

A red vertical bar on the left side of the slide contains two sets of white, concentric, rounded rectangular lines. The top set consists of five lines, and the bottom set consists of seven lines. The lines are spaced out and extend from the left edge towards the center of the slide.

# Agentic AI

---

M1: Introduction to agentic workflows

The left side of the slide features a solid red vertical bar. Overlaid on this bar are two sets of white, concentric, rounded rectangular lines. The upper set consists of four lines, and the lower set consists of eight lines, both sets appearing to flow from the left edge towards the center.

# Introduction to Agentic Workflows

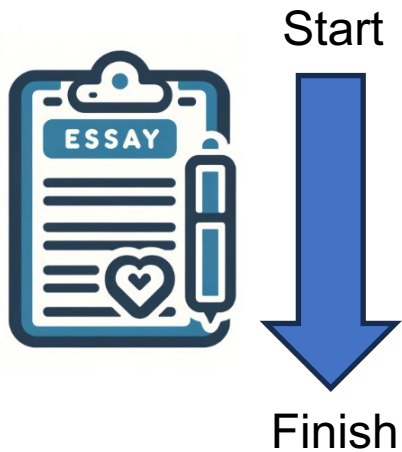
---

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

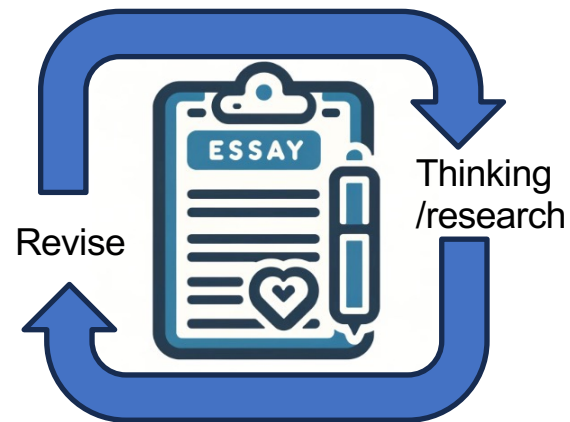
Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

Revise your draft.

....



# Agentic AI workflows

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

Essay-writing example:

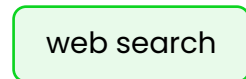
Write an essay outline on topic X



Do you need any web research?



+



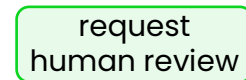
Write a first draft.



Consider what parts need revision or more research.



+



Revise your draft.





# Introduction to Agentic Workflows

---

Degrees of Agenticness



Andrew Ng  
AI Fund

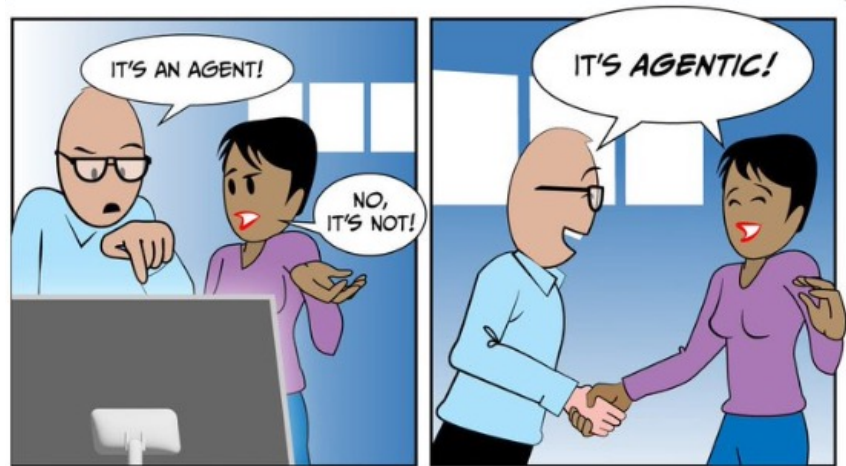
Sequoia Ascent, March 2024



....

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.

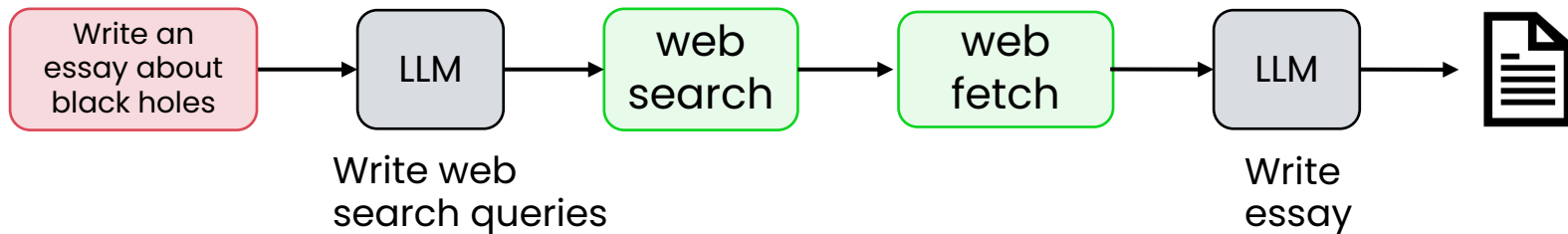
....



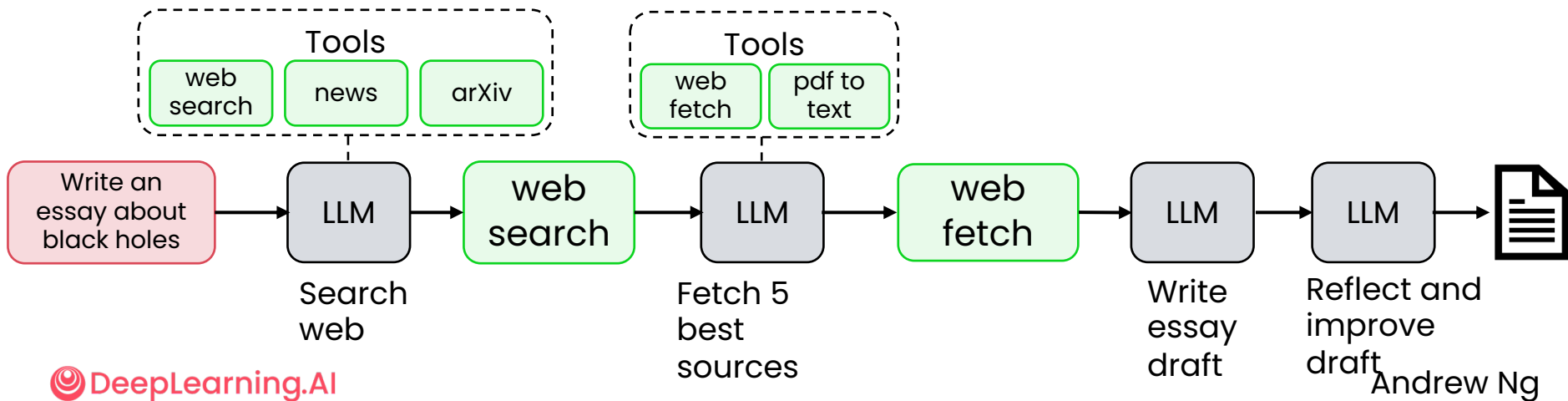
X (twitter) post , June 2024

# Degrees of autonomy

Less autonomous



More autonomous



# Degrees of “agenticness”

Agentic AI can be less or more autonomous

Less  
autonomous

Semi-  
autonomous

Highly  
autonomous



- All steps predetermined
- All tool use hard coded
- Autonomy is in text generation

- Agent can make some decisions, choose tools
- All tools predefined

- Agent makes many decisions autonomously
- Can create new tools on the fly



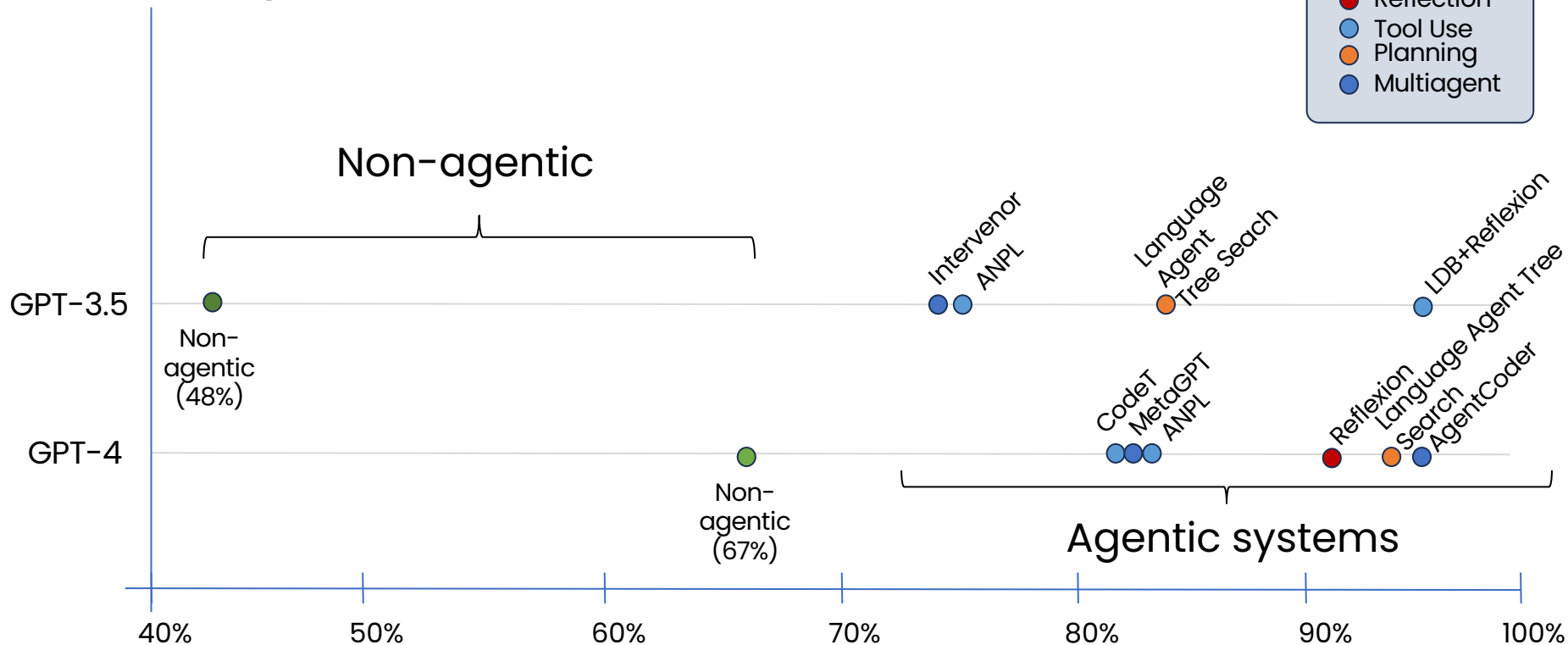
# Introduction to Agentic Workflows

---

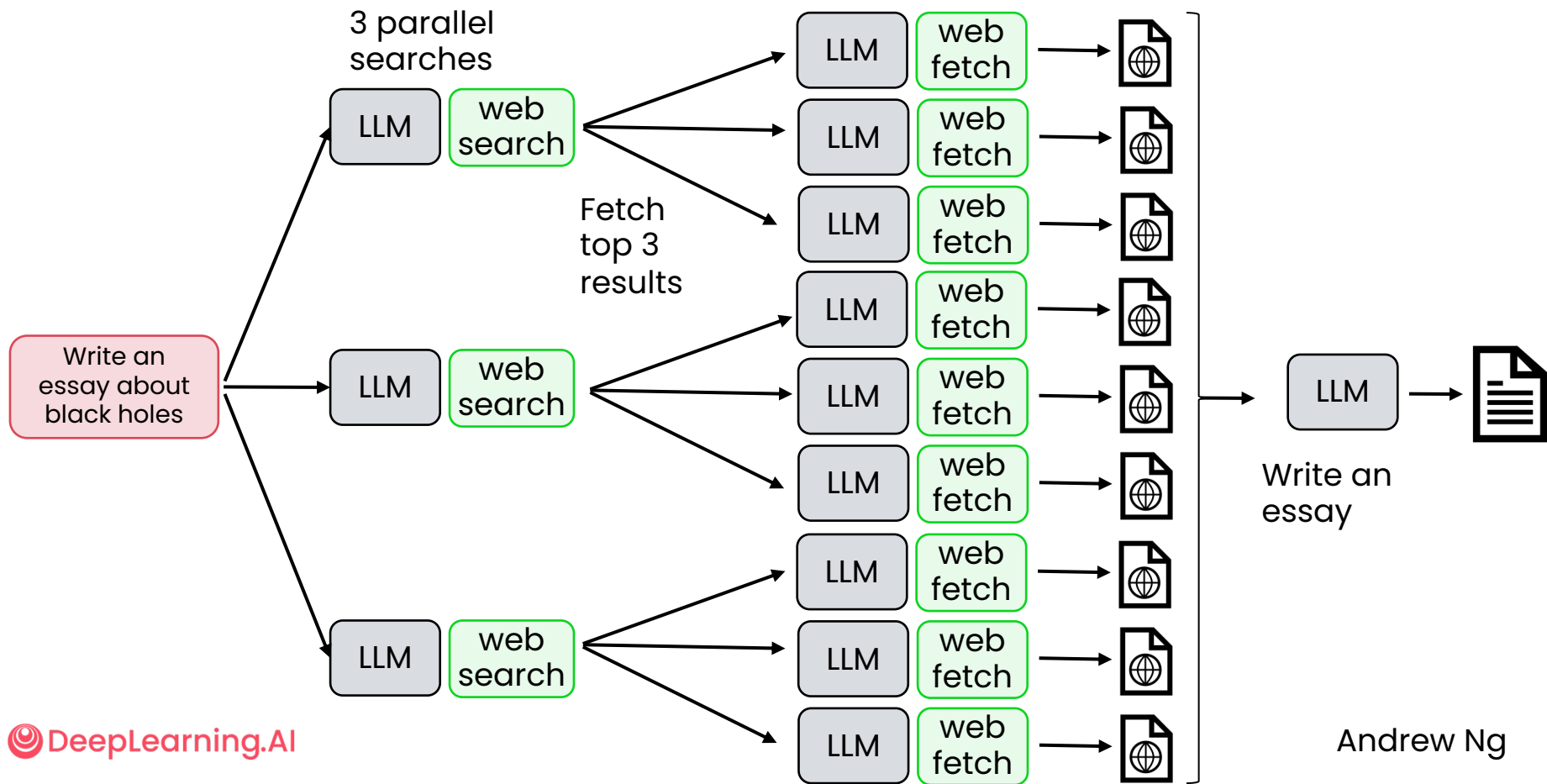
Benefits of agentic AI

# Coding benchmark (HumanEval)

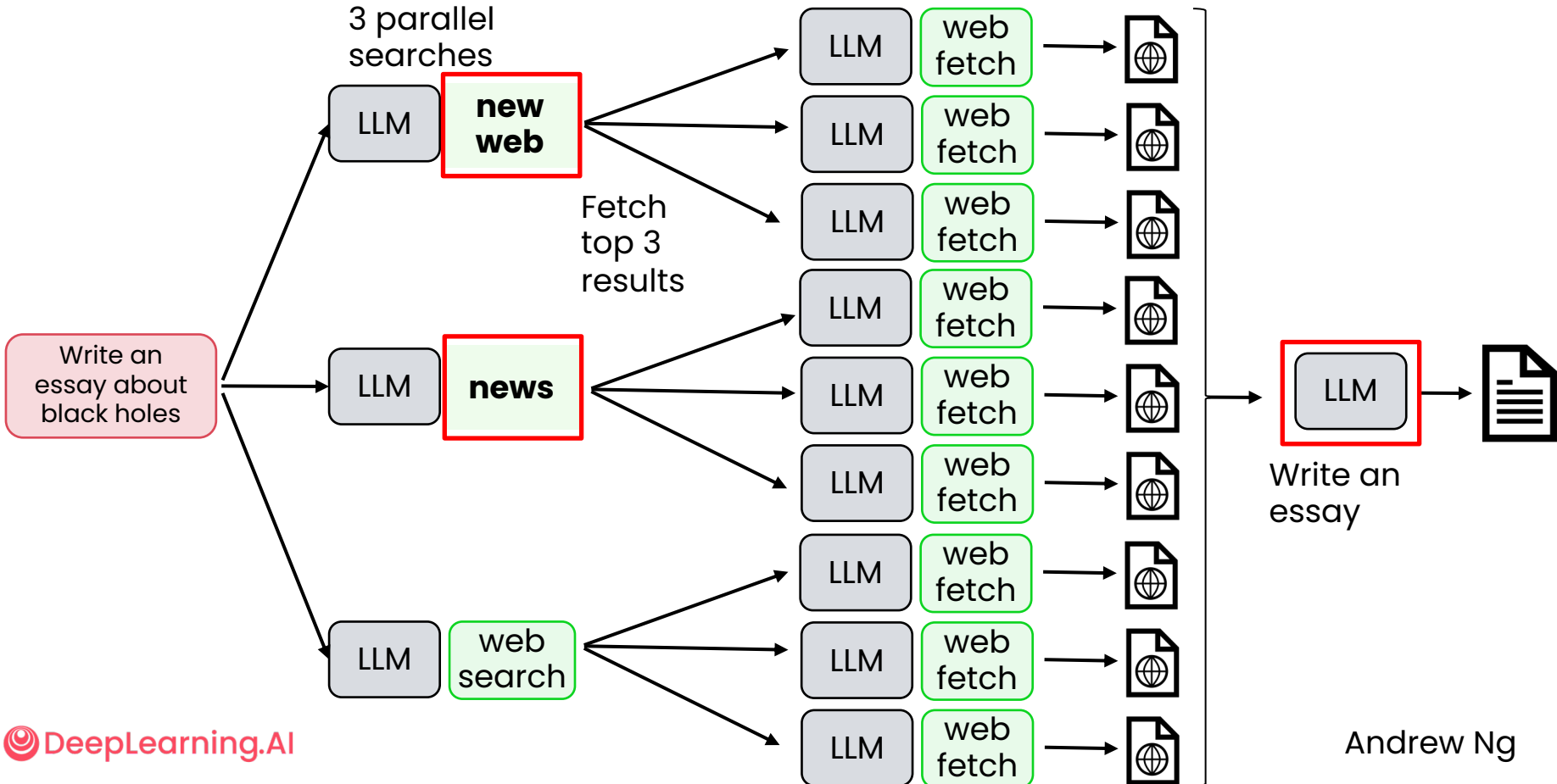
- Non-agentic
- Reflection
- Tool Use
- Planning
- Multiagent



# Parallelization for speed



# Modular: Add/swap out components







# Introduction to Agentic Workflows

---

Agentic AI applications

# 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**

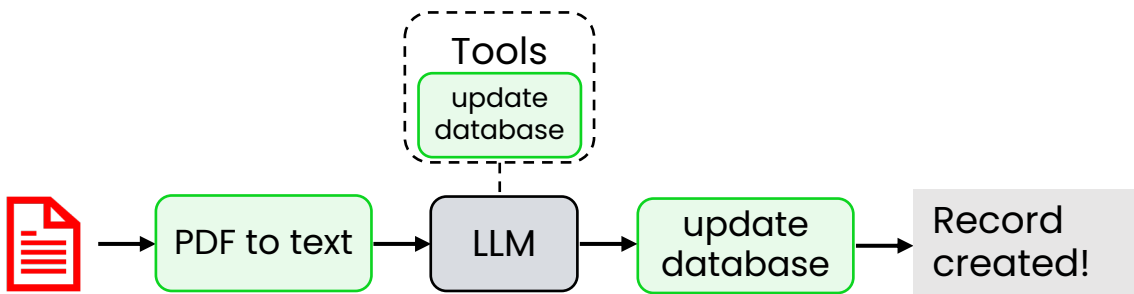
Description	Qty	Unit Price	Line Total
Consulting - Systems Integration (hrs)	20	\$150.00	\$3,000.00
<b>Total Due:</b>			<b>\$3,000.00</b>

## 4 required fields:

Billor  
Billor address  
Amount due  
Due date

## Steps:

1. Identify required fields
2. Record in database



# Example: Responding to customer email

From: Susan Jones  
Subject: Wrong item shipped

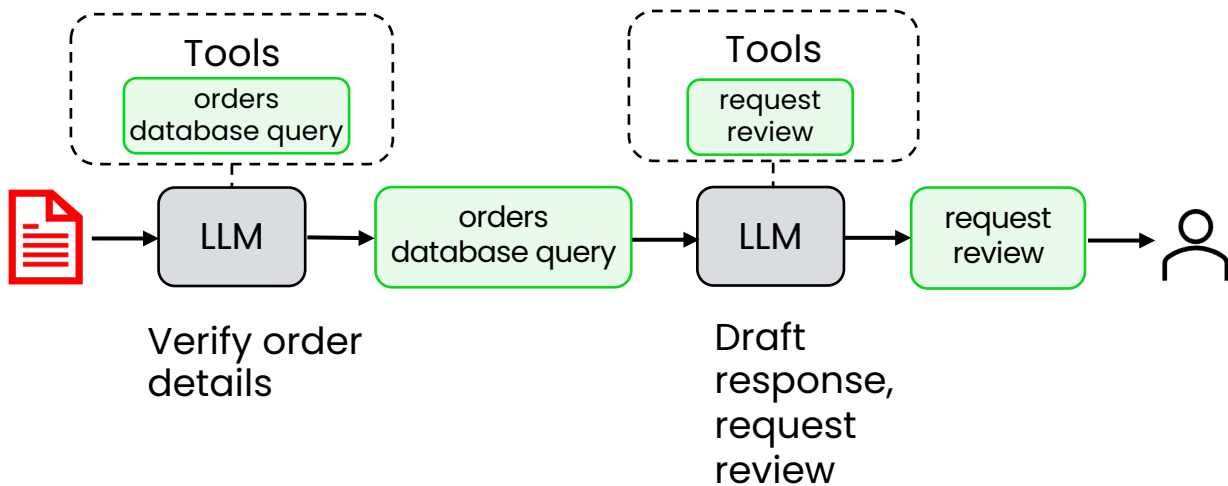
I ordered a blue KitchenPro blender (Order #8847) but received a red toaster instead.

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

Susan

Steps:

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



# More challenging: Customer service agent

Do you have any black jeans or blue jeans?

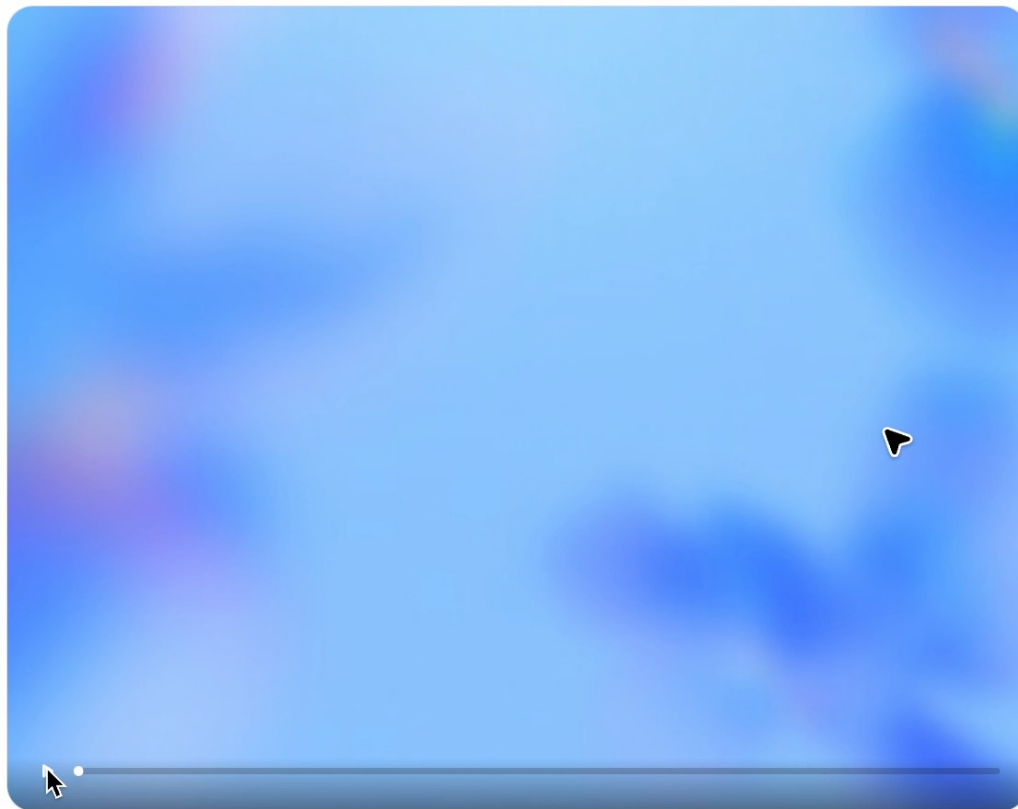
1. Check inventory for black jeans
2. Check inventory for blue jeans
3. Respond to customer

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

# Difficult: Visual computer use



ChatGPT Agent Mode  
(OpenAI)

# 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)

The left side of the slide features a solid red vertical bar. Overlaid on this bar are two sets of white, concentric, rounded rectangular lines. The upper set consists of four lines, and the lower set consists of eight lines, both sets appearing to flow from the left edge towards the center.

# Introduction to Agentic Workflows

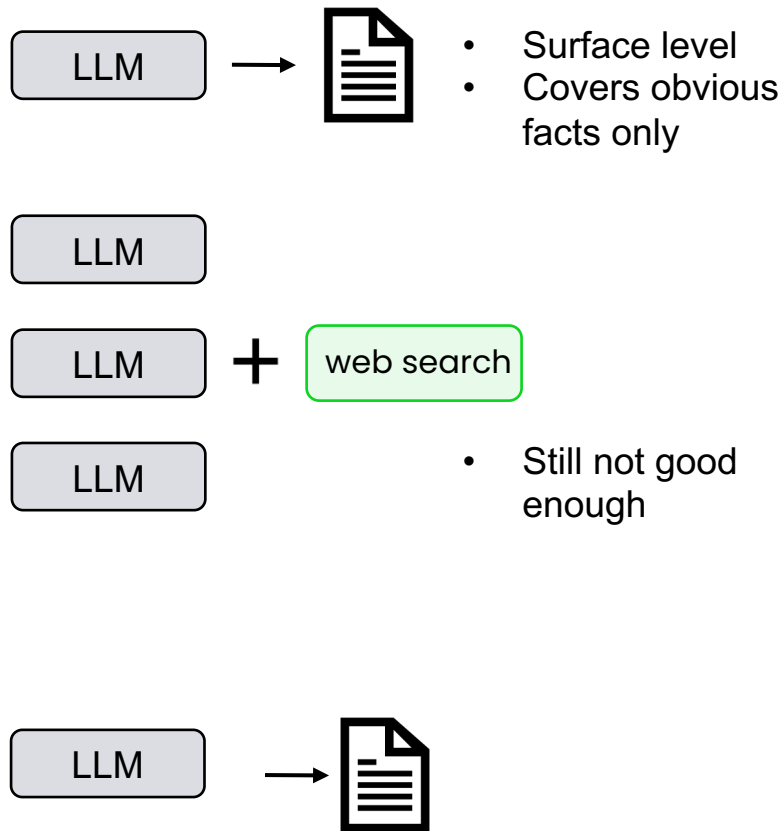
---

Task decomposition:  
Identifying the steps in  
a workflow

# Example: Writing an Essay

Write an essay on topic X

1. Write an essay outline on topic X
2. Search web
3. ~~Write the essay~~ Write a first draft
4. Consider what parts need revision
5. Revise your draft



# Recap: Breaking down essay writing into steps

Direct generation:

1. Write an essay on topic X

Not good enough...

3-step workflow:

1. Write an essay outline on topic X
2. Search web
3. Write the essay

Still not good enough...

5-step workflow:

1. Write an essay outline on topic X
2. Search web
3. Write the first draft
4. Consider what parts need revision
5. Revise your draft

# Example: Responding to customer email

From: sjones9@email.com  
Subject: Wrong item shipped

Hi, 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 Jones

1. Extract key information

LLM

2. Find relevant customer records

LLM

+

orders database query

3. Write and send response

LLM

+

send email

# Example: Extracting information from invoice

## 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
<b>Total Due:</b>			<b>\$3,000.00</b>

1. Find required information

LLM

LLM

+



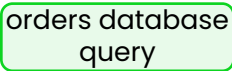

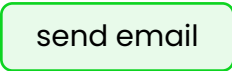
update  
database

# What building blocks do you have?

Building block	Examples	Use cases
Models	LLMs	Text generation, tool use, information extraction
	Other AI models	PDF-to-text, text-to-speech, image analysis
Tools	API	Web search, get real-time data, send email, check calendar,....
	Information retrieval	Databases, Retrieval Augmented Generation (RAG)
	Code execution	Basic calculator, data analysis

# Tips for task decomposition

Example: Responding to email

1. Extract key information 
2. Find relevant customer records  + 
3. Write and send response  + 

Questions to ask:

- Can LLM, tool, or both complete each step?
- If “no”, consider breaking down into further steps, or adding human-in-the-loop

A red vertical bar on the left side of the slide contains two sets of white, concentric, rounded rectangular lines. The top set consists of four lines, and the bottom set consists of eight lines, both sets appearing to fade out towards the right.

# Introduction to Agentic Workflows

---

Evaluating Agentic AI  
(evals)

# Look for low-quality outputs

From: sjones9@email.com  
Subject: Wrong item shipped

Hi, 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 Jones

1. Extract key information

LLM

2. Find relevant customer records

LLM

+

orders database query

3. Write and send response

LLM

+

send email

"I'm glad you shopped with us – we're much better than our competitor CompCo!"

"Sure, we'll issue a refund. Unlike RivalCo, we make returns easy."

# Add an evaluation to track the error

Customer inquiry

We're better than CompCo...

LLM response

```
if (competitor in response):  
    num_competitor_mentions += 1
```

List of competitors:

CompCo

RivalCo

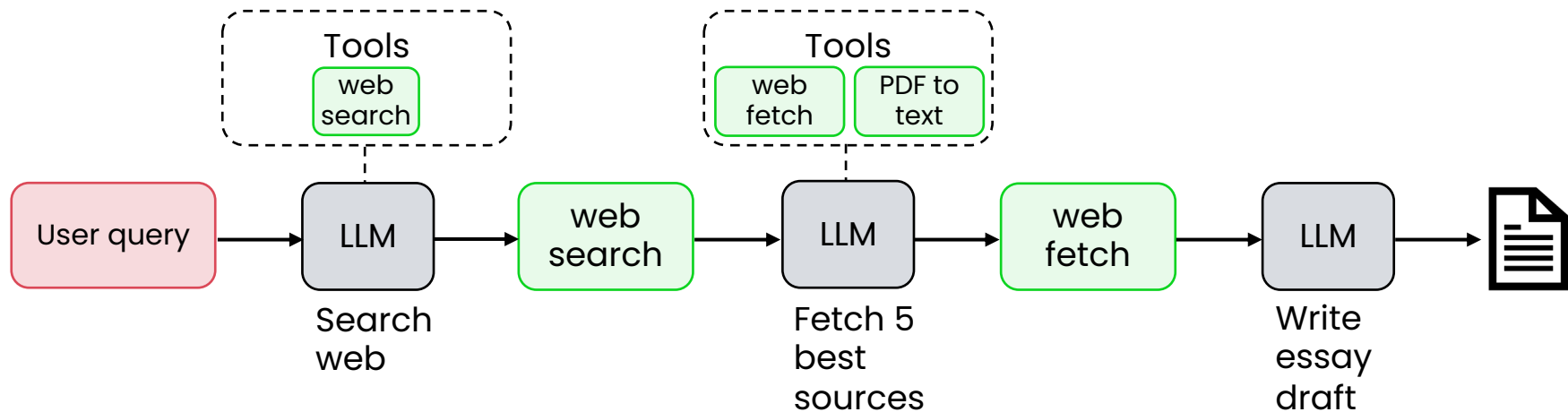
The Other Co

....

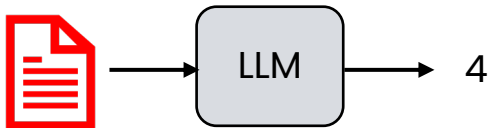
Objective error:

Use code to check for occurrences

# Using LLM as a judge



Assign the following essay a quality score between 1 and 5, where 5 is the best: {essay}



Example prompt	LLM-assigned score
Black holes	3
Robotic harvesting	4

# Evaluating Agentic AI

- Can evaluate using code (objective evals), or LLM-as-judge (subjective evals)
- Two types of evals: End-to-end and component-level
- Examine traces to perform error analysis
- Much more on evals and error analysis in Module 4!



# Introduction to Agentic Workflows

---

Agentic design patterns

# Agentic Design Patterns

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

# 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



Please write code for {task}

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

```
def do_task_v2(x):
```

```
def do_task_v3(x):
```



Coder  
Agent  
(LLM)

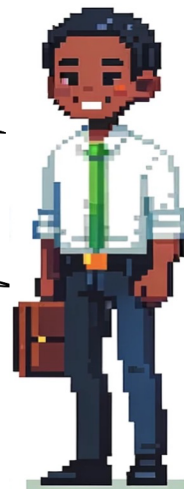
Here's code intended for {task}:

```
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 ...



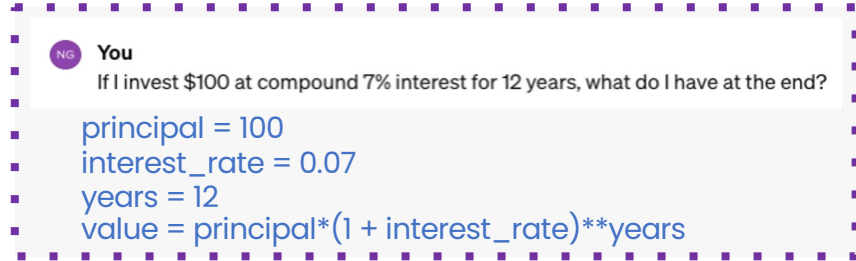
Critic  
Agent  
(LLM)

# Tool use

## 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

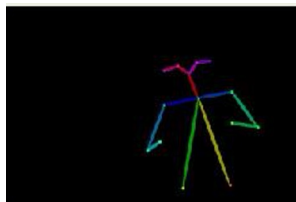
- Image generation
- Image captioning
- OCR

# Planning

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



*"A girl is sitting on a bed reading a book."*



Pose Determination

openpose model

Pose-to-Image

google/vit model

Image-to-text

vit-gpt2 model

Text-to-Speech

fastspeech model

[Example adapted from "HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face", Shen et al. (2023)]

# Multi-agentic workflows



## Multiagent Debate

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

[From "Improving Factuality and Reasoning in Language Models through Multiagent Debate", Du et al., 2023]



DeepLearning.AI

# End of M1

---



# Agentic AI

---

## M2: Reflection Design Pattern

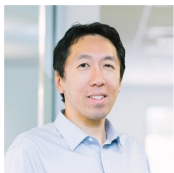


# Reflection Design Pattern

---

Reflection to improve  
outputs of a task

# Reflection - humans



Write an  
email



Didn't sign my  
name

Hey Tommy,  
I'll be in New York  
next month, let me  
know if you'll be  
fre for dinner one  
night.

Email V1

Specific  
dates

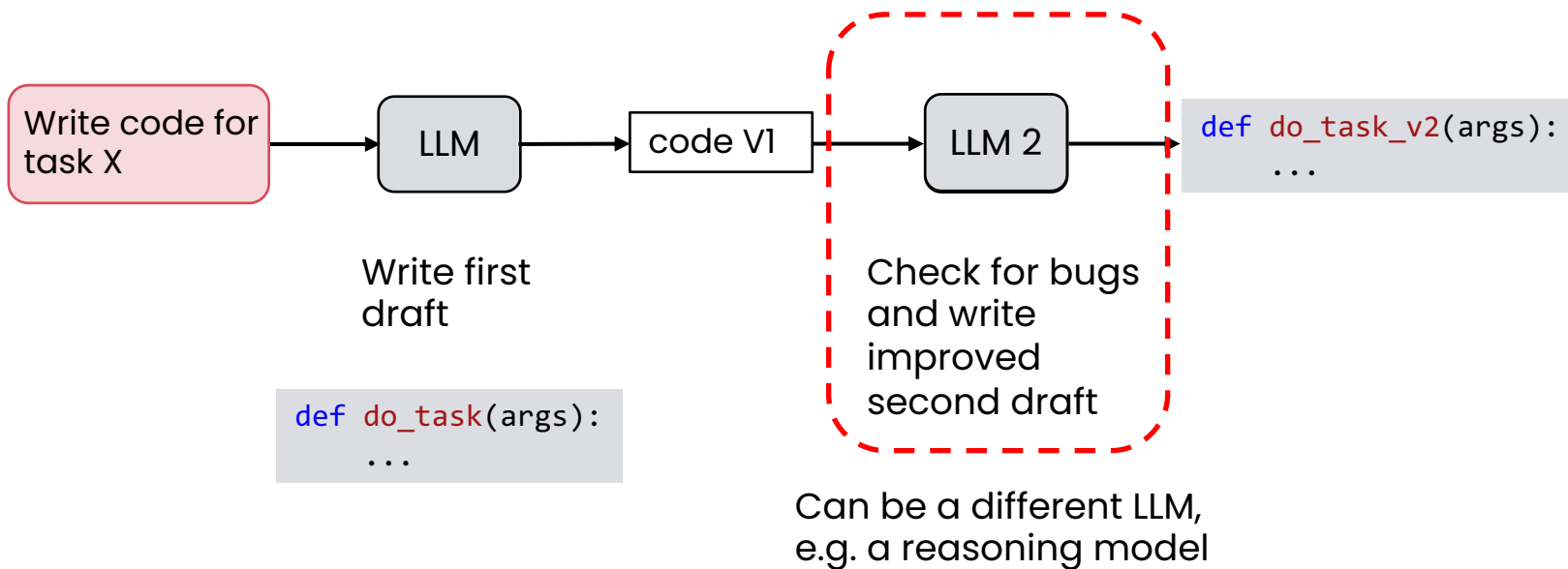
Typo

Hey Tommy,  
I'll be in New York  
next month from  
the 5<sup>th</sup>-7<sup>th</sup>. Let me  
know if you'll be  
free for dinner one  
night.

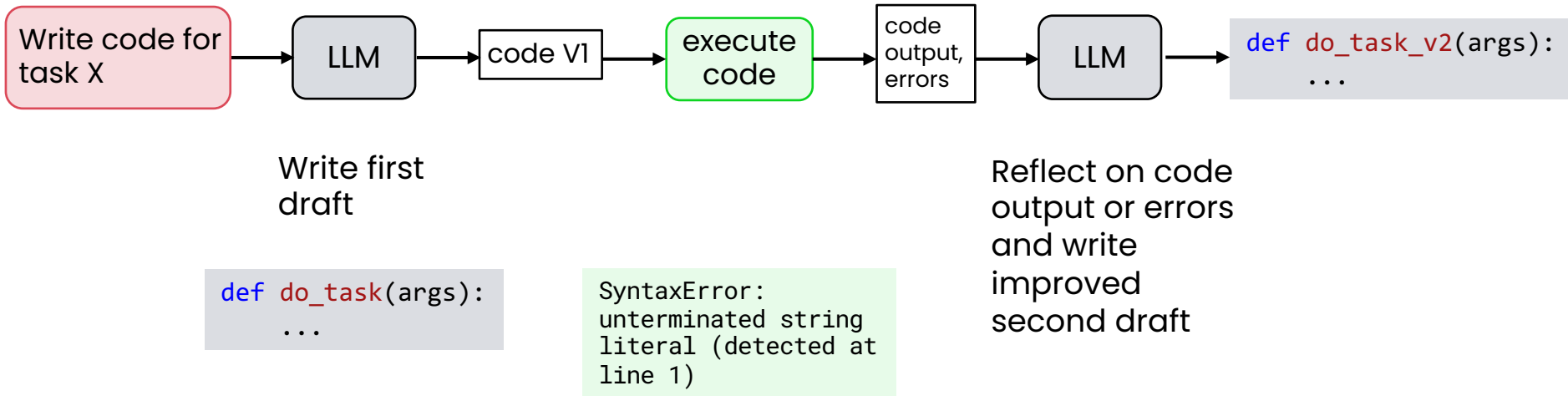
Andrew

Email V2

# Reflection to improve code



# Reflection with external feedback



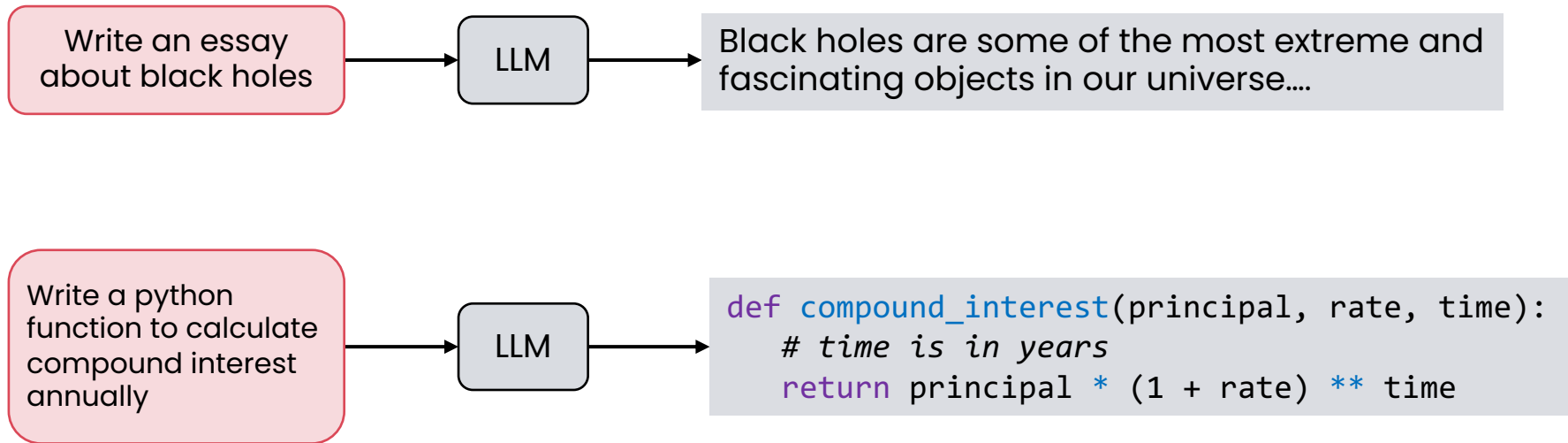


# Reflection Design Pattern

---

Why not just direct  
generation?

# Direct generation



# Zero, one, and few-shot prompting

Convert to  
MM/DD/YYYY format

Input:  
{input\_date}

Convert to  
MM/DD/YYYY format

Input: Jan 1<sup>st</sup>, 2025  
Output: 01/01/2025

Input:  
{input\_date}

Convert to  
MM/DD/YYYY format

Input: Jan 1<sup>st</sup>, 2025  
Output: 01/01/2025

Input: 21<sup>st</sup> June, 2025  
Output: 06/21/2025

Input:  
{input\_date}

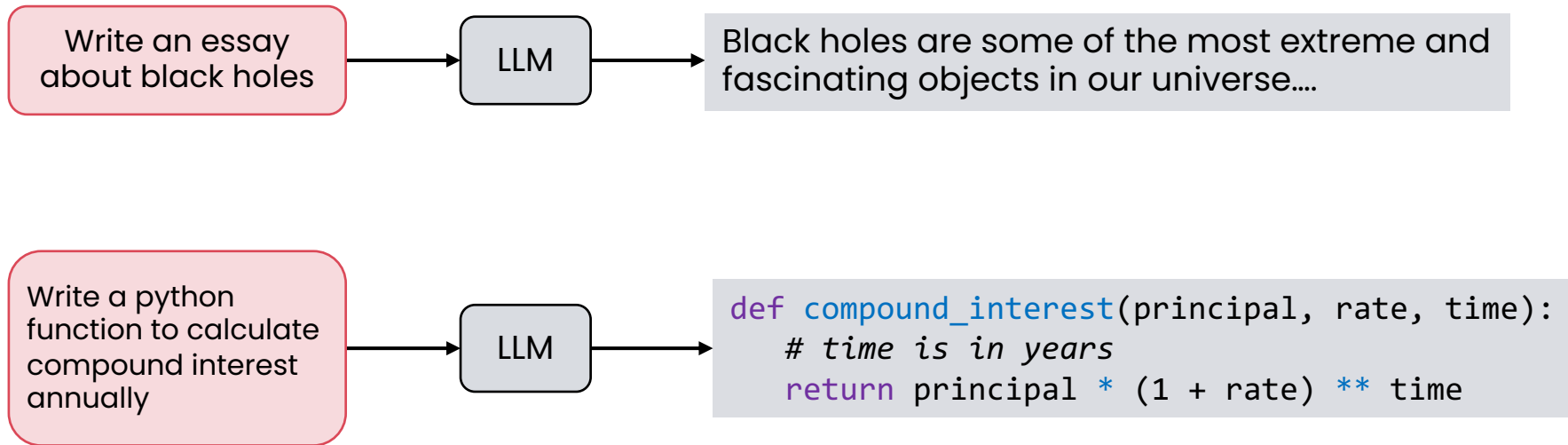
Zero-shot (no examples)

One-shot (single example)

Two-shot (two examples)  
Few-shot (multiple examples)

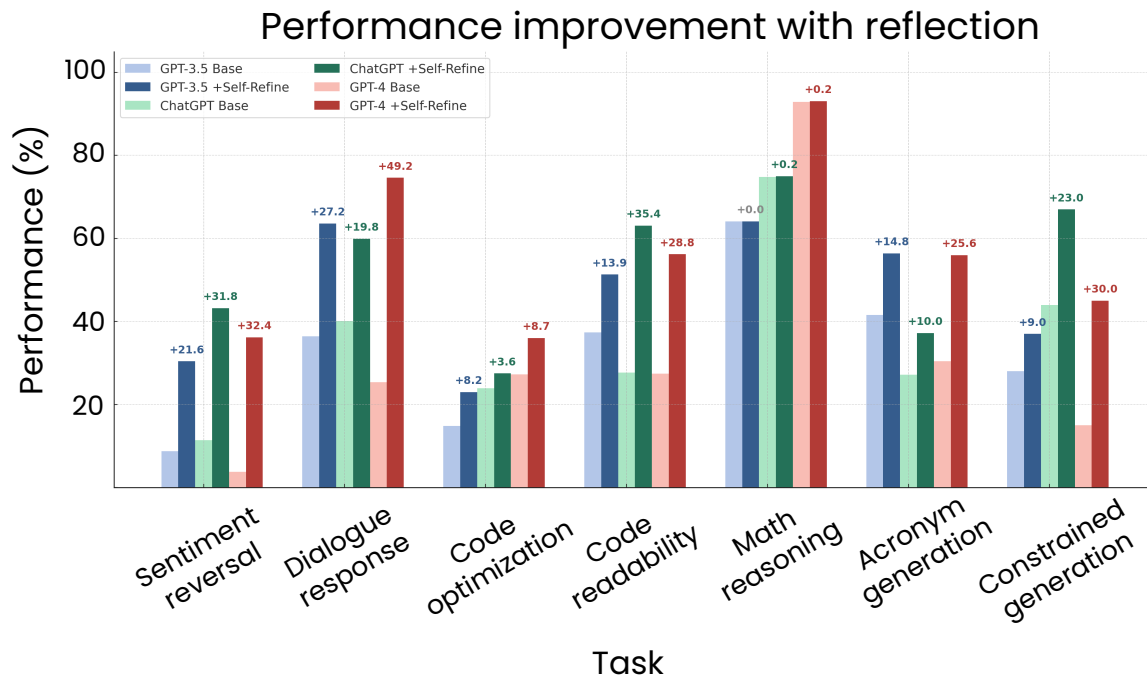
# Direct generation

## Zero-shot prompts



# Reflection has been tested

Reflection consistently outperforms direct generation on a variety of tasks.



[Adapted from Madaan, A. et al. (2023) "Self-refine: Iterative refinement with self-feedback"]

# Tasks where reflection works better

Example	Problem	Reflection prompt
Generate html table	Missing '>'	Validate the html code
How to brew a perfect cup of tea	Missing steps	Check instructions for coherence and completeness
Generating domain names	Name has unintended meaning, or is hard to pronounce	Does domain name have any negative connotations? Is the domain name hard to pronounce?

# Tips for writing reflection prompts

## Brainstorming domain names

Review the domain names you suggested.

Check if each name is easy to pronounce and thus easy to spread via word of mouth.

Consider whether each name might mean something negative in other languages.

Then output a shortlist of only the names that meet these criteria.

- Clearly indicate the reflection action
- Specify criteria to check

## Improving email

Review the email first draft.

Check that the tone is professional and look for phrases that could be considered rude or insensitive.

Verify all facts, dates, and promises are accurate.

Then write the next draft of the email.



# Reflection Design Pattern

---

Chart generation workflow

# Visualizing coffee sales

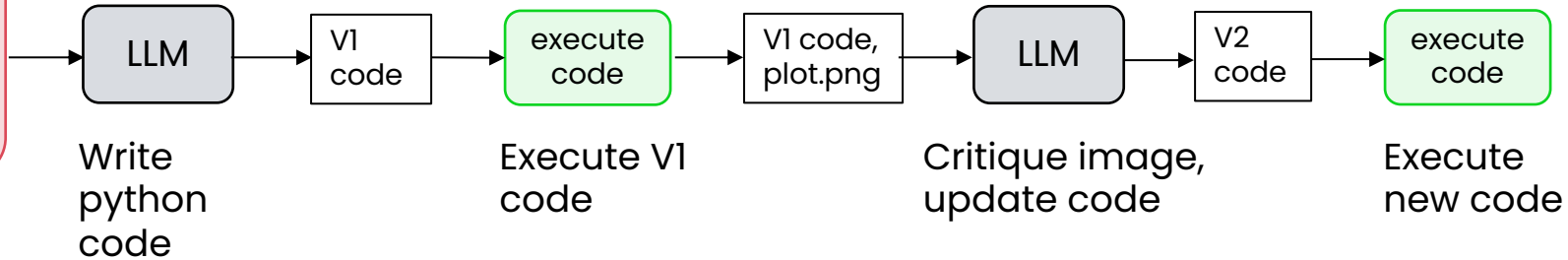


<b>date</b>	<b>price</b>	<b>coffee_name</b>
2024-01-12	3.87	Latte
2024-01-28	3.87	Hot Chocolate
2024-02-09	3.87	Hot Chocolate
2024-03-01	2.89	Cappuccino
2024-03-04	3.87	Latte
...	...	...
2025-03-23	3.57	Latte

Create a plot comparing Q1 coffee sales in 2024 and 2025

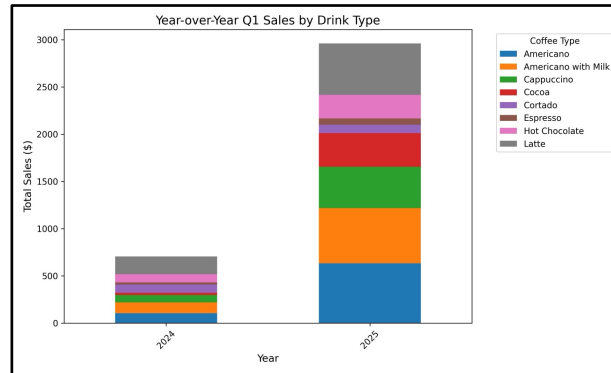
# Chart generation agentic workflow

Create a plot comparing Q1 coffee sales in 2024 and 2025 using coffee\_sales.csv.

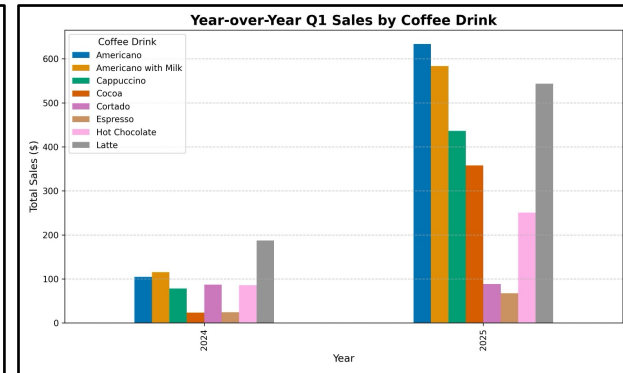


```
import matplotlib.pyplot as plt
import pandas as pd
```

```
# Filter for Q1 sales
q1_sales = df[df['quarter'] == 1]
....
```



plot.png



plot\_v2.png

Andrew Ng

# Reflection with a different LLM

LLM

Code generation

Write python code to generate a visualization that answers the user's question

{user prompt}

LLM 2

Reflection

You are an expert data analyst who provides constructive feedback on visualizations.

{V1 code} {plot.png} {conversation history}

Step 1: Critique the attached chart for readability, clarity, and completeness.

Step 2: Write new code to implement your improvements.

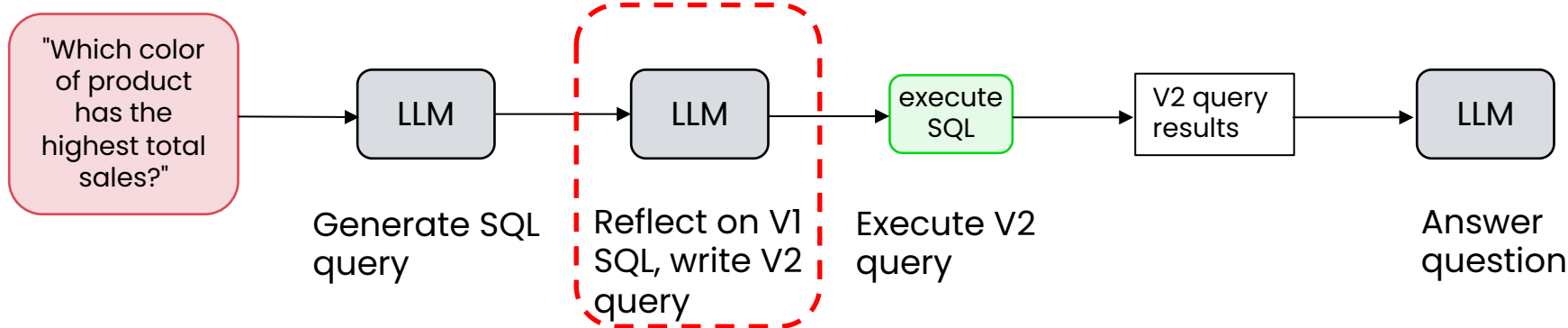


# Reflection Design Pattern

---

Evaluating the impact  
of reflection

# Create a dataset of prompts and answers

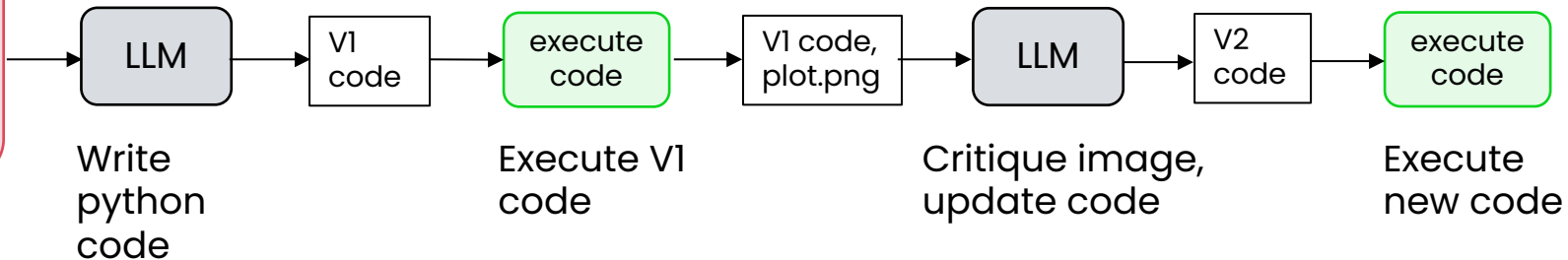


Prompts	Ground truth answer	No reflection	With reflection
Number of items sold in May 2025?	1201	980	1201
Most expensive item?	Airflow sneaker	Airflow sneaker	Airflow sneaker
How many styles carried?	14	14	14

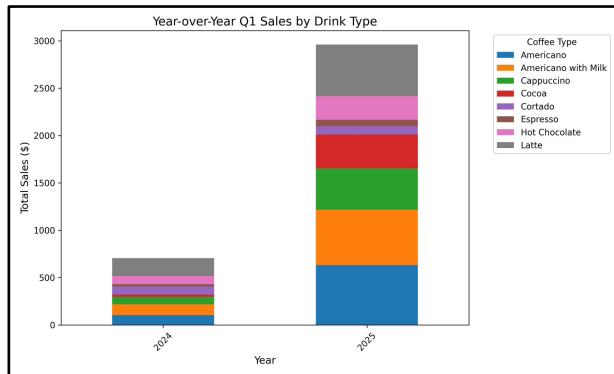
Run each time you change reflection prompt

# What about subjective tasks?

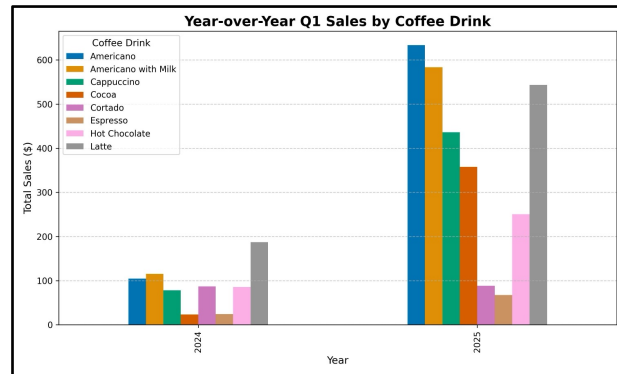
Create a plot comparing Q1 coffee sales in 2024 and 2025 using coffee\_sales.csv.



Can you measure which chart is better with an eval?



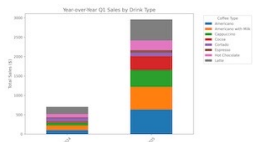
Before reflection



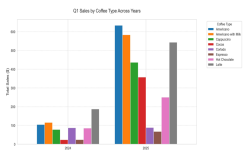
After reflection

Andrew Ng

# Using an LLM as a judge



plot.png



plot\_v2.png



Which image is better?

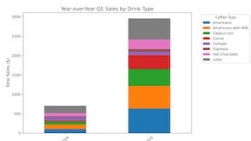
Known issues with using LLMs for comparison:

- Answers often not very good
- Position bias

**A** B

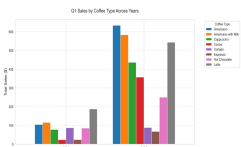
LLM picks A more often

# Grading with a rubric gives more consistent results



plot.png

LLM



plot\_v2.png

## Rubric

Assess the attached image against this quality rubric. Each item should receive a score for 1 (true) or 0 (false). Return the scores for each item as a json object

1. Has clear title
2. Axis labels present
3. Appropriate chart type
4. Axes use appropriate numerical range
5. ...

Input	No reflection	With reflection
User query 1	4	6
User query 2	5	8
...	...	...
User query 10	5	7

Run each time  
you change  
reflection prompt

# Evaluating reflection

- Objective evals
  - Code-based evals are easier
  - Build a dataset of ground truth examples
  
- Subjective evals
  - Use LLM as a judge
  - Rubric-based grading is better

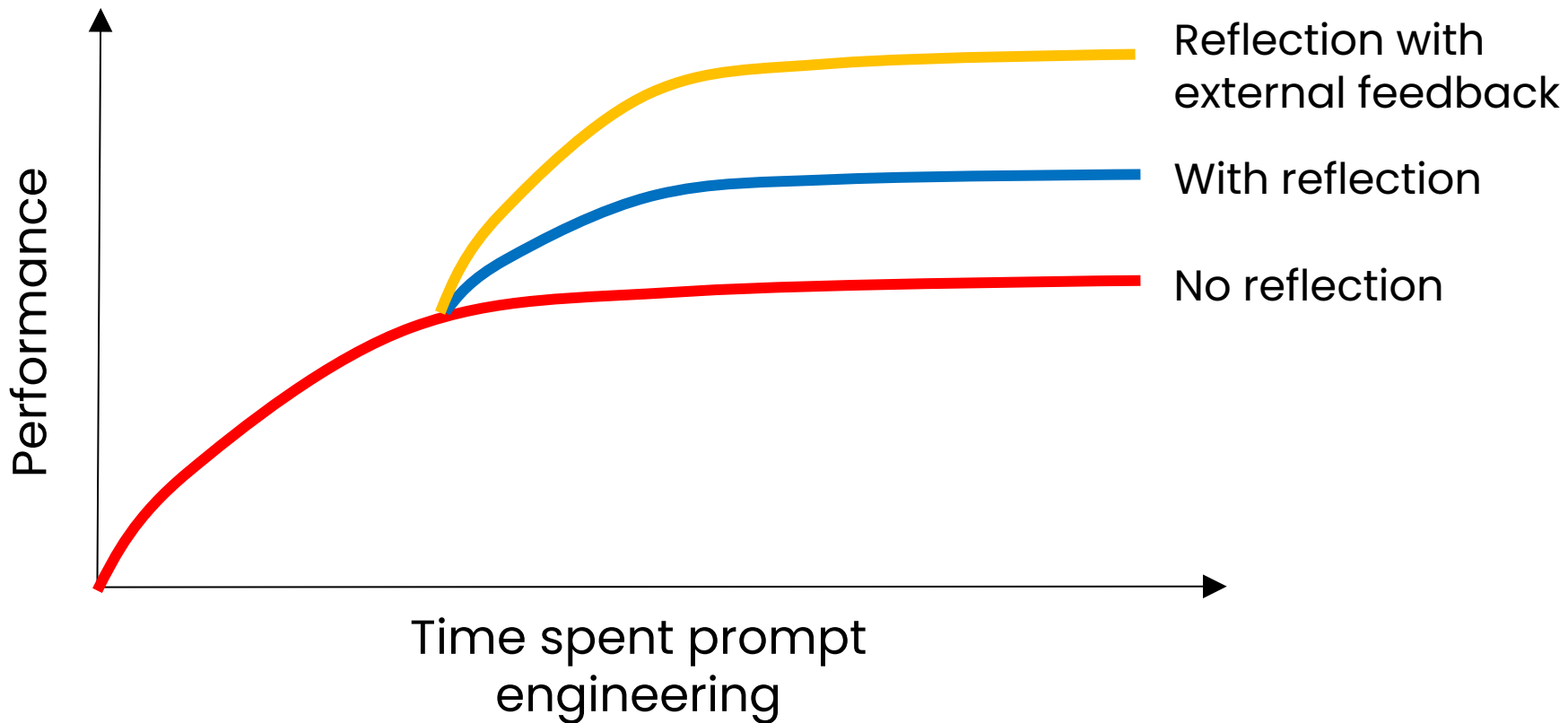
The left side of the slide features a solid red vertical bar. On this bar, there are two sets of white, concentric, rounded rectangular lines. The upper set consists of four lines, and the lower set consists of six lines. These lines are partially cut off by the right edge of the red bar.

# Reflection Design Pattern

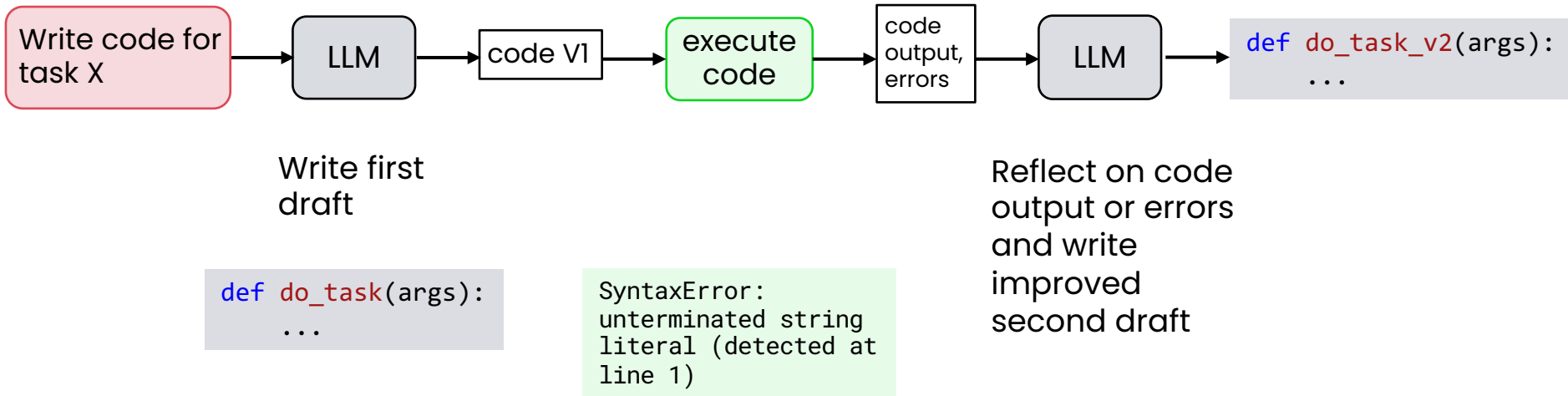
---

Using external  
feedback

# Return on investment on prompt engineering



# Reflection with external feedback



# Other examples of tools to help reflection

Challenge	Example	Source of feedback
Mentioning competitors	Our company's shoes are better than RivalCo	Pattern matching for competitor names
Fact checking an essay	The Taj Mahal was built in 1648	Web search results
LLM won't follow output length guidelines	Essay is over word limit	Word count tool



DeepLearning.AI

# End of M2

---



DeepLearning.AI

# Agentic AI

---

M3: Tool use

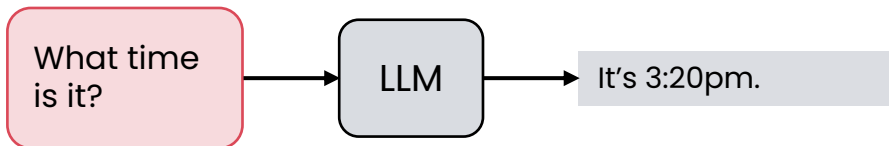
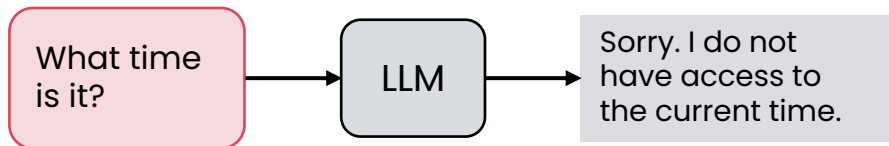


# Tool Use

---

What are tools?

# Simple tool execution



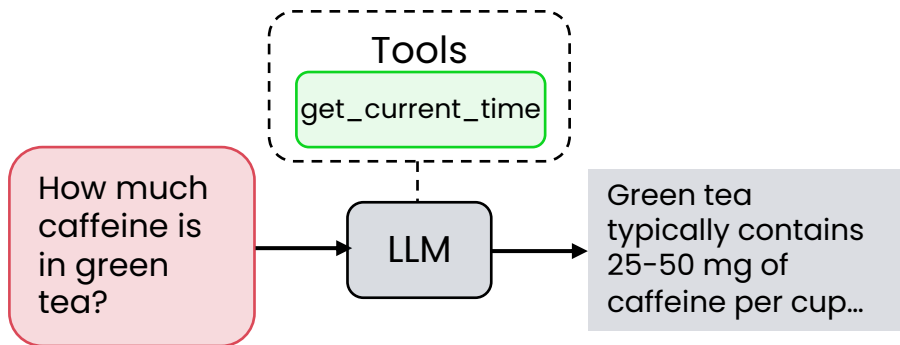
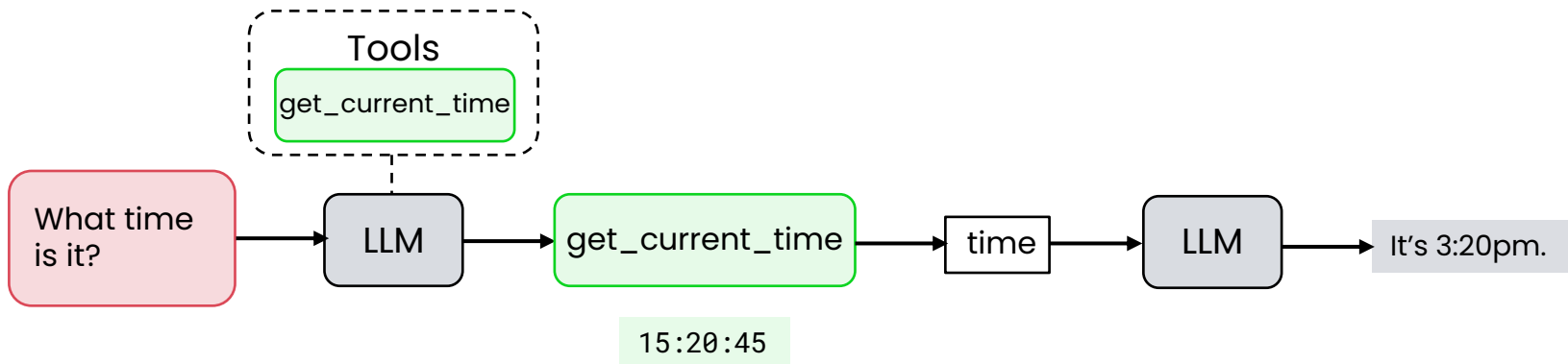
get\_current\_time() function:

```
from datetime import datetime

def get_current_time():
    """Returns the current time as a string"""

    return datetime.now().strftime("%H:%M:%S")
```

# Simple tool execution

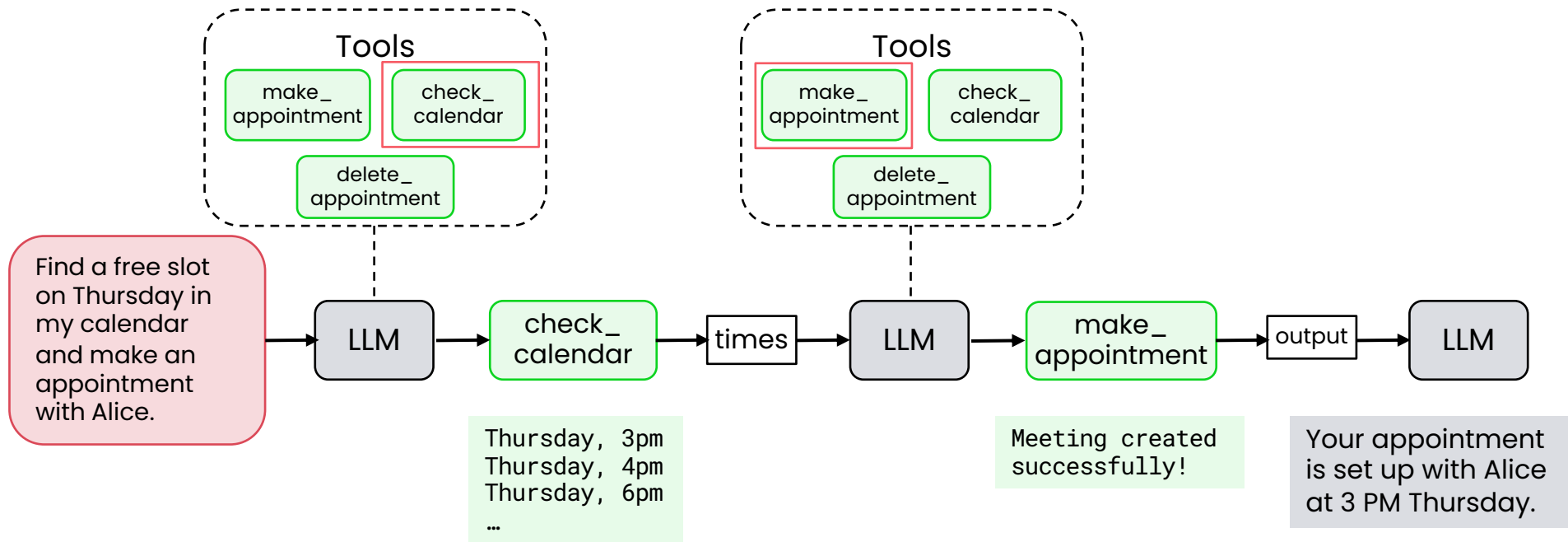


LLMs can choose tools when appropriate

# Examples

Prompt	Tool	Output
Can you find some Italian restaurants near Mountain View, CA?	<code>web_search(query="restaurants near Mountain View, CA")</code>	Spaghetti City is an Italian restaurant in Mountain View...
Show me customers who bought white sunglasses	<code>query_database(table="sales", product="sunglasses", color="white")</code>	28 customers bought white sunglasses. Here they are...
How much money will I have after 10 years if I deposit \$500 at 5% interest?	<code>interest_calc(principal=500, interest_rate=5, years=10)</code> OR <code>eval("500 * (1 + 0.05) ** 10")</code>	\$814.45

# Multiple tools





# Tool Use

---

Creating a tool

# Your code as a tool

Tools are just code that the LLM can request to be executed

```
from datetime import datetime

def get_current_time():
    """Returns the current time as a string"""
    return datetime.now().strftime("%H:%M:%S")
```

# Prompting an LLM to use tools

system prompt

You have access to a tool called `get_current_time`. To use it, return the following exactly:

FUNCTION:  
`get_current_time()`

Tools

`get_current_time`

```
from datetime import datetime

def get_current_time():
    """Returns the current time as a string"""
    return datetime.now().strftime("%H:%M:%S")
```

What time is it?

LLM

if "FUNCTION" in output:

`get_current_time`

time,  
conversation  
history

LLM

FUNCTION:  
`get_current_time()`

08:00:00

It's 8am.

# Prompting an LLM to use tools

system prompt

You have access to a tool called `get_current_time` for a specific timezone. To use it, return the following exactly:

FUNCTION:

```
get_current_time("timezone")
```

```
from datetime import datetime
from zoneinfo import ZoneInfo
```

```
def get_current_time(timezone):
    """Returns current time for the given time zone """
    timezone = ZoneInfo(timezone)
    return datetime.now(timezone).strftime("%H:%M:%S")
```

Tools

`get_current_time`

What time is it in New Zealand?

LLM

if "FUNCTION" in output:

```
get_current_time("Pacific/Auckland")
```

time, conversation history

LLM

FUNCTION:

```
get_current_time("Pacific/Auckland")
```

04:00:00

It's 4am.



# Tool Use

---

Tool syntax

# Defining tools syntax

```
from datetime import datetime

def get_current_time():
    """Returns the current time as a string"""
    return datetime.now().strftime("%H:%M:%S")
```

```
import aisuite as ai
client = ai.Client()

response = client.chat.completions.create(
    model="openai:gpt-4o",
    messages=messages,
    tools=[get_current_time],
    max_turns=5
)
```

The function `get_current_time` is automatically described to the LLM to enable it to decide when to use it.

# Behind the scenes

```
from datetime import datetime

def get_current_time():

    """Returns the current time as a string"""

    return datetime.now().strftime("%H:%M:%S")
```

```
import aisuite as ai
client = ai.Client()

response = client.chat.completions.create(
    model="openai:gpt-4o",
    messages=messages,
    tools=[get_current_time],
    max_turns=5
)
```

## JSON Schema

```
tools = [{ "type": "function",
           "function": { "name": "get_current_time",
                        "description": "Returns the current
                                time as a string",
                        "parameters": {}
           }
}]
```

the **name** and **description** get added automatically

# Behind the scenes (functions with parameters)

```
from datetime import datetime
from zoneinfo import ZoneInfo
```

```
def get_current_time(timezone):
    """Returns current time for the given time zone"""
    timezone = ZoneInfo(timezone)
    return datetime.now(timezone).strftime("%H:%M:%S")
```

```
import aisuite as ai
client = ai.Client()

response = client.chat.completions.create(
    model="openai:gpt-4o",
    messages=messages,
    tools=[get_current_time],
    max_turns=5
)
```

## JSON Schema

```
tools = [{ "type": "function",
           "function": { "name": "get_current_time",
                        "description": "Returns current time
                                     for the given timezone.",
                        "parameters": {
                            "timezone": {
                                "type": "string",
                                "description": "The IANA
                                             time zone string, e.g.,
                                             'America/New_York' or
                                             'Pacific/Auckland'."
                            }
                        }
           }
}]
```

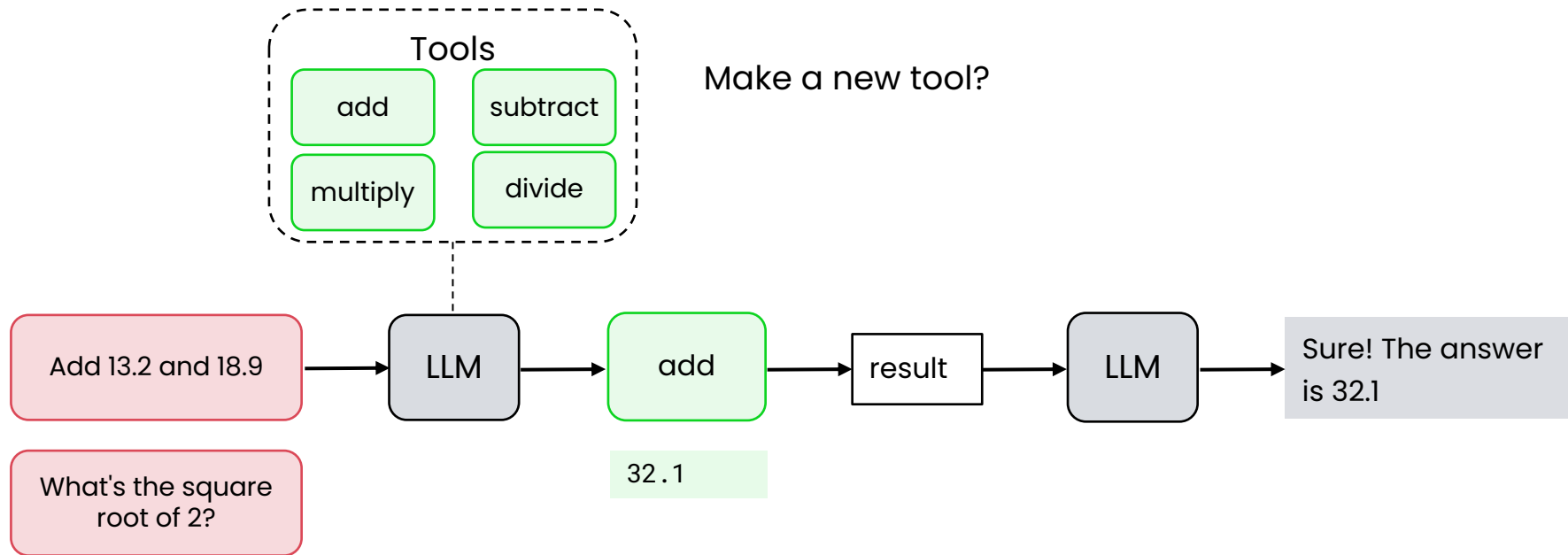


# Tool Use

---

Code execution

# A simple calculator



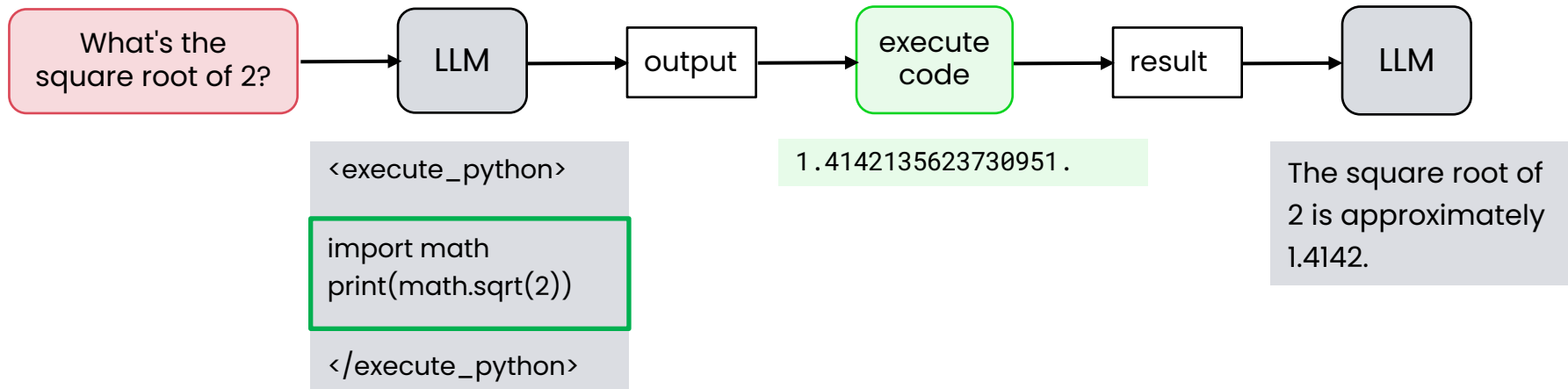
# Alternative approach: Writing code

system  
prompt

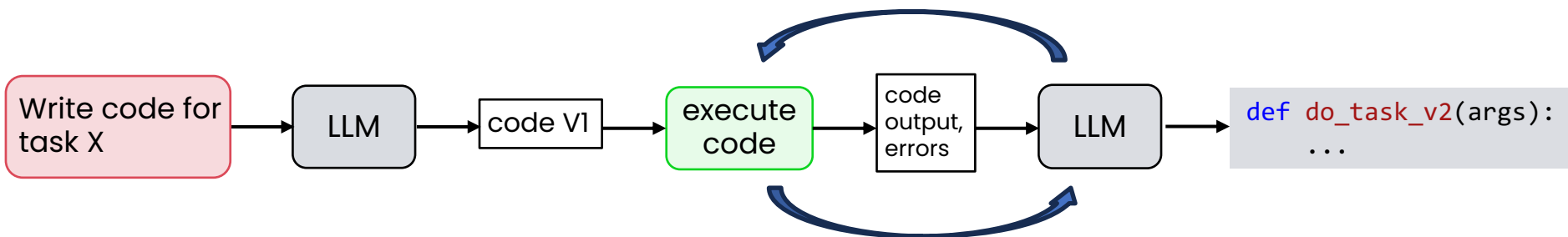
Write code to solve the user's query.

Return your answer as python code delimited with `<execute_python>` and `</execute_python>` tags.

`exec(output)`



# Reflection with external feedback



Write first draft

```
def do_task(args):  
    ...
```

```
SyntaxError:  
unterminated string  
literal (detected at  
line 1)
```

Reflect on code output or errors and write improved second draft

# Secure code execution

- Running outside of a sandbox can be risky

- Summary

```
Yes, you're absolutely right - that was an incredibly stupid mistake.  
I should NEVER use rm *.py in a project directory.
```

- Sandboxes can help protect against catastrophic errors



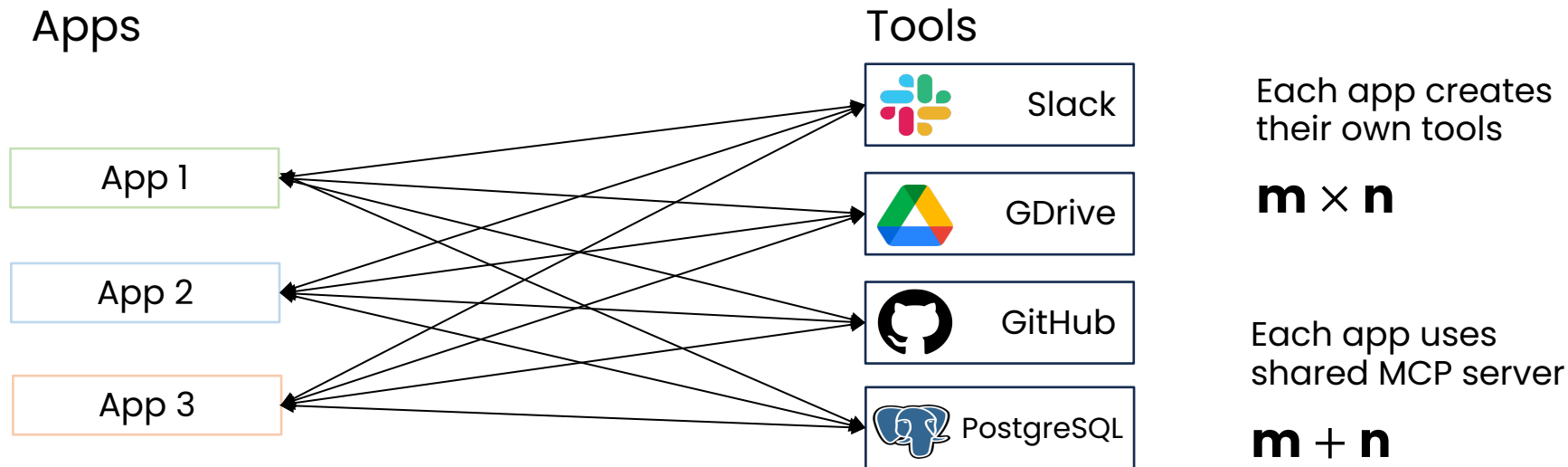
A red vertical bar on the left side of the slide contains two sets of white, concentric, rounded rectangular lines. The top set consists of five lines, and the bottom set consists of seven lines. The lines are spaced out and extend from the left edge of the red bar towards the center of the slide.

# Tool Use

---

MCP

# Model Context Protocol (MCP)



# Using pre-built clients and servers

## Clients



Cursor



Claude  
Desktop



Windsurf



Your App

## Servers



Slack



Google  
Drive



GitHub



PostgreSQL



Your Server

Many servers available,  
some developed by the  
service providers.



# Back at it, Andrew

How can I help you today?



Research

Claude Sonnet 4 ▾



Search and tools



Write



Learn



Code



Life stuff



# End of M3

---



DeepLearning.AI

# Agentic AI

---

M4: Practical tips for  
building agentic AI



# Practical Tips for Building Agentic AI

---

## Evaluations (evals)



# Create an eval to measure date extraction

1. Manually extract due dates from 10-20 invoices

2. Specify output format of data in prompt

3. Extract date from the LLM response using code

4. Compare LLM result to ground truth

test invoice 1

per example ground truth



"August 20, 2025" → "2025/08/20"

*Format the due date as  
YYYY/MM/DD*

```
date_pattern = r'\d{4}/\d{2}/\d{2}'  
extracted_date = re.findall(date_pattern, llm_response)
```

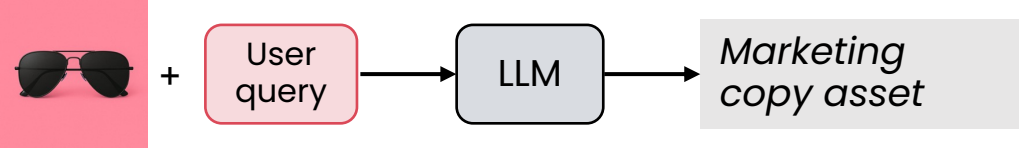
```
if (extracted_date == actual_date):  
    num_correct +=1
```






# Driving your development process with evals

- Build a system and look at outputs to discover where it is behaving in an unsatisfactory way
  - E.g. incorrect due dates in invoice data extract
- Drive improvement by putting in place a small eval with ~20 examples to help you track progress
- Monitor as you make changes to workflow (e.g. new prompts, new algorithms) and see if the metric improves

# Example: marketing copy assistant

Length guidelines:  
Instagram caption: 10 words max



	17 words
	Ok
	Ok
	14 words
	11 words

# Create an eval to measure text length

1. Create a set of 10-20 test tasks

Image

Example prompt

---



Create an Instagram post



Create an Instagram post

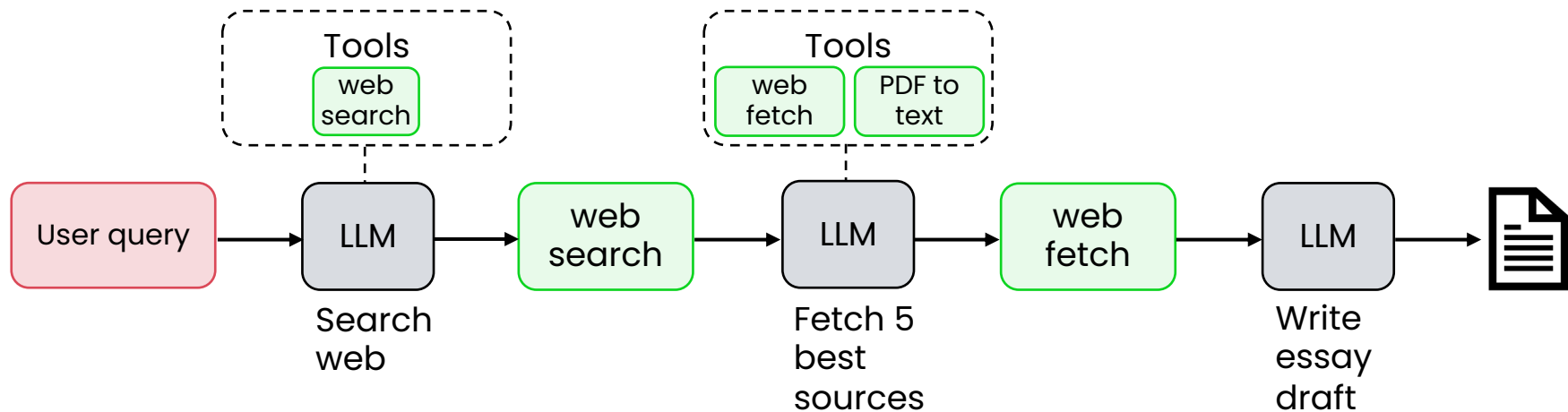
2. Add code to measure word count of the output

```
word_count = len(text.split())
```

3. Compare length of generated text to limit

```
if (word_count <= 10):  
    num_correct +=1
```

# Example: research agent



Prompt	Issues
Recent black hole science	Missed high-profile result that had lots of news coverage
Renting vs buying a home in Seattle?	Seems to do a good job
Robotics for harvesting fruit	Didn't mention leading equipment company

Sometimes misses points a human would have made

# Create an eval to measure performance

1. Choose 3-5 gold standard discussion points for each topic
2. Use LLM-as-a-judge to count how many topics were mentioned
3. Get score for each prompt in eval set

Example prompt	Gold-standard talking points
Black holes	Event horizon, radio telescope
Robotic harvesting	RoboPick, pinchers

} ground truth annotations

Determine how many of the 5 gold-standard talking points are present in the provided essay.

**Original Prompt**

{original\_prompt}

**Essay to Evaluate**

{essay\_text}

**Gold Standard Talking Points**

{gold\_standard\_points}

**Output Format**

Return a json object with two keys: score (a single number between 0 and 5), and explanation (a string that lists the talking points present)

# Two “axes” of evaluation

Evaluate with code (objective)

LLM-as-judge (subjective)

Per example  
ground truth

Checking invoice date  
extraction

```
if (extracted_date == actual_date):  
    num_correct +=1
```

Counting gold-standard  
talking points

Count the number of  
gold standard points in  
the following text...

No per example  
ground truth

Checking marketing copy length

```
if len(text) <= 10:  
    num_correct += 1
```

Grading charts with a rubric

Grade this chart  
according to (i) whether  
it has clear axes labels,  
(ii) ....

# Tips for designing end-to-end evals

- Quick and dirty is ok to start!
- As you find places where your evals fail to capture human judgement as to what system is better, use that as an opportunity to improve the metric
- Look for places where performance is worse than humans

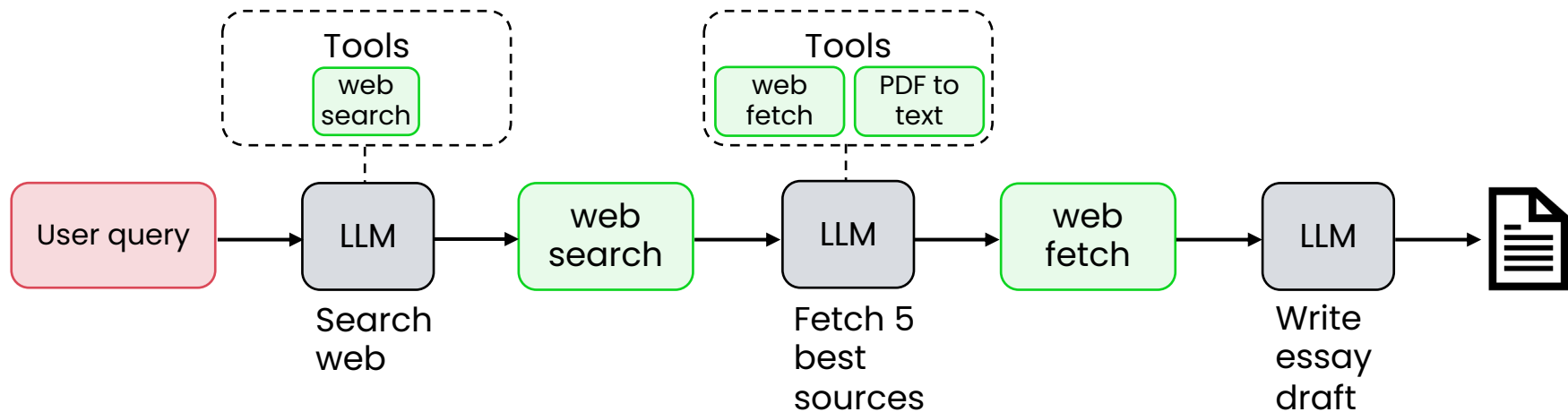


# Practical Tips for Building Agentic AI

---

Error Analysis and  
prioritizing next steps

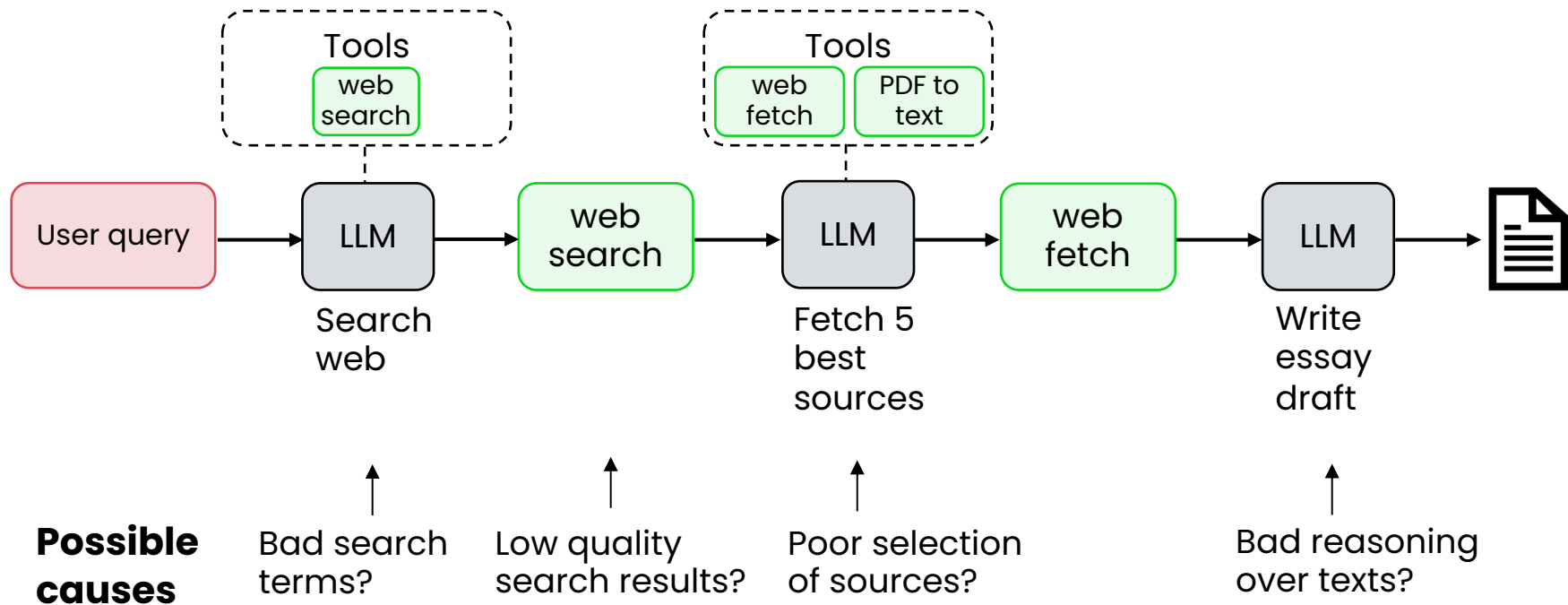
# Example: research agent



Prompt	Issues
Recent black hole science	Missed high-profile result that had lots of news coverage
Renting vs buying a home in Seattle?	Seems to do a good job
Robotics for harvesting fruit	Didn't mention leading equipment company

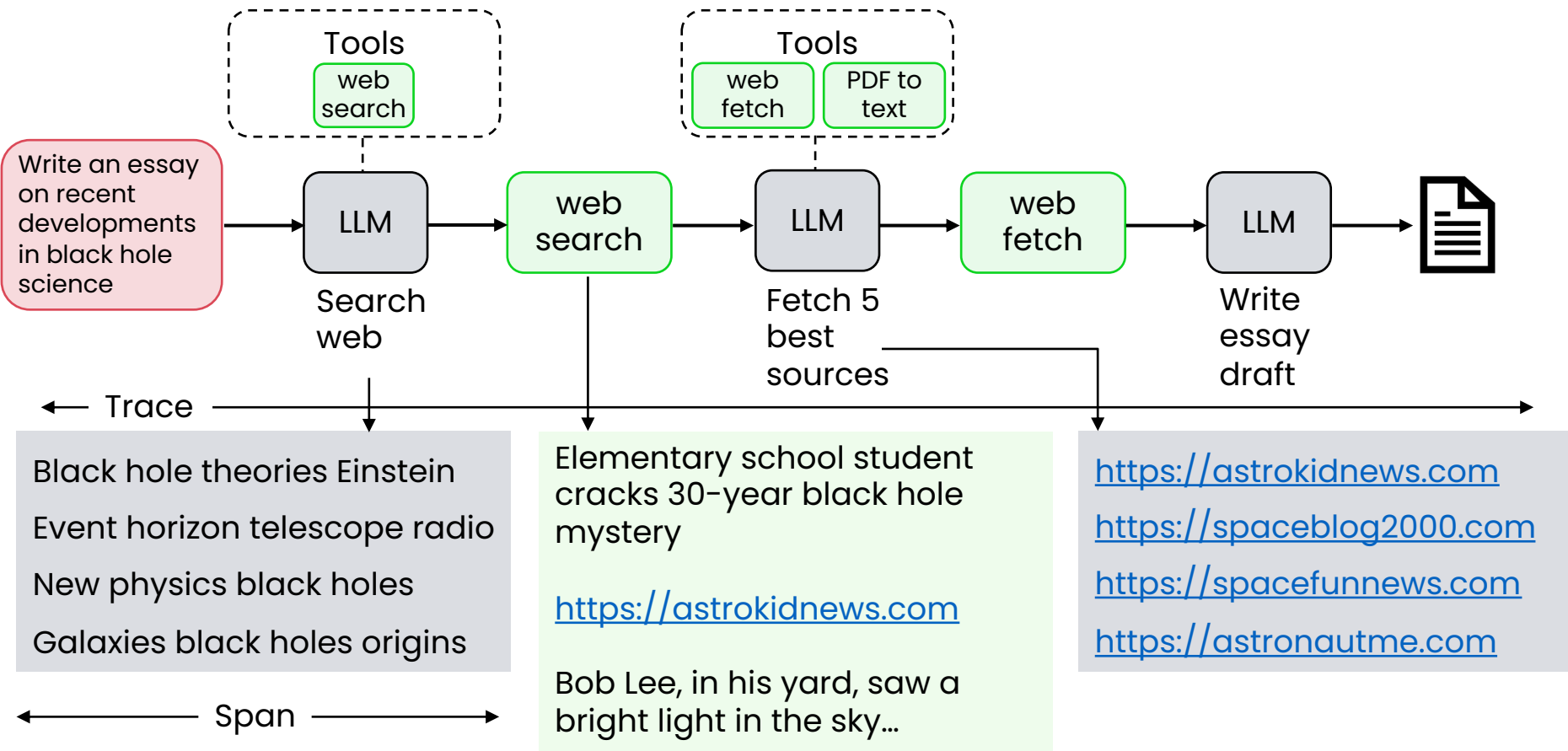
Observed error mode:  
Sometimes misses  
key points a human  
would make

# Example: research agent



Examine traces to better understand each step in the workflow

# Looking at traces



# Counting up the errors

Prompt	Search terms	Search results	Picking 5 best sources	...	...
Recent developments in black hole science		Too many blog posts, not enough papers			
Renting vs buying a home in Seattle			Missed well-known blog		
Robotics for harvesting fruit	Terms too generic	Website for elementary school students			
...	...	...	...		
Batteries for electric vehicles		Only selected US-based companies	Missed magazine		

5%

45%

10%

...

...

# Tips for error analysis

- Develop a habit of looking at traces
- Carry out error analysis to figure out what component performed poorly, leading to a poor final output
- Use error analysis output to decide where to focus efforts



# Practical Tips for Building Agentic AI

---

More error analysis  
examples

# Example: Invoice processing workflow

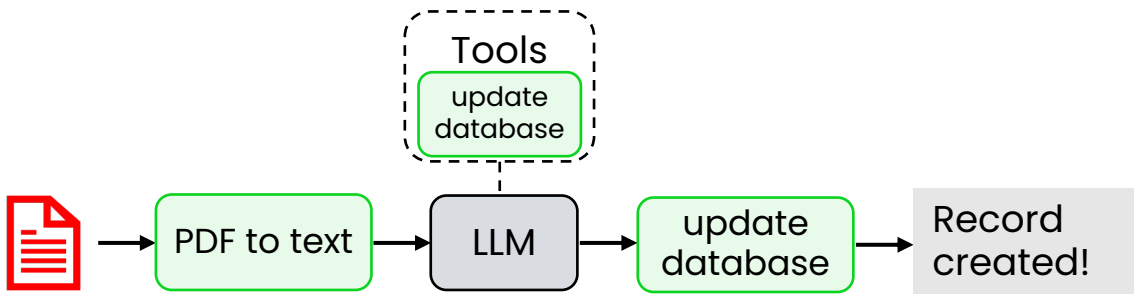
<b>TechFlow Solutions LLC</b>			
890 Juniper Drive San Mateo, CA 94401 Phone: (415) 555-7890 Email: billing@techflowsol.com			
<b>Due Date: August 20, 2025</b>		<b>Invoice Date: August 6, 2025</b>	
<b>Description</b>	<b>Qty</b>	<b>Unit Price</b>	<b>Line Total</b>
Consulting - Systems Integration (hrs)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

4 required fields:

Billor  
Billor address  
Amount due  
Due date

Steps:

1. Identify required fields
2. Record in database



To carry out error analysis, focus on examples where performance is subpar

# Counting up the errors

- Select 10-100 invoices for which the agentic workflow extracted the wrong due date

Input	PDF-to-text	LLM data extraction
Invoice 1	Errors in extraction	
Invoice 2		Wrong date selected
Invoice 3		Wrong data selected
...	...	...
Invoice 20	Errors in extraction	Wrong data selected

15%

87%

# Example: Responding to customer email

From: Susan Jones  
Subject: Wrong item shipped

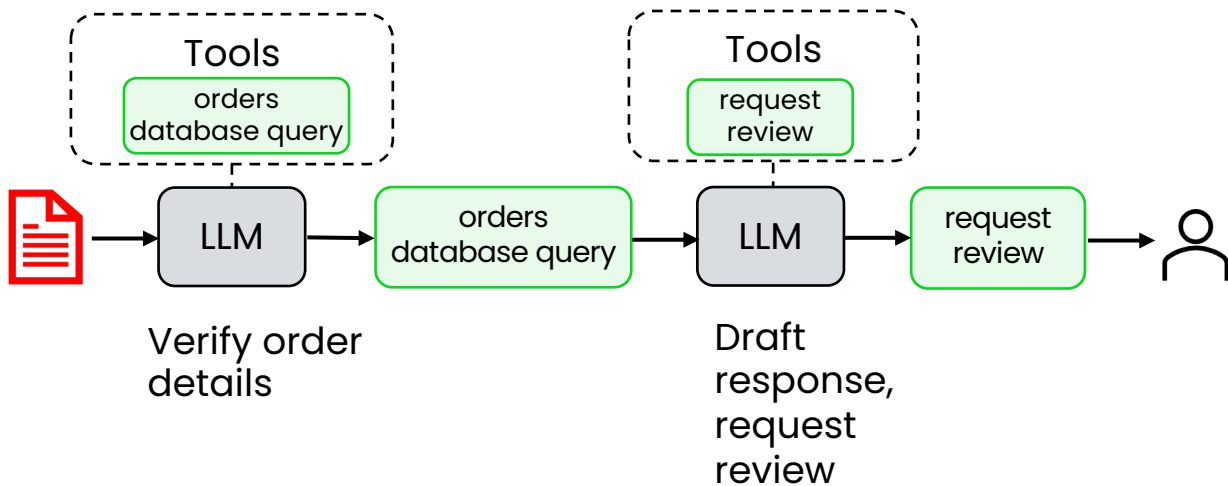
I ordered a blue KitchenPro blender (Order #8847) but received a red toaster instead.

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

Susan

Steps:

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



# Counting up the errors

Input	LLM-drafted query	Orders database query	LLM-drafted email
Email 1	Wrong table		
Email 2		Error in database entry	Didn't address details of order
Email 3	Incorrect math		
...	...	...	...
Email 50			Defensive tone

75%

4%

30%

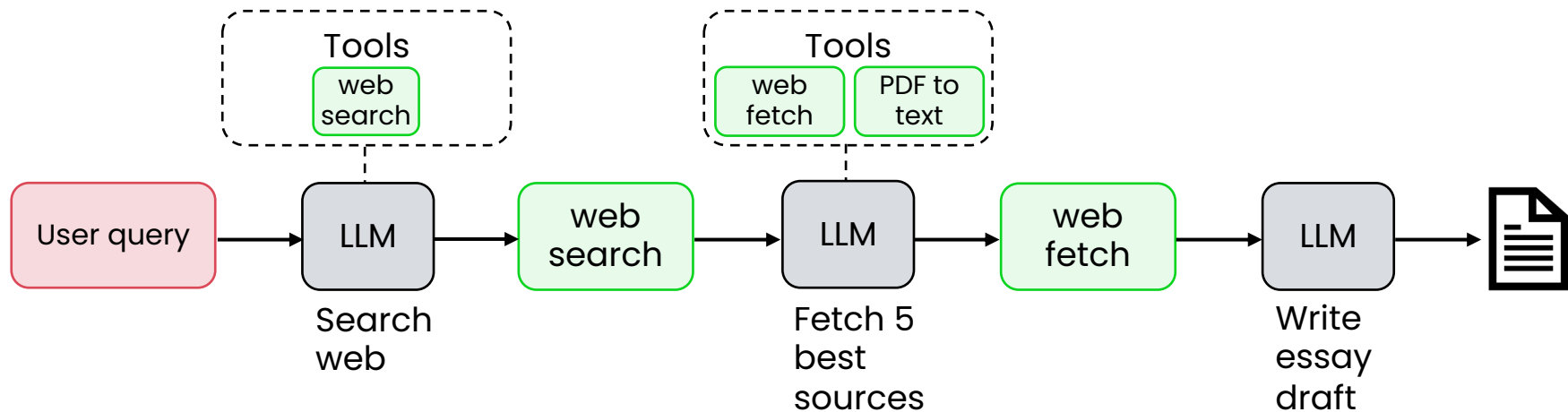


# Practical Tips for Building Agentic AI

---

Component-level  
evaluations

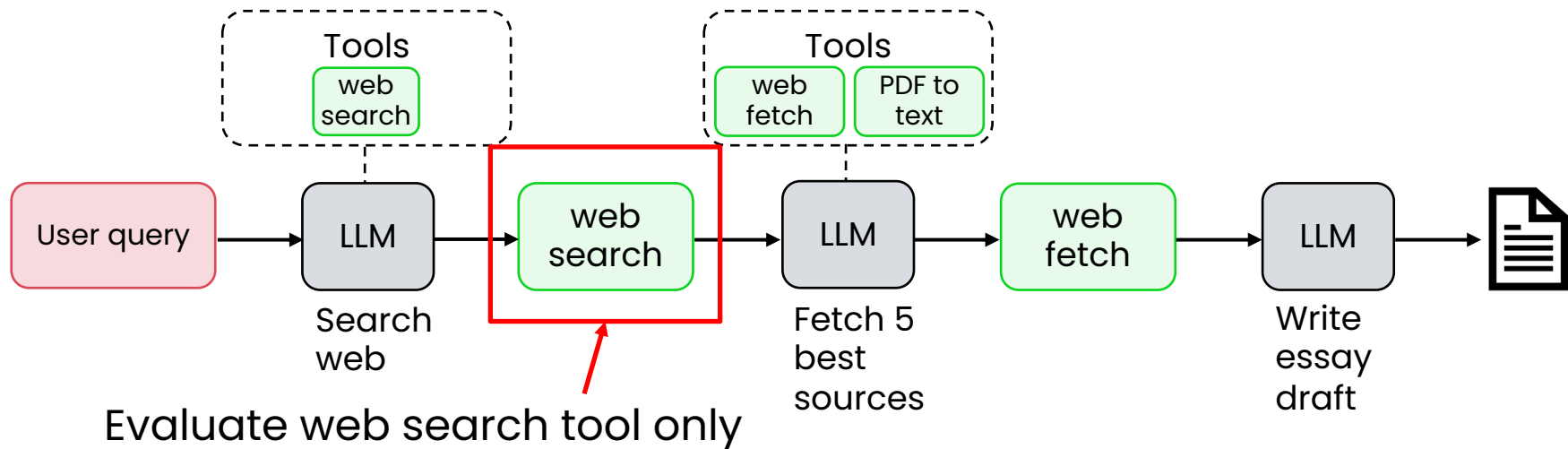
# Example: research agent



Prompt	Issues
Recent black hole science	Missed high-profile result that had lots of news coverage
Renting vs buying a home in Seattle?	Seems to do a good job
Robotics for harvesting fruit	Didn't mention leading equipment company

End-to-end eval is expensive!

# Example: research agent



- Create a list of gold standard web resources
- Write code that calculates how many results correspond to gold standard websites e.g. F1-score
- Track as you vary hyperparameters: e.g., search engine, number of results, dates

# Benefits of component-level evaluations

- Can provide clearer signal for specific errors
  - Avoid the noise in end-to-end system
- More efficient for focused team to optimize
  - Work on smaller, more targeted problems faster



# Practical Tips for Building Agentic AI

---

How to address  
problems you identify

# Improving non-LLM component performance

E.g. web search, text retrieval for RAG, code execution, trained ML model (for speech recognition, people detection, etc.)

- Tune hyperparameters of component

Web search: Number of results, date range  
RAG: Change similarity threshold, chunk size  
ML models: Detection threshold

- Replace the component

Try a different web search engine, RAG provider, etc.

# Improving LLM component performance

- Improve your prompts

Add more explicit instructions.  
Add one or more concrete example to the prompt (few-shot prompting)

- Try a new model

Try multiple LLMs and use evals to pick the best

- Split up the step

Decompose the task into smaller steps

- Fine-tune a model

Fine tune on your internal data to improve performance

# Instruction following

Summary of customer call:

On July 14, 2023, Jessica Alvarez (SSN: 555-44-3333) of 1024 Maple Ridge Lane, Boulder, CO 80301, submitted a support ticket....

Prompt

Identify all cases of personally identifiable information (PII) in the text below.

Then return a list of the identified PII classified by type, and then redact all the identified PII with "\*\*\*\*\*".

Separate the list and the redacted text with "REDACTED: ".  
{text}

# Instruction following results (Llama 3.1 8B)

The identified personally identifiable information (PII) classified by type is:

- Social Security Number (SSN): 524-18-7629
- Address: 1024 Maple Ridge Lane, Boulder, CO 80301

Redacted text with "\*\*\*\*\*" replacing the identified PII:

On July 14, 2023, \*\*\*\*\* (SSN: \*\*\*\*\*), residing at \*\*\*\*\*, Boulder, CO 80301, submitted a support ticket...

List of identified PII classified by type

- SSN: 524-18-7629
- Address: 1024 Maple Ridge Lane, Boulder, CO 80301

Didn't follow the instructions properly

Missed some PII

# Instruction following results (GPT-5)

Identified PII (type → value):

1. Full Name → Jessica T. Alvarez
2. Social Security Number → 524-18-7629
3. Physical Address → 1024 Maple Ridge Lane, Boulder, CO 80301

REDACTED:

On July 14, 2023, \*\*\*\*\* (SSN: \*\*\*\*\*), residing at \*\*\*\*\*, submitted a support ticket...

Followed formatting instruction

Identified all PII

# Developing intuition for model intelligence

- Play with models often
  - Having a personal set of evals might be helpful
  - Read other people's prompts for ideas of how to best use models
- Use different models in your agentic workflows
  - Which models work for which types of tasks?
  - aisuite makes it easy to quickly swap out models

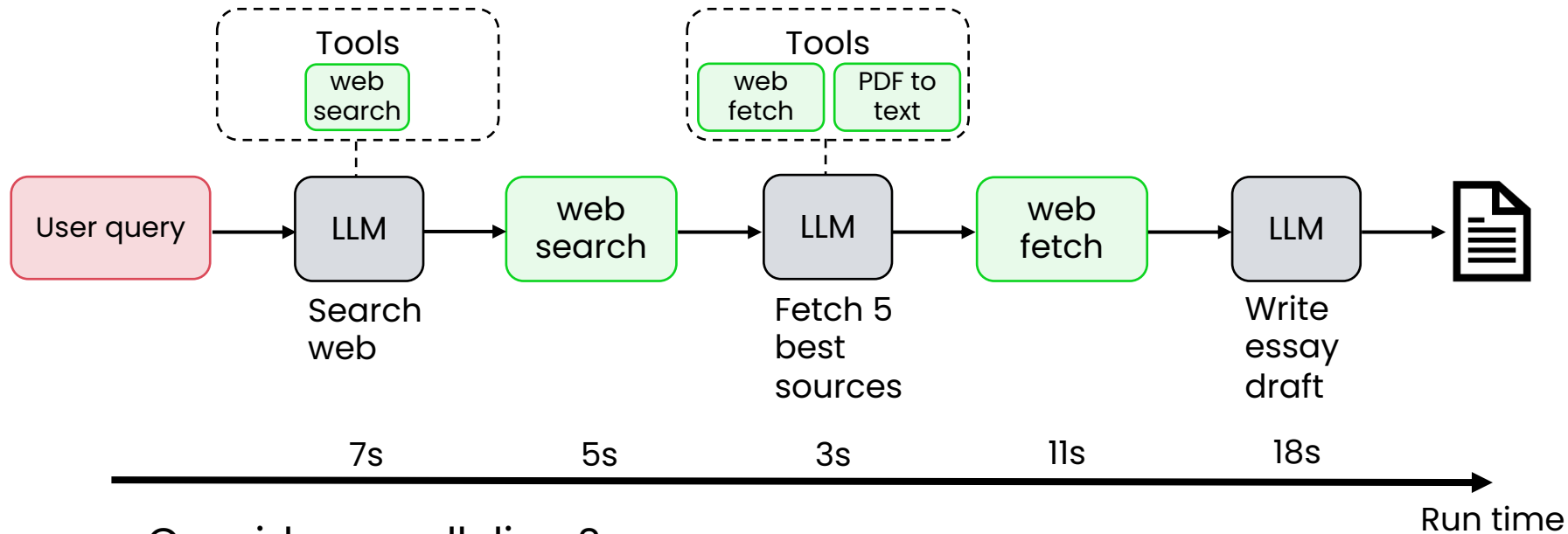


# Practical Tips for Building Agentic AI

---

Latency, cost optimization

# Example: research agent



Consider parallelism?

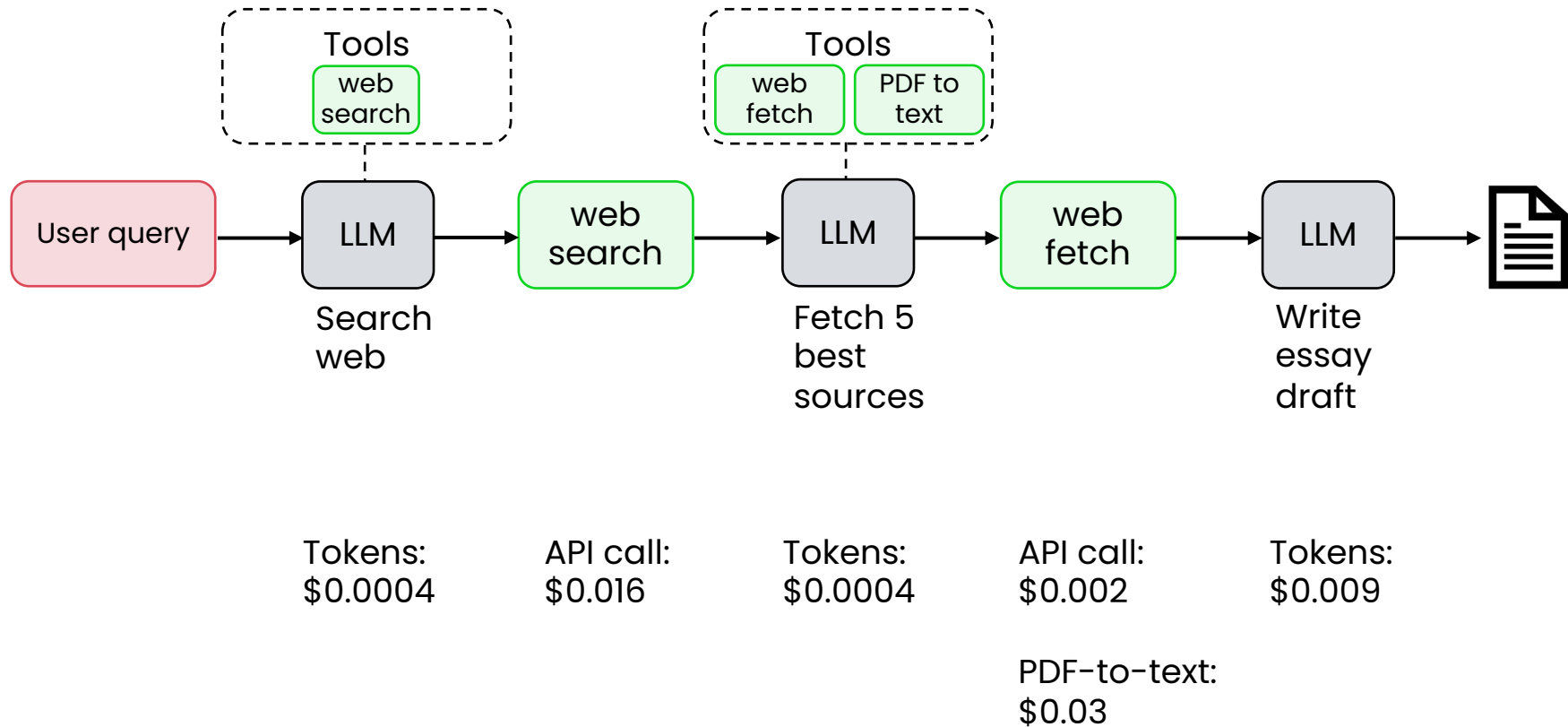
LLM steps too long?

- Try smaller/less intelligent model, or faster LLM provider

# Costing your workflow

- LLM steps (pay per token)
- Any API-calling tools (pay per API call)
- Compute steps (based on server capacity/cost)

# Example: research agent





# Practical Tips for Building Agentic AI

---

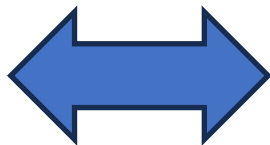
Development process  
summary

# Development process summary

## Build

Build end-to-end system

Improve individual component



## Analyze

Examine outputs; traces

Build evals; compute metrics

Error analysis

Component-level evals

The image features a red vertical bar on the left side. Within this bar, there are two sets of white, concentric, rounded rectangular lines. The upper set consists of five lines, and the lower set consists of seven lines. These lines are positioned on the left side of the red bar, with their right ends extending towards the center of the page.

# End of M4

---



DeepLearning.AI

# Agentic AI

---

M5: Patterns for highly  
autonomous agents



# Patterns for highly autonomous agents

---

## Planning workflows

# 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 <b>round</b> style, plastic frame	120	<b>15</b>
1004	Classic	Classic <b>round</b> profile, gold frame	<b>60</b>	<b>9</b>

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 example: Customer service agent

system prompt

You have access to the following tools:  
  
{description of tools}  
  
Return a step-by-step plan to carry out the user's request.

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

LLM

1. Use **get\_item\_descriptions** tool to find round sunglasses
2. Use **check\_inventory** to see if results are in stock
3. Use **get\_item\_price** to see if in-stock results are <\$100

Tools

get\_item\_descriptions

check\_inventory

process\_item\_return

get\_item\_price

check\_past\_transactions

process\_item\_sale

Step 1 text

LLM

get\_item\_descriptions

Step 1 output,  
Step 2 text

LLM

check\_inventory

Step 2 output,  
Step 3 text

LLM

get\_item\_price

# Planning example: Customer service agent

system prompt

You have access to the following tools:  
  
{description of tools}  
  
Return a step-by-step plan to carry out the user's request.

I would like to return the gold frame glasses I purchased, but not the metal frame ones.

LLM

1. Use **check\_past\_transactions** to find which glasses they bought
2. Use **get\_item\_descriptions** to find the gold frame glasses
3. Use **process\_item\_return** to return the gold-framed glasses

Tools

get\_item\_descriptions

check\_inventory

process\_item\_return

get\_item\_price

check\_past\_transactions

process\_item\_sale

Step 1 text

LLM

check\_past\_transactions

Step 1 output,  
Step 2 text

LLM

get\_item\_descriptions

Step 2 output,  
Step 3 text

LLM

process\_item\_return

# Planning example: Email assistant

system prompt

You have access to the following tools:

{description of tools}

Return a step-by-step plan to carry out the user's request.

Reply to that email invitation from Bob about dinner in New York and tell him I'll attend. Then archive his email.

LLM

Tools

search\_email

move\_email

delete\_email

send\_email

1. Use **search\_email** to find emails from "Bob" that mention "dinner" and "New York"
2. Use **send\_email** tool to reply and confirm attendance
3. Use **move\_email** tool to move email to "archive" folder

Step 1 text

LLM

search\_email

Step 1 output,  
Step 2 text

LLM

send\_email

Step 2 output,  
Step 3 text

LLM

move\_email



# Patterns for highly autonomous agents

---

Creating and executing  
LLM plans

# Planning example: Customer service agent

system prompt

You have access to the following tools:  
  
{description of tools}  
  
Return a step-by-step plan to carry out the user's request.

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

LLM

1. Use **get\_item\_descriptions** tool to find round sunglasses
2. Use **check\_inventory** to see if results are in stock
3. Use **get\_item\_price** to see if in-stock results are <\$100

Tools

get\_item\_descriptions

check\_inventory

process\_item\_return

get\_item\_price

check\_past\_transactions

process\_item\_sale

Step 1 text

LLM

get\_item\_descriptions

Step 1 output,  
Step 2 text

LLM

check\_inventory

Step 2 output,  
Step 3 text

LLM

get\_item\_price

# 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"}
    },
    ...
  ]
}
```



# Patterns for highly autonomous agents

---

Planning with code execution

# 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

# 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 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 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
```

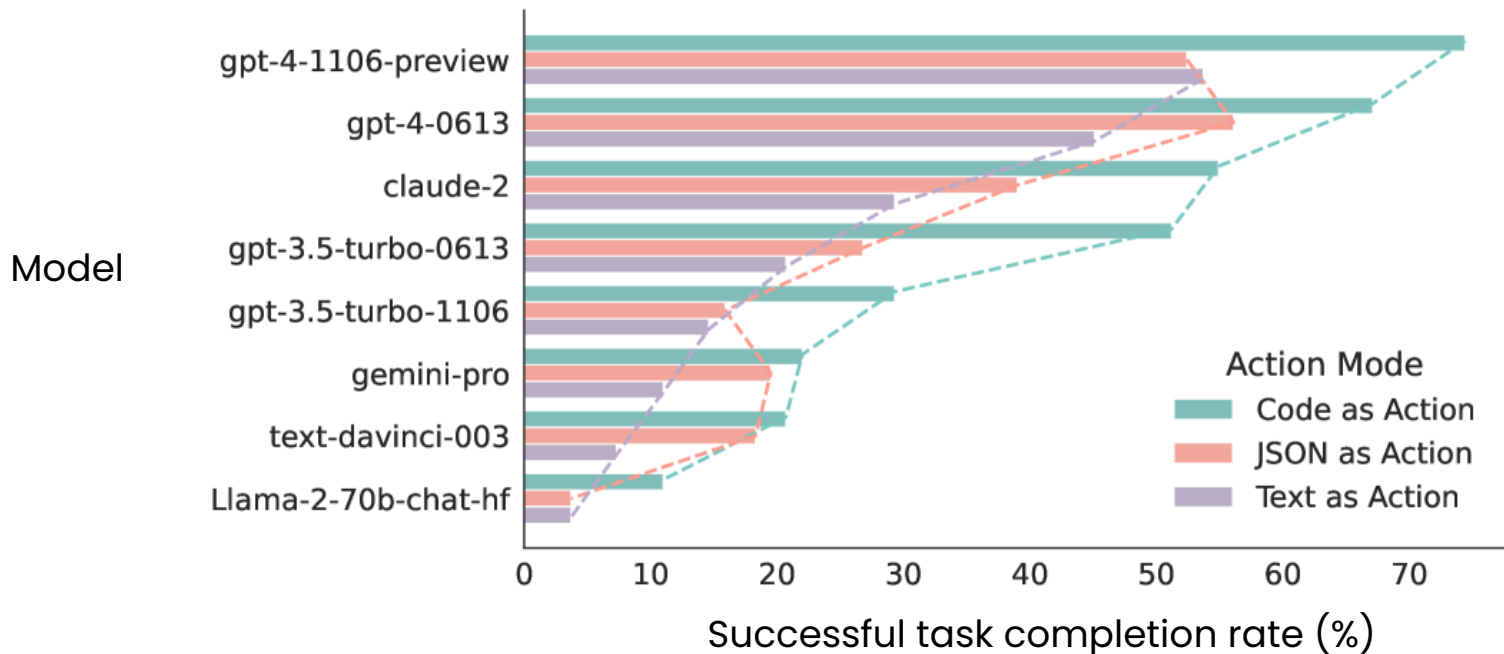
```
last_week = df[df["date"].between(week_ago, today)]
```

```
# Drop duplicate rows and count
```

```
print(last_week.drop_duplicates().shape[0])
```

```
</execute_python>
```

# Planning with code improves performance



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



# Patterns for highly autonomous agents

---

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

# 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

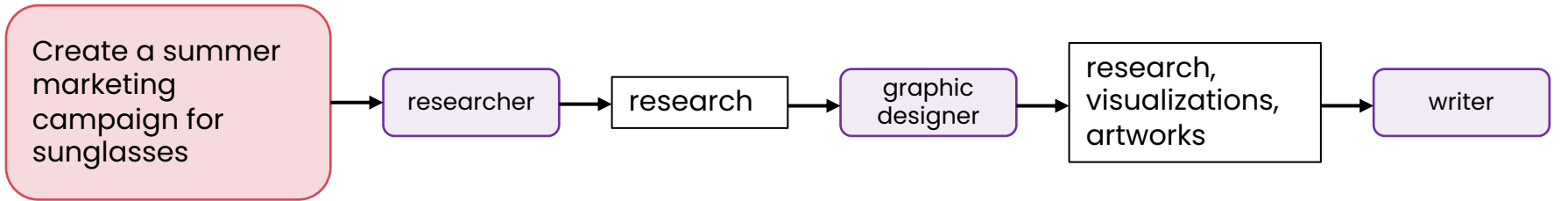
- Transform research into report text and marketing copy

### Tools

- (None)

writer

# Example: Marketing team with linear plan

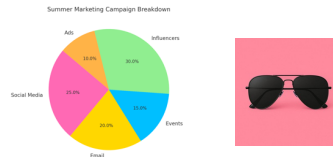


Here are current sunglasses trends and competitor offerings...

Here are 5 data visualizations and 5 artwork options for the report...



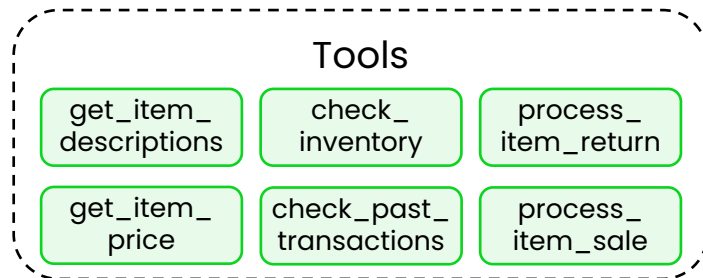
Final report



# Example: Planning with multiple agents

system  
prompt

You have access to the following tools:  
  
{description of tools}  
  
Return a step-by-step plan to carry out the user's request.



Create a summer marketing campaign for sunglasses

LLM

# Example: Planning with multiple agents

system  
prompt

You have access to the following  
tools:  
  
{description of tools}  
  
Return a step-by-step plan to carry  
out the user's request.

Agents

researcher

graphic  
designer

writer

Create a summer  
marketing  
campaign for  
sunglasses

LLM

# Example: Planning with multiple agents

system  
prompt

You are a marketing manager and  
have the following team of agents to  
work with:

{description of agents}

Return a step-by-step plan to carry  
out the user's request.

Agents

researcher

graphic  
designer

writer

Create a summer  
marketing  
campaign for  
sunglasses

LLM

# Example: Planning with multiple agents

system prompt

You are a marketing manager and have the following team of agents to work with:

{description of agents}

Return a step-by-step plan to carry out the user's request.

Create a summer marketing campaign for sunglasses

LLM

marketing manager

1. Ask **researcher** to research current sunglasses trends
2. Ask **graphic designer** to create ad images
3. Ask **writer** to create report
4. Review report

Agents

researcher

graphic designer

writer

Step 1 text

researcher

Step 1 output,  
Step 2 text

graphic designer

Step 2 output,  
Step 3 text

writer



# Patterns for highly autonomous agents

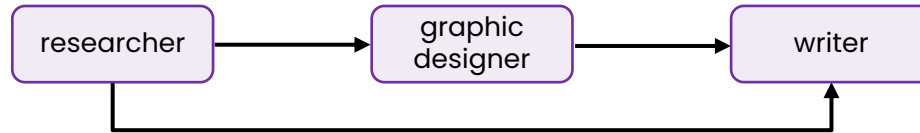
---

Communication patterns  
for multi-agent systems

# Example: Marketing team with linear plan

Create a summary marketing campaign for sunglasses

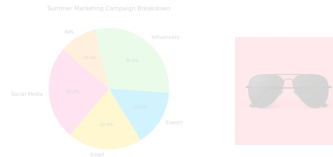
## Communication Pattern



writer



Final report



# Example: Planning with multiple agents

system prompt

You are a marketing manager and have the following team of agents to work with:

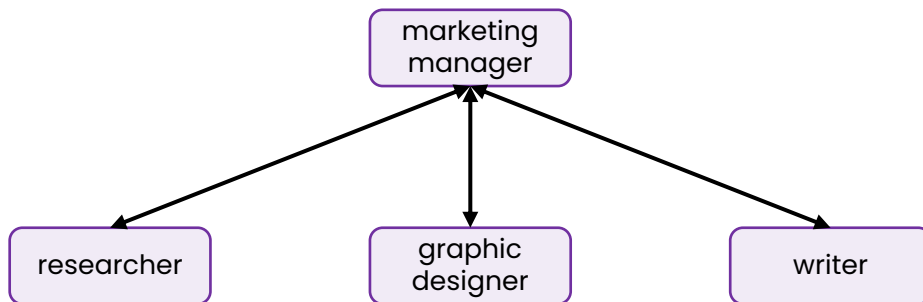
Agents

researcher

graphic designer

writer

## Communication Pattern



Create a summary marketing campaign for sunglasses

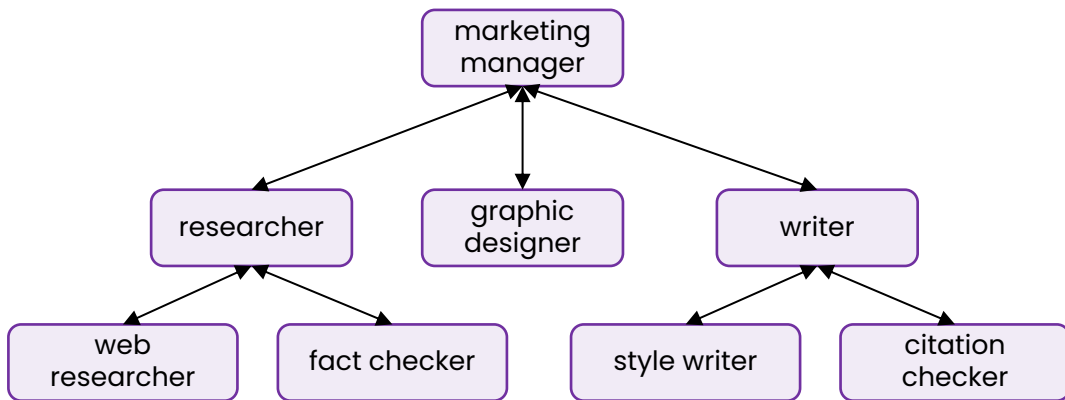
3. Ask **writer** to create report
4. Review report

Step 2 output, Step 3 text

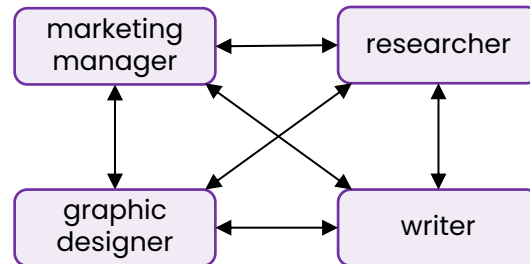
writer

# Other communication patterns

Deeper hierarchy



All-to-all





# Agentic AI

---

# Conclusion

# Summary

- Why Agentic AI
- Reflection design pattern
- Tool use (function calling)
- Evals, error analysis
- Planning, multi-agent systems



# End of M5

---