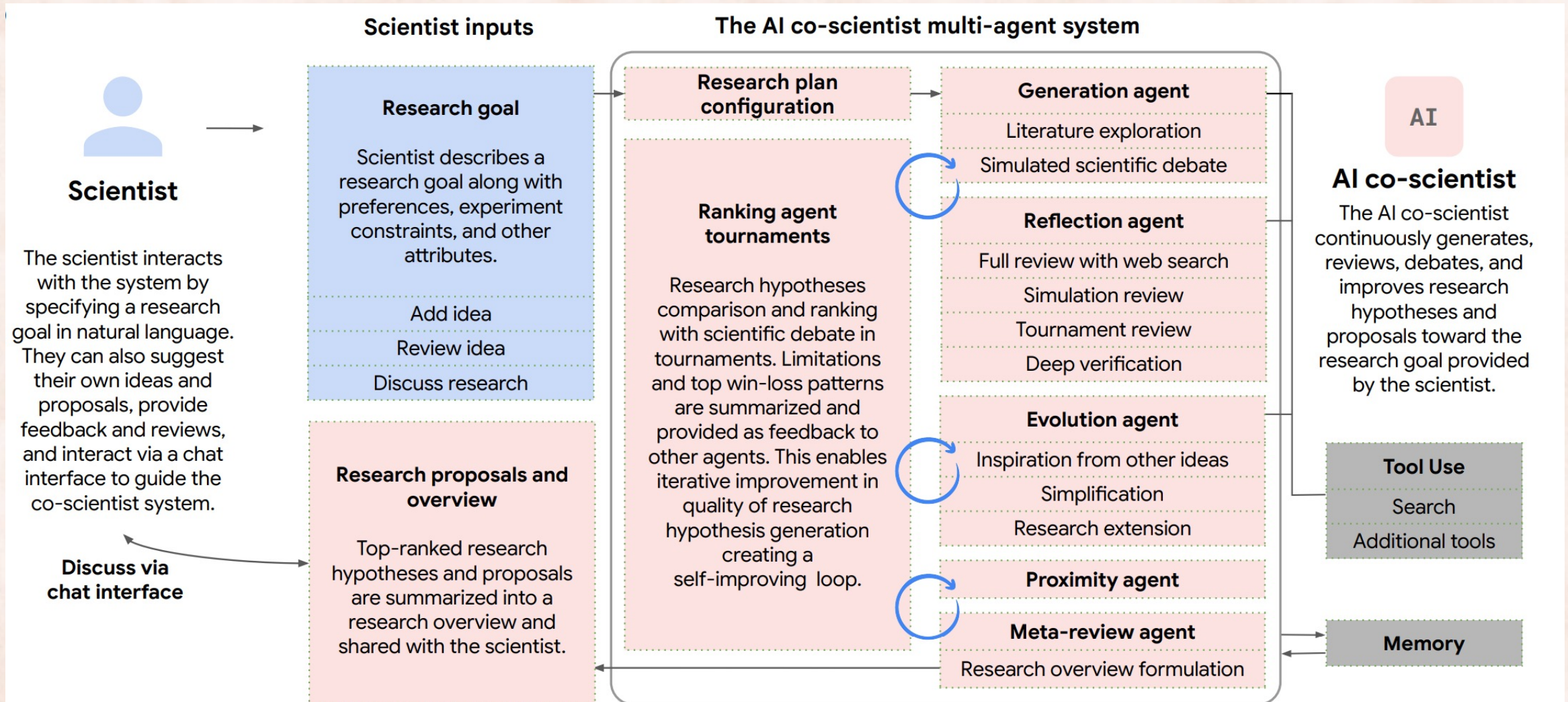
ChatGPT Agent Mode
(OpenAI)

Towards an AI co-scientist

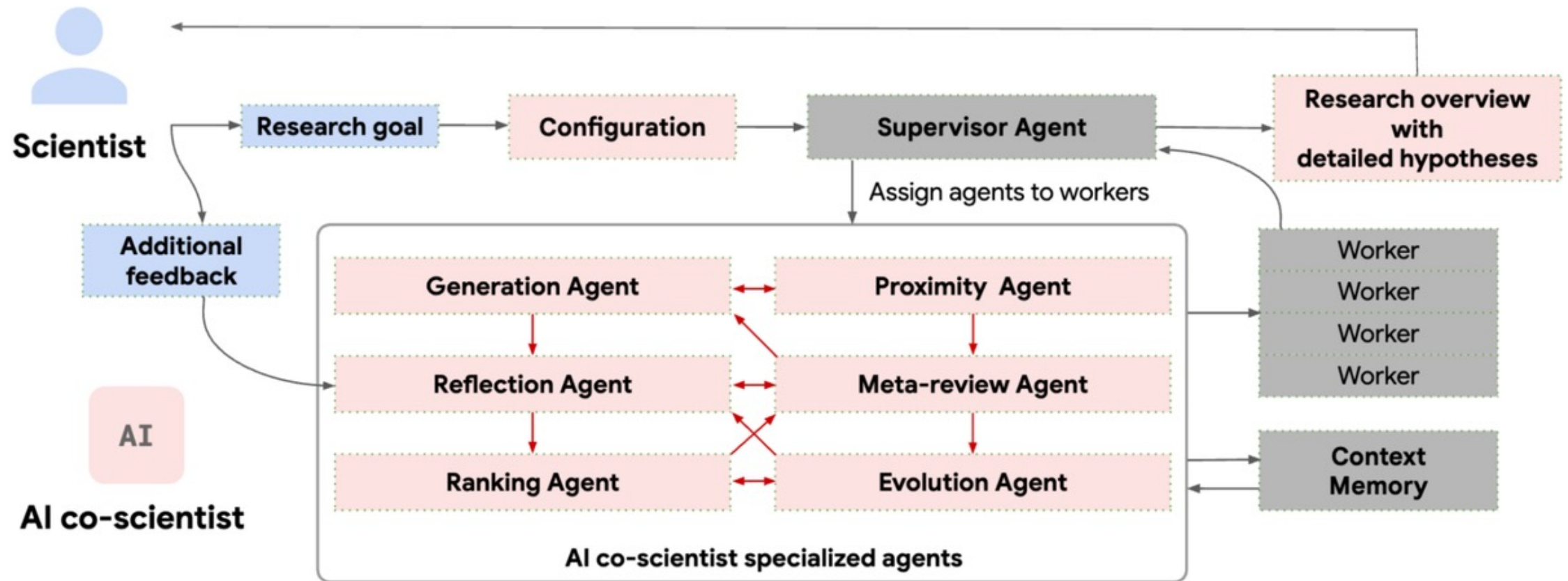
- A **multi-agent system** intended to help uncover new knowledge and to formulate demonstrably novel research hypotheses and proposals.
 - Building upon prior evidence and aligned to scientist-provided research objectives and guidance.
- Design incorporates a **generate**, **debate**, and **evolve** approach to hypothesis generation, inspired by the scientific method.
 - (1) **Multi-agent architecture** with an **asynchronous task execution framework** for flexible compute scaling;
 - (2) A **tournament evolution** process for **self-improving hypotheses generation**.
- Development and validation in three biomedical areas:
 - drug repurposing.
 - novel target discovery.
 - explaining mechanisms of bacterial evolution and anti-microbial resistance.

<https://research.google/blog/accelerating-scientific-breakthroughs-with-an-ai-co-scientist>

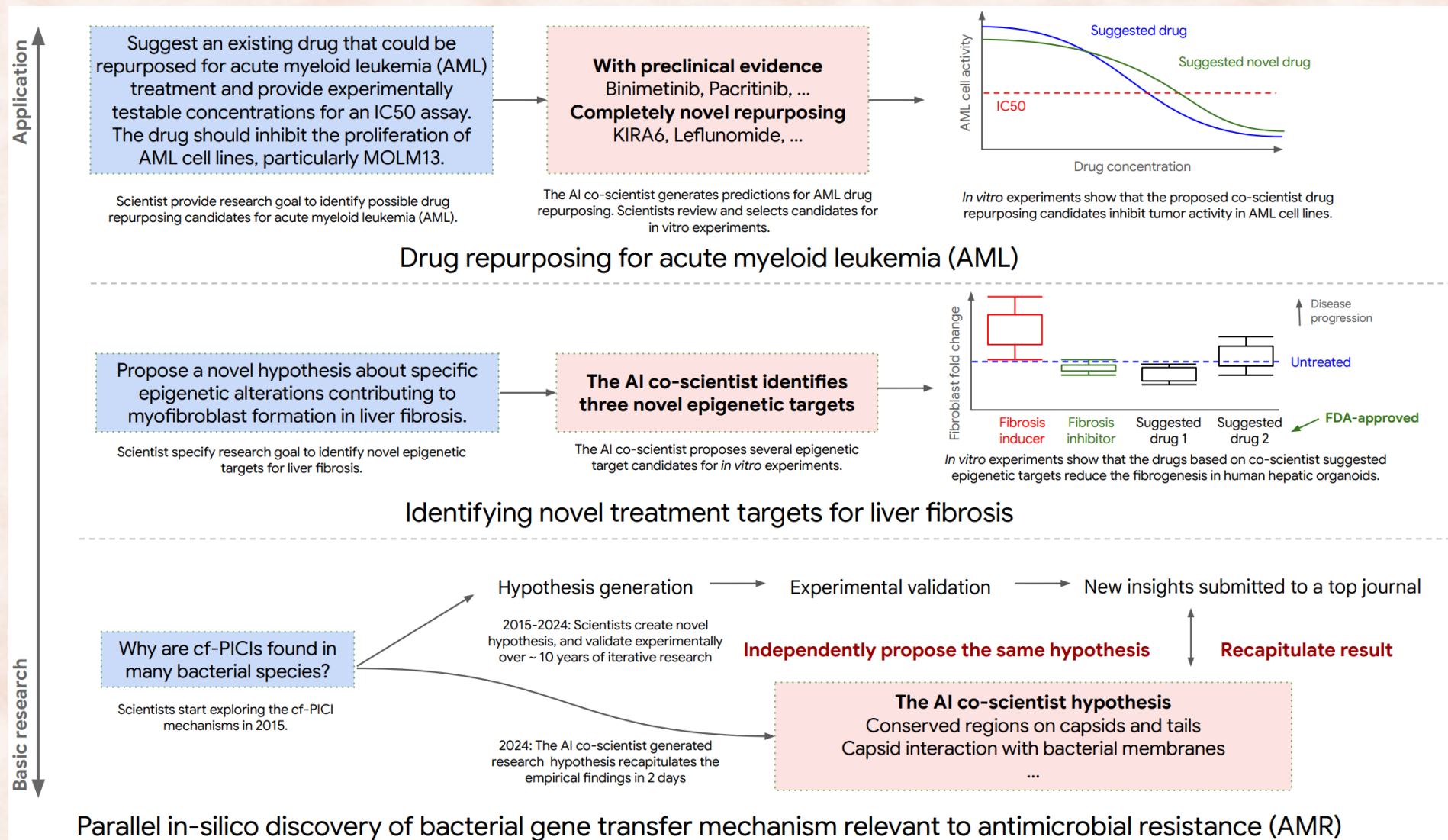
The AI Co-Scientist System Design



Agentic Workflow



Basic Research and Applications



Supplemental Materials

- The [Agentic AI](#) course taught by Andrew Ng on the [DeepLearning.AI](#) platform.
 - Video lectures.
 - Jupyter notebooks.
- Google's [Agent Development Kit](#):
 - [Agents](#) in ADK.
 - [Multi-Agent Systems](#) in ADK.
 - Using MCP with ADK:
 - <https://google.github.io/adk-docs/mcp>
 - <https://google.github.io/adk-docs/tools/mcp-tools>
- OpenAI's [AgentKit](#).