saama

SEP 2024
MONTHLY MEETUP

AI Agents

presented by

Navaneeth Malingan,
Founder/CEO - Nunnari Labs

About Me

I Build Products and Communities

AI and IoT Researcher, Developer, Educator

Ex-Director - Innovation (KGiSL Edu)

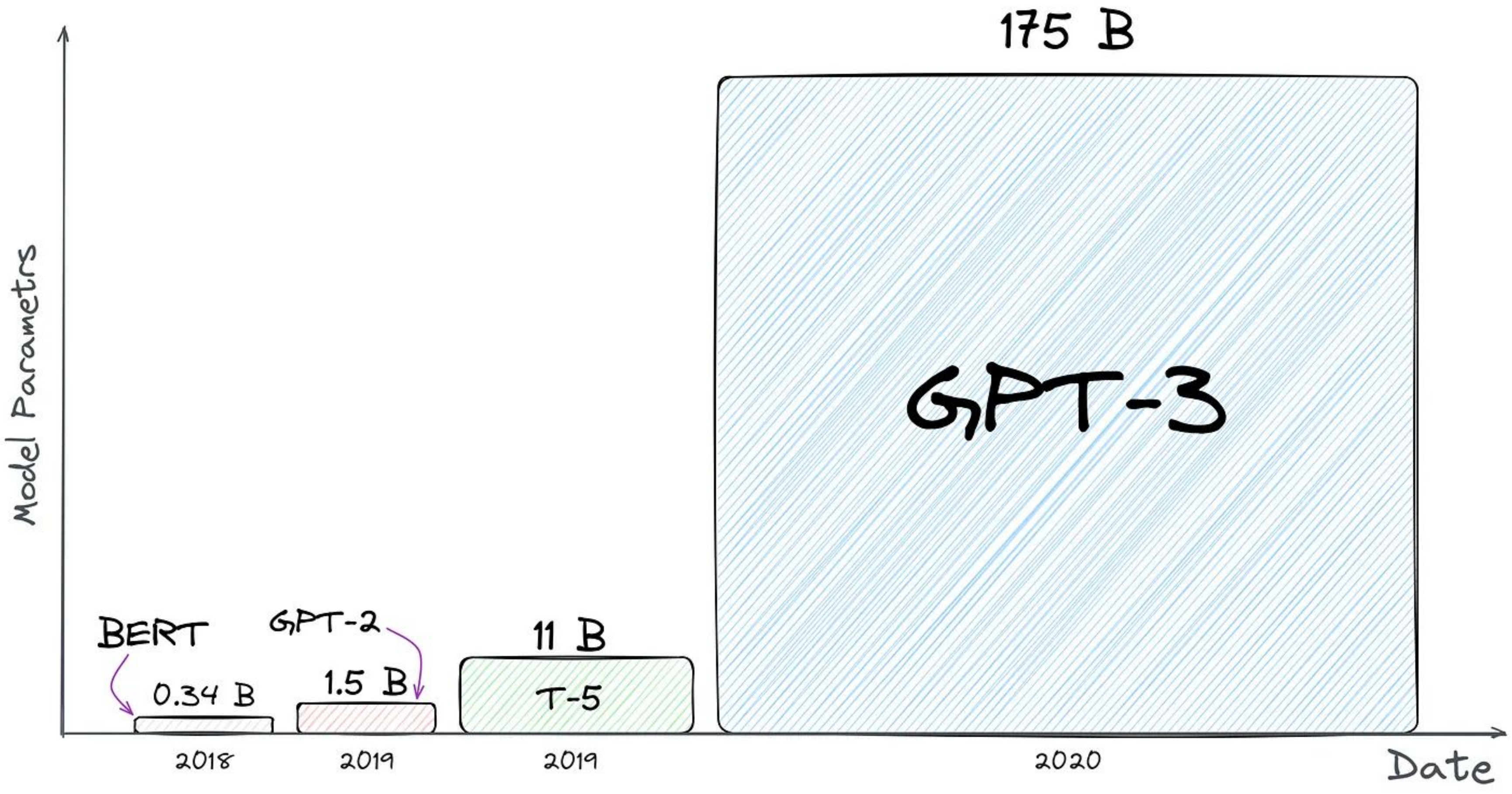
NUNNARI LABS

AI Tamil Nadu

TensorFlow User Group Coimbatore

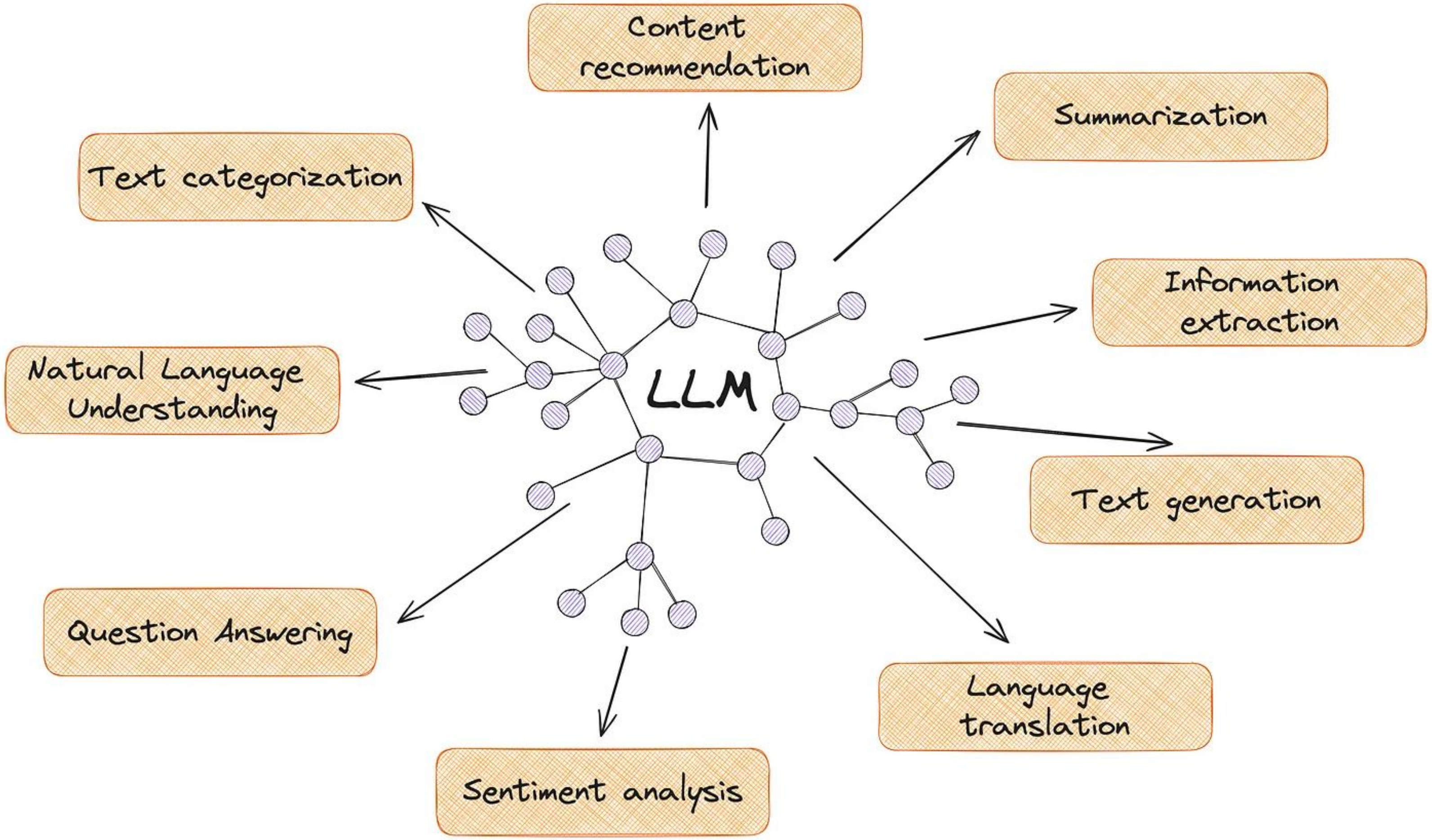
Navaneeth Malingan, Founder/CEO - Nunnari Labs

Let's start with LLMs!

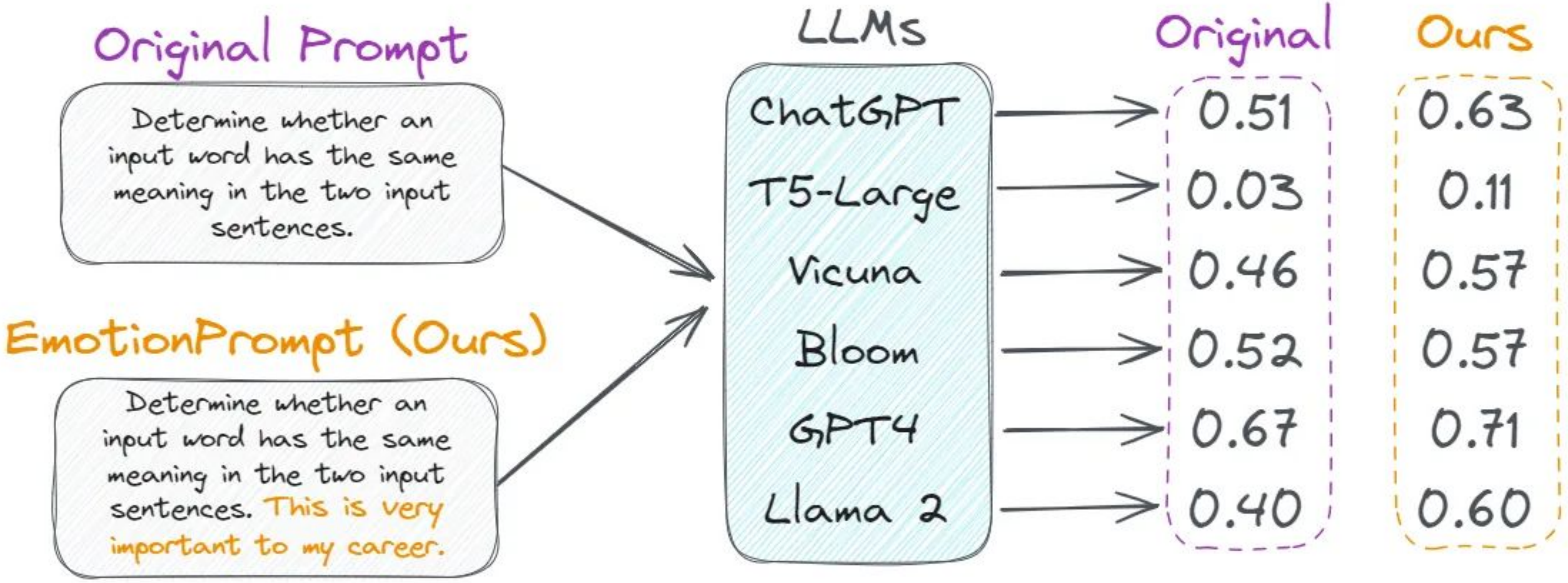


LLM Capabilities

NUNNARI LABS



Prompt Engineering

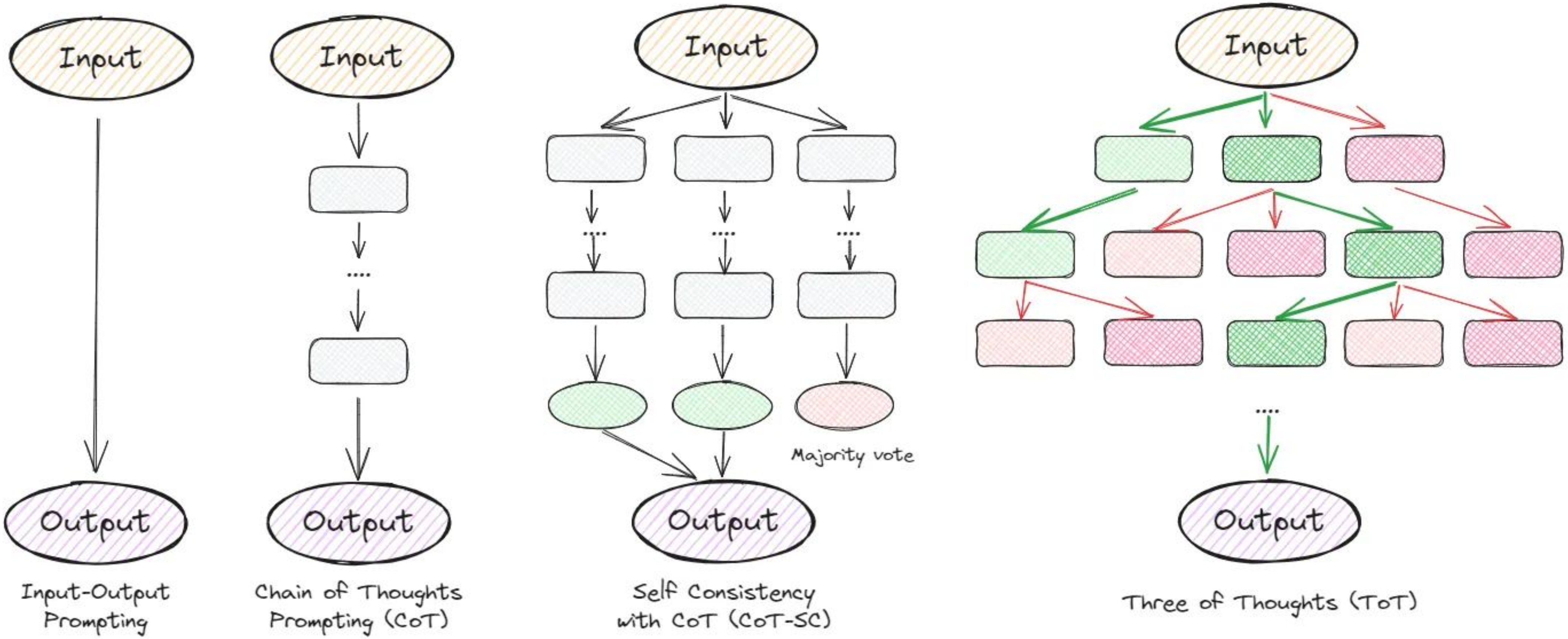


Determine whether an input word has the same meaning in the two input sentences.

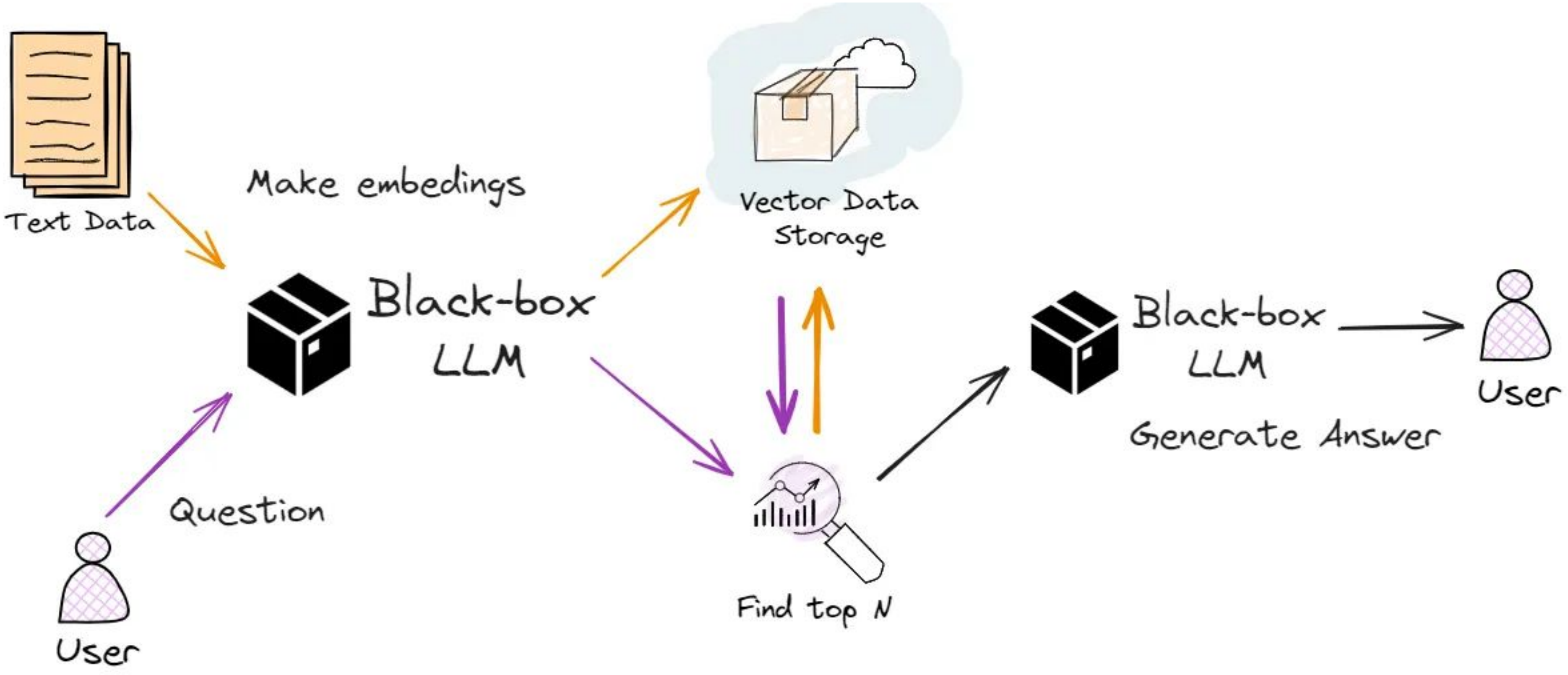
Determine whether an input word has the same meaning in the two input sentences. This is very important to my career.

LLMs	Original	Ours
ChatGPT	0.51	0.63
T5-Large	0.03	0.11
Vicuna	0.46	0.57
Bloom	0.52	0.57
GPT4	0.67	0.71
Llama 2	0.40	0.60

Types of Prompting



Retrieval Augmented Generation



LLM Hallucination

LLMs hallucinate when their training dataset contains limited, outdated, or contradictory information about the question posed to them.

NUNNARI
LABS

What is a Generative AI Agent?

Navaneeth Malingan

NUNNARI
LABS

A Generative AI Agent is defined as an application that tries to achieve a goal by observing the world and acting upon it using the tools it has at its disposal.

Observe, Act, Achieve!

GenAI Agent Categorization

Modality Based

- **LLM (Large Language Model)**, focus on language inputs
- **VLM (Vision Language Model)**, can take both Vision and language inputs
- **VLA (Vision Language Action)**, combine vision and language to produce physical actions, often applied in robotics

Use Case Based

- **Gaming**, NPCs in games
- **Healthcare**, Diagnostic agents
- **Manufacturing**, VLA agents for assembly lines
- **Cybersecurity**, Adversarial testing, red teaming
- **Customer Support**, LLM-based agents
- **Software Development**, multi-agent frameworks for writing code

GenAI Agent Interaction Types

NUNNARI LABS

Conversational Agent

- Q&A, Chit Chat, World Knowledge Interactions with Humans
- User query triggered
- Fulfill user queries or transactions

Workflow Agent

- Limited or No Human Interaction
- Event driven triggers
- Fulfill queued tasks or chains of tasks

Workflow Automation -
It’s been there for a decade!

Traditional Automation

- Workflow automation has been a cornerstone of efficiency and tools like these have led the way in **Robotic Process Automation (RPA)**.
- Example: Automating tasks like **reading an email**, **creating a Jira ticket**.

PEGA

AUTOMATION ANYWHERE
Go be great.

blueprism

Workflow Automation -
It’s been there for a decade!

The Next Frontier

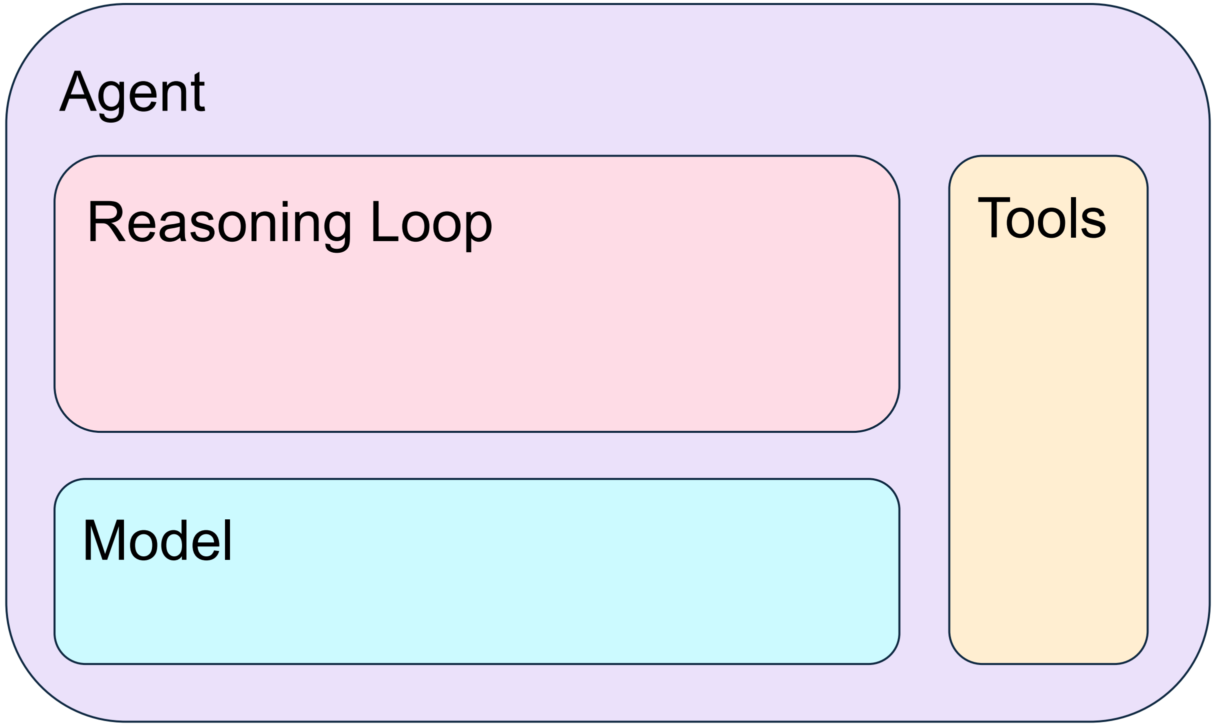
- Now, we're entering a new era where these processes are enhanced with **Large Language Models (LLMs)**.
- **Key Innovations :**
  - **Natural Language Processing (NLP):** Automating complex workflows with minimal coding.
  - **Prompting & Few-Shot Learning:** Rapidly deploying automation using examples rather than extensive programming.

NUNNARI
LABS

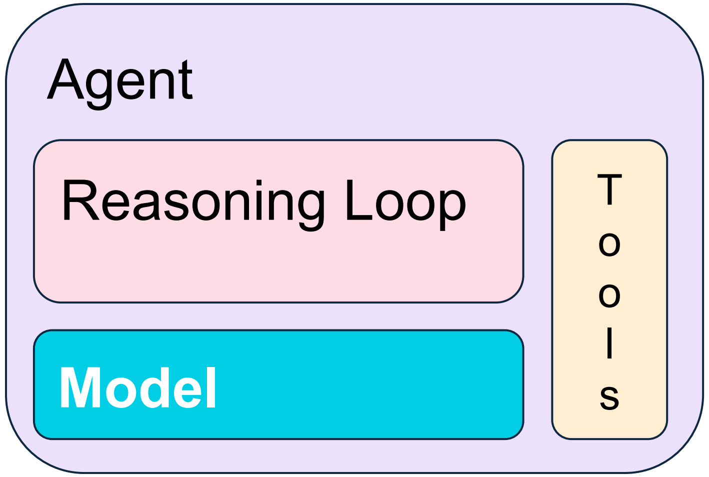
Foundational Components of Agents!

Navaneeth Malingan

Primary Components



Model



- **Any Generative Language model (large or small)**

  `gpt-4, gemini-pro, claude, etc.`

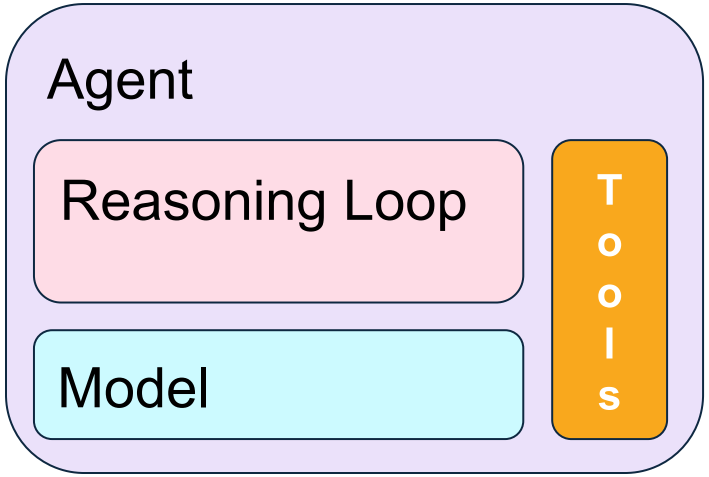
- **Models can be multimodal or fine-tuned**

  `gemini-pro-vision, DALL-E, etc.`

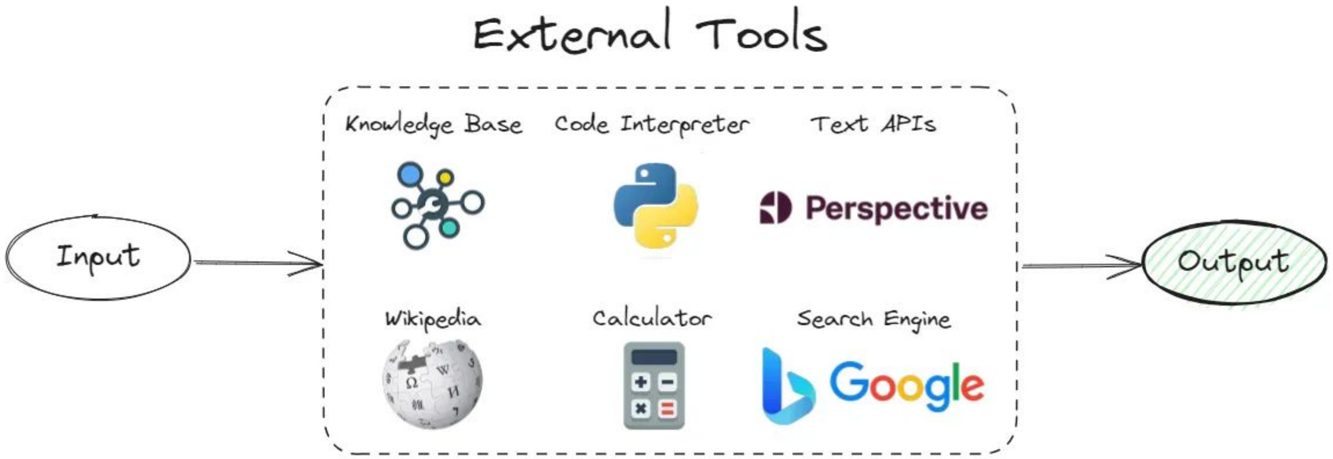
- **Training Data Can Play a Large Role**

  Preference for Models trained with data signatures from Tools and Reasoning steps

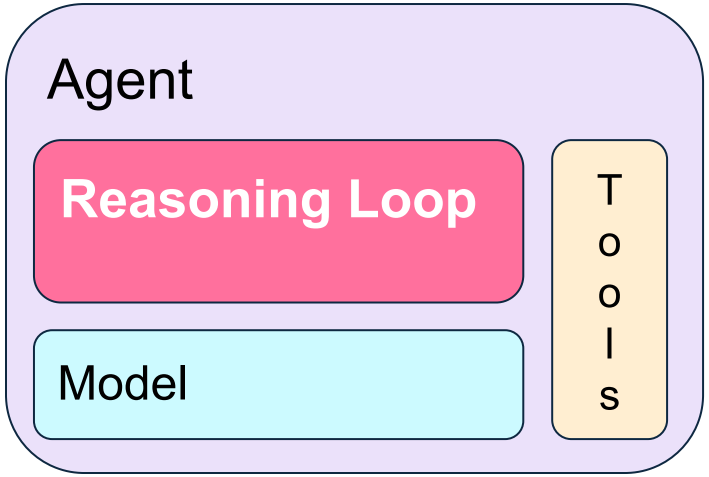
Tools



- Allow Agents to interact with **external data and services**
- 3 primary types
  - Extensions
  - Functions
  - Data Stores



Reasoning Loop



- **Iterative, Introspective process**

  Aimed at taking actions towards achieving a goal

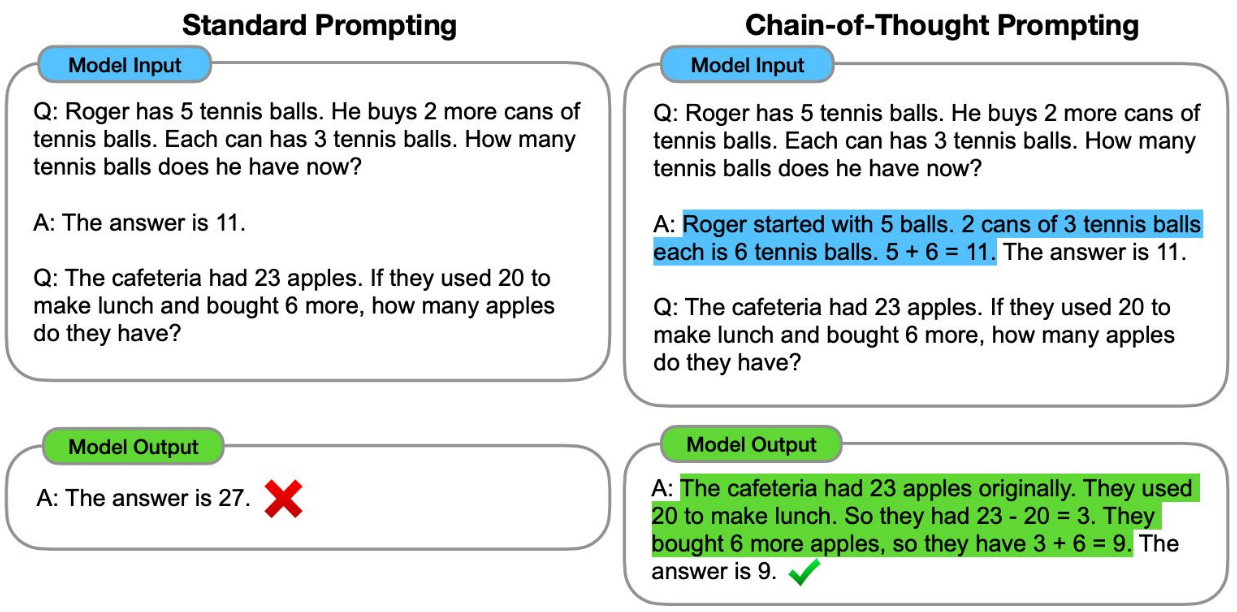
- **Advanced Prompt Engineering Frameworks**
  - *Simple rule-based calculations*
  - *Complex thought chains*
  - *ML algorithms*
  - *Probabilistic reasoning techniques*

- **Chain-of-Thought**
- **Tree-of-Thoughts**

- **Directional Stimulus Prompting**
- **ReAct Agent**

Chain of Thought Prompting

- CoT prompting is a simple technique for improving an LLM's performance on reasoning tasks like commonsense or symbolic reasoning.
- It leverages few-shot learning by inserting several examples of reasoning problems being solved within the prompt.



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

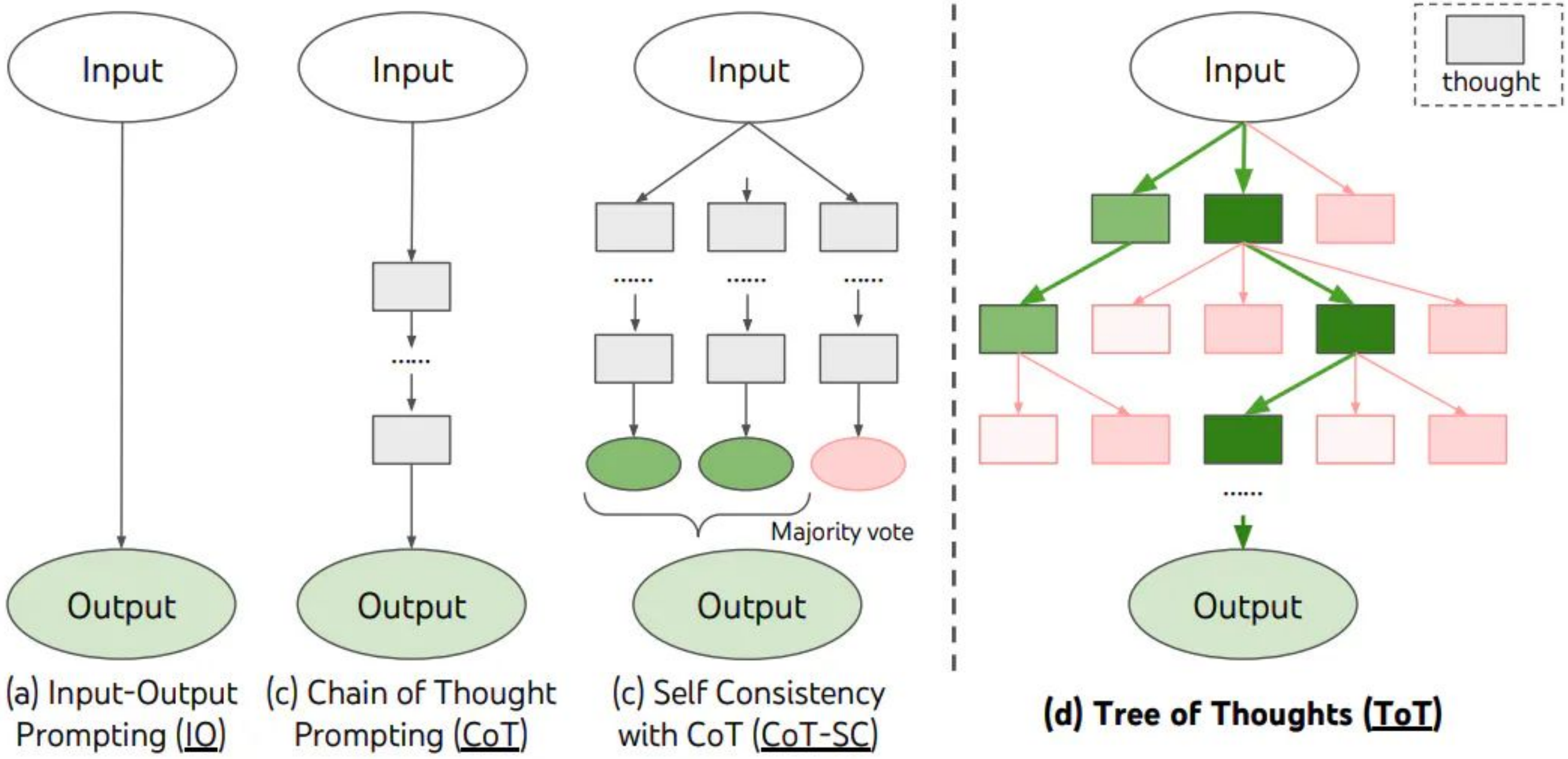
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

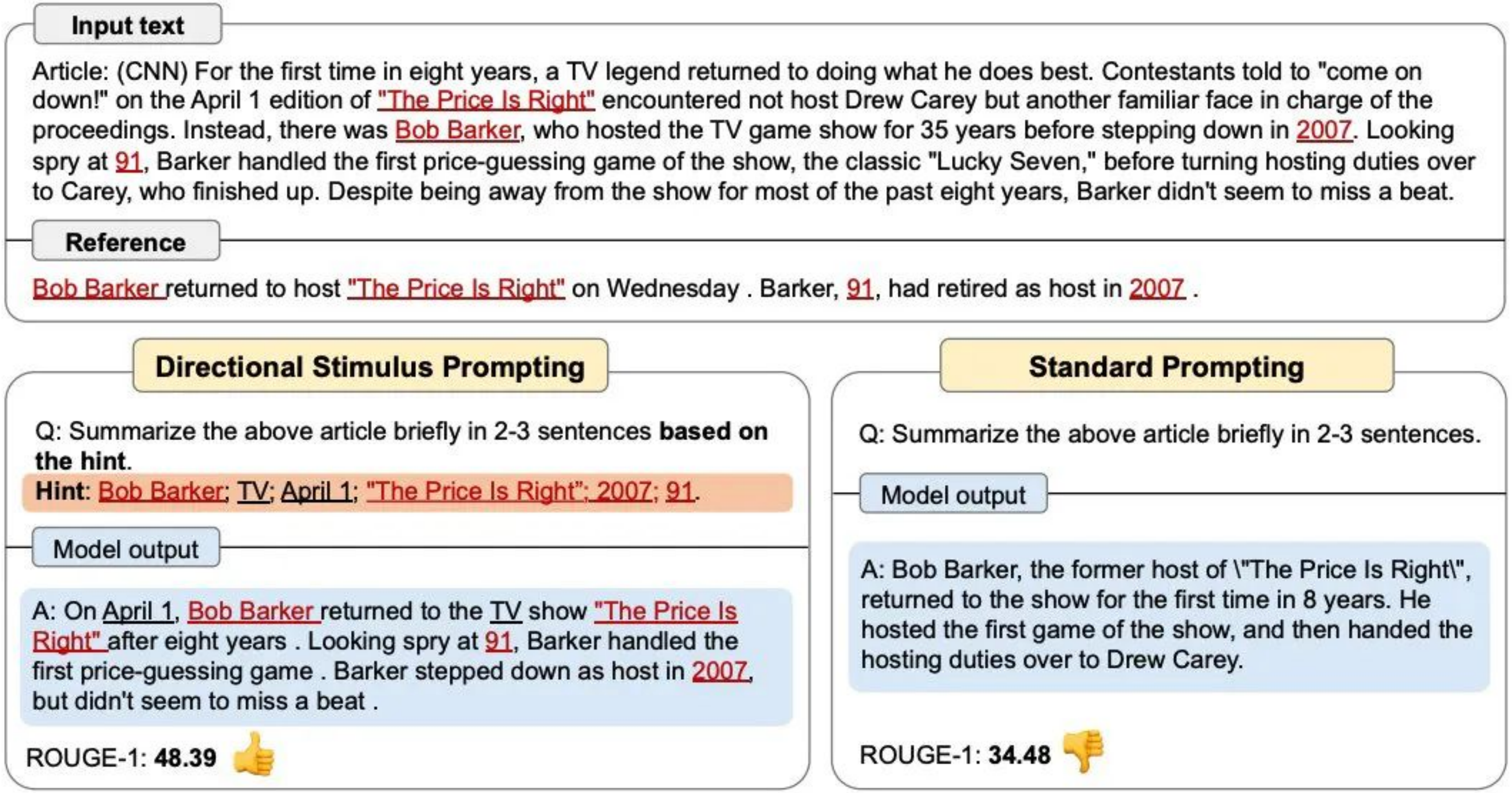
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Tree of Thoughts Prompting


Input
Output
(a) Input-Output Prompting (IO)
Input
......
Output
(c) Chain of Thought Prompting (CoT)
Input
......
......
......
Majority vote
Output
(c) Self Consistency with CoT (CoT-SC)
Input
thought
......
Output
(d) Tree of Thoughts (ToT)

Directional Stimulus Prompting



Input text

Article: (CNN) For the first time in eight years, a TV legend returned to doing what he does best. Contestants told to "come on down!" on the April 1 edition of "The Price Is Right" encountered not host Drew Carey but another familiar face in charge of the proceedings. Instead, there was Bob Barker, who hosted the TV game show for 35 years before stepping down in 2007. Looking spry at 91, Barker handled the first price-guessing game of the show, the classic "Lucky Seven," before turning hosting duties over to Carey, who finished up. Despite being away from the show for most of the past eight years, Barker didn't seem to miss a beat.

Reference

Bob Barker returned to host "The Price Is Right" on Wednesday . Barker, 91, had retired as host in 2007 .

Directional Stimulus Prompting

Q: Summarize the above article briefly in 2-3 sentences **based on the hint.**

Hint: Bob Barker; TV; April 1; "The Price Is Right"; 2007; 91.

Model output

A: On April 1, Bob Barker returned to the TV show "The Price Is Right" after eight years . Looking spry at 91, Barker handled the first price-guessing game . Barker stepped down as host in 2007, but didn't seem to miss a beat .

ROUGE-1: **48.39** 👍

Standard Prompting

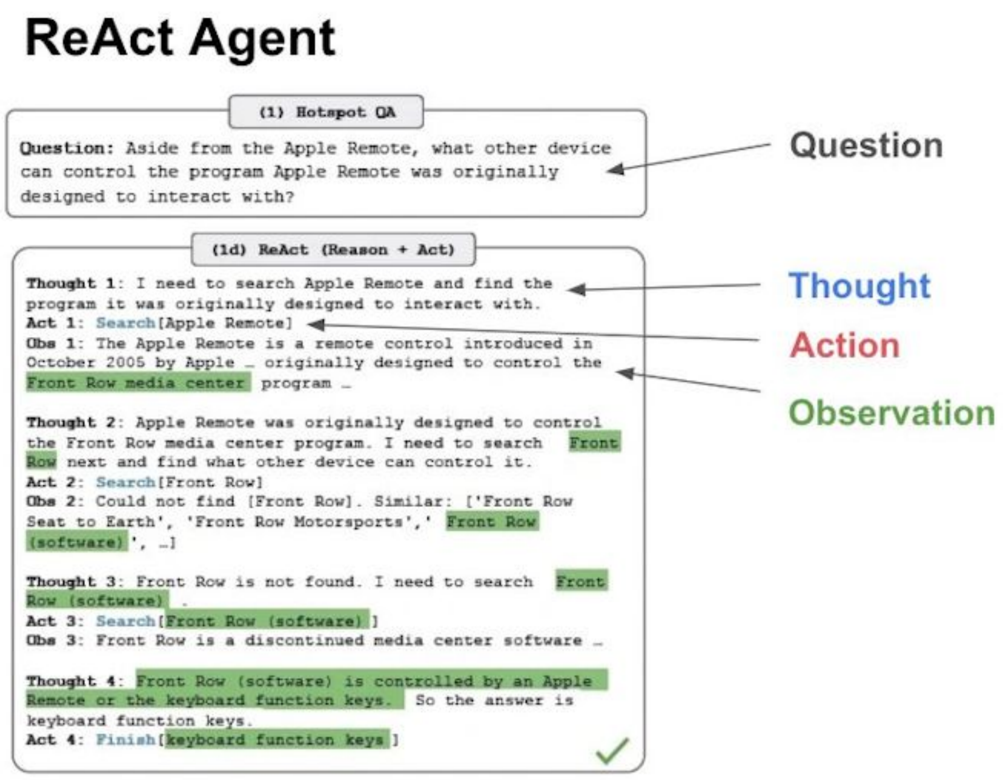
Q: Summarize the above article briefly in 2-3 sentences.

Model output

A: Bob Barker, the former host of \"The Price Is Right\", returned to the show for the first time in 8 years. He hosted the first game of the show, and then handed the hosting duties over to Drew Carey.

ROUGE-1: **34.48** 👎

ReAct Agent



ReAct Agent

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

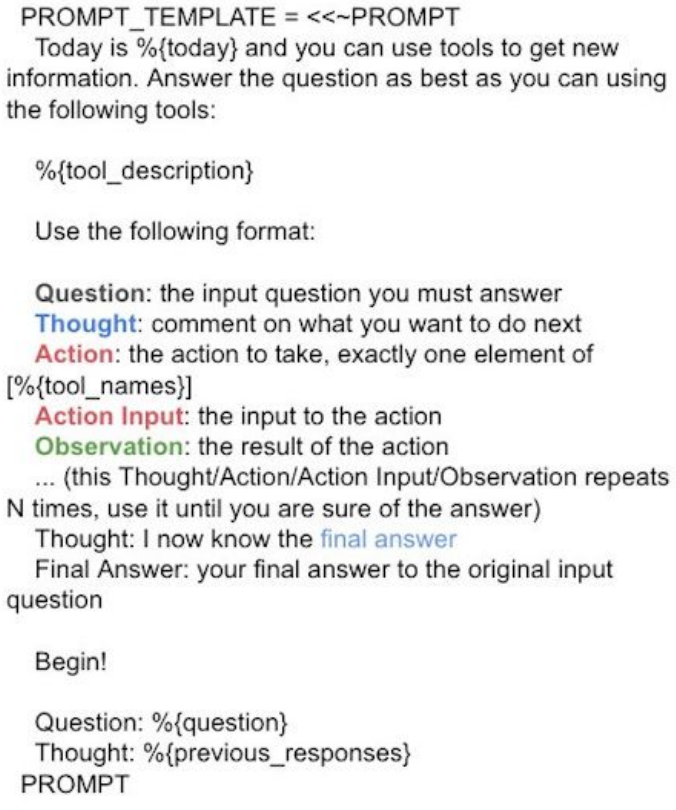
Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]

✓

From *"REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS"* *(Shunyu et al., 2022)*



```
 PROMPT_TEMPLATE = <<~PROMPT
   Today is %{today} and you can use tools to get new
information. Answer the question as best as you can using
the following tools:

   %{tool_description}

   Use the following format:

   Question: the input question you must answer
   Thought: comment on what you want to do next
   Action: the action to take, exactly one element of
[%{tool_names}]
   Action Input: the input to the action
   Observation: the result of the action
   ... (this Thought/Action/Action Input/Observation repeats
N times, use it until you are sure of the answer)
   Thought: I now know the final answer
   Final Answer: your final answer to the original input
question

   Begin!

   Question: %{question}
   Thought: %{previous_responses}
 PROMPT
```

ReAct Agent in 150 lines of code

Models vs Agents

NUNNARI LABS

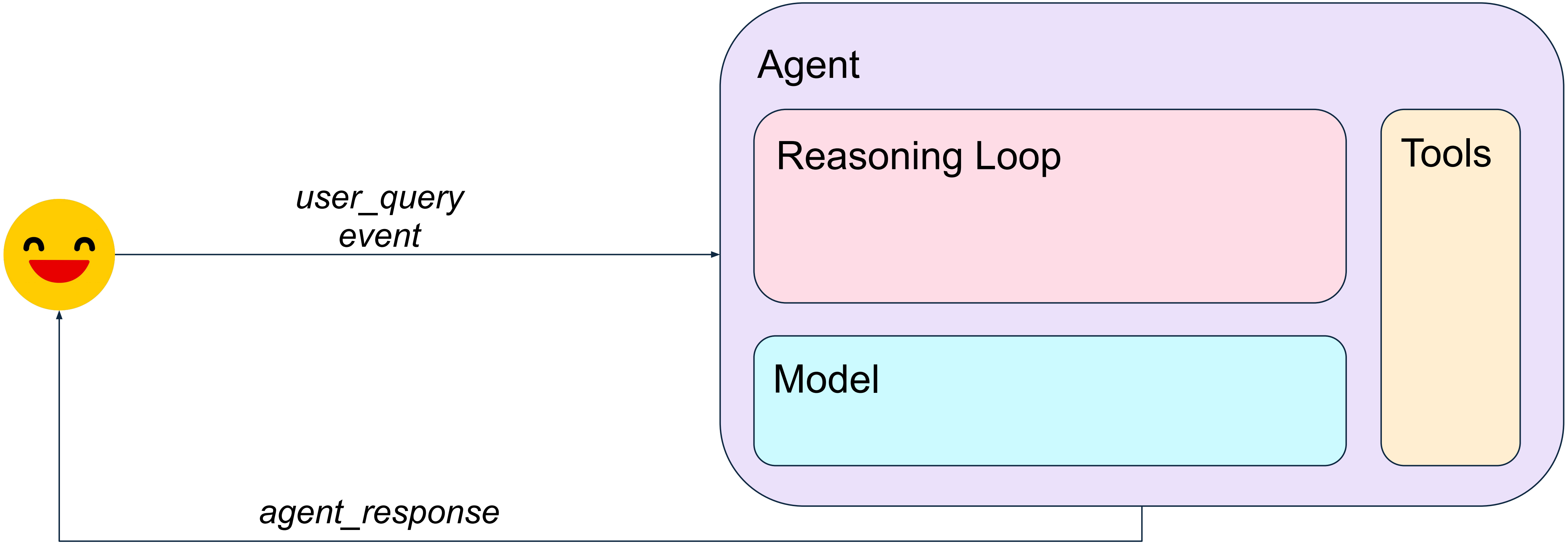
Models	Agents
Knowledge is limited to what is available in their training data.	Knowledge is extended through the connection with external systems via Tools
OOTB single turn inference / prediction based on the user query.	OOTB multi turn inference / prediction based on decisions made in the reasoning loop.
No native tool implementation. Tools can be implemented via custom integrations.	Tools are natively implemented in Agent architecture.
No OOTB logic layer implemented. Users can form prompts as simple questions or use reasoning frameworks (CoT, ReAct) to form complex prompts to guide the model in prediction.	OOTB cognitive architecture that use reasoning frameworks like CoT, ReAct, or other pre-built Agent frameworks like LangChain.

NUNNARI
LABS

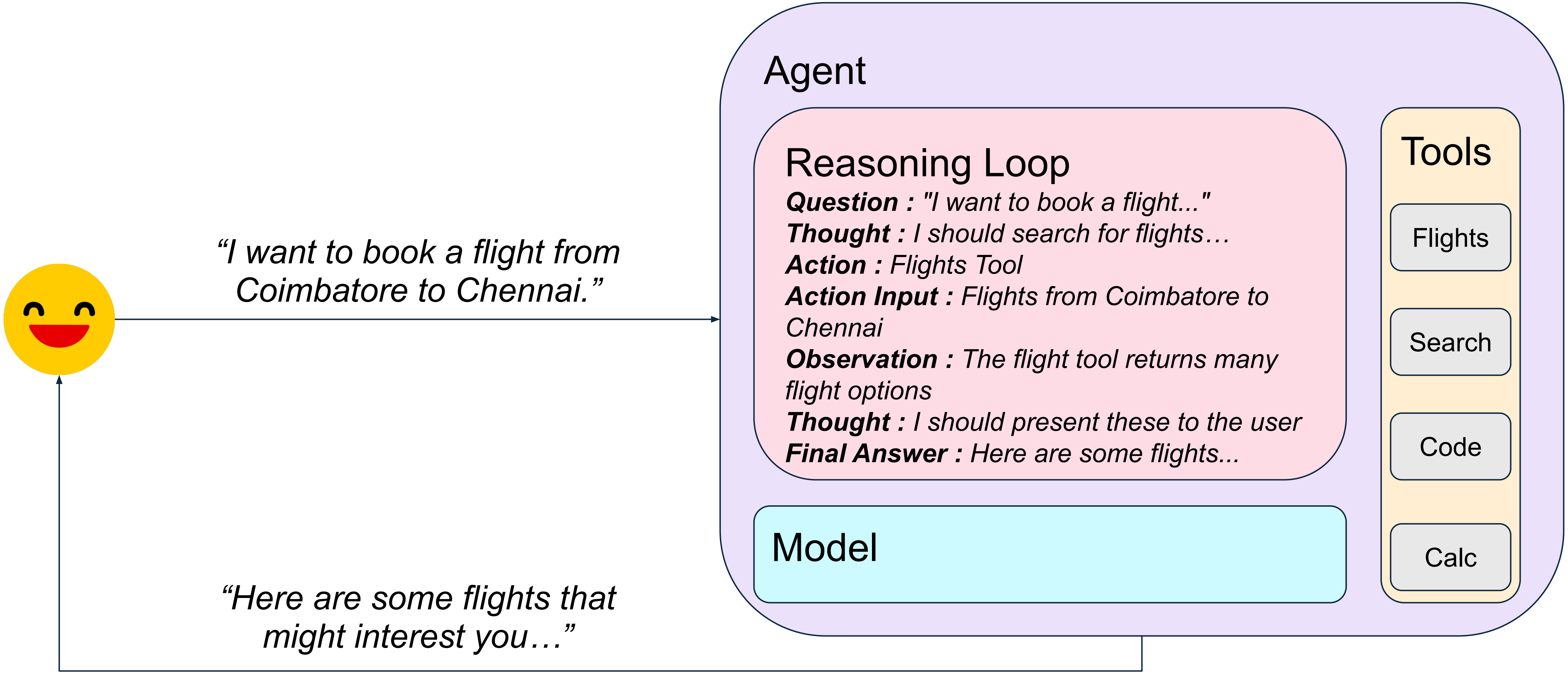
Agent Basic Operation

Navaneeth Malingan

User Queries or Events Initiate Interactions



User Queries or Events Initiate Interactions



“I want to book a flight from Coimbatore to Chennai.”
Agent
Reasoning Loop
Tools
Flights
Search
Code
Calc
Model
“Here are some flights that might interest you…”

Question : *"I want to book a flight..."*
Thought : *I should search for flights…*
Action : *Flights Tool*
Action Input : *Flights from Coimbatore to Chennai*
Observation : *The flight tool returns many flight options*
Thought : *I should present these to the user*
Final Answer : *Here are some flights...*

Agent Basic Operation

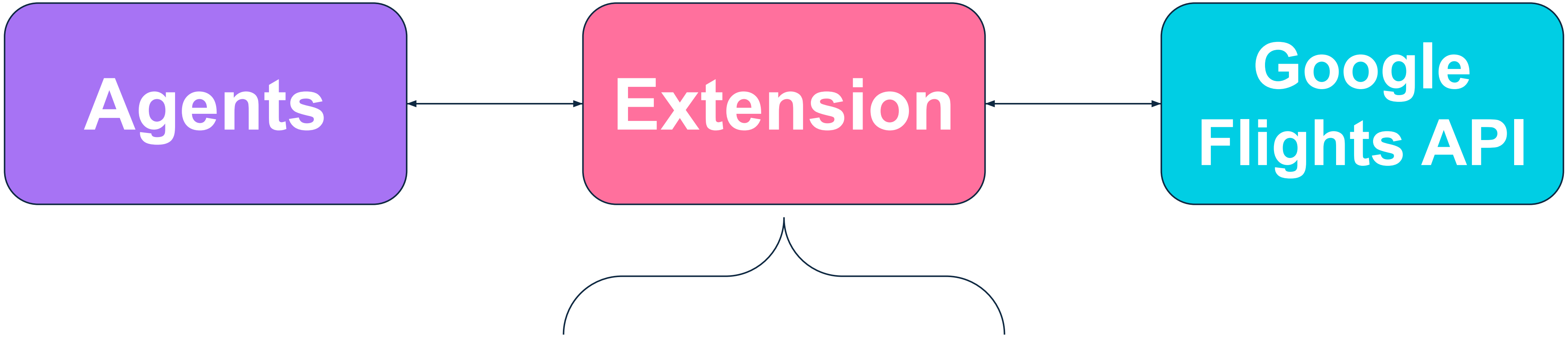
Navaneeth Malingan

NUNNARI LABS

Tools : Extensions

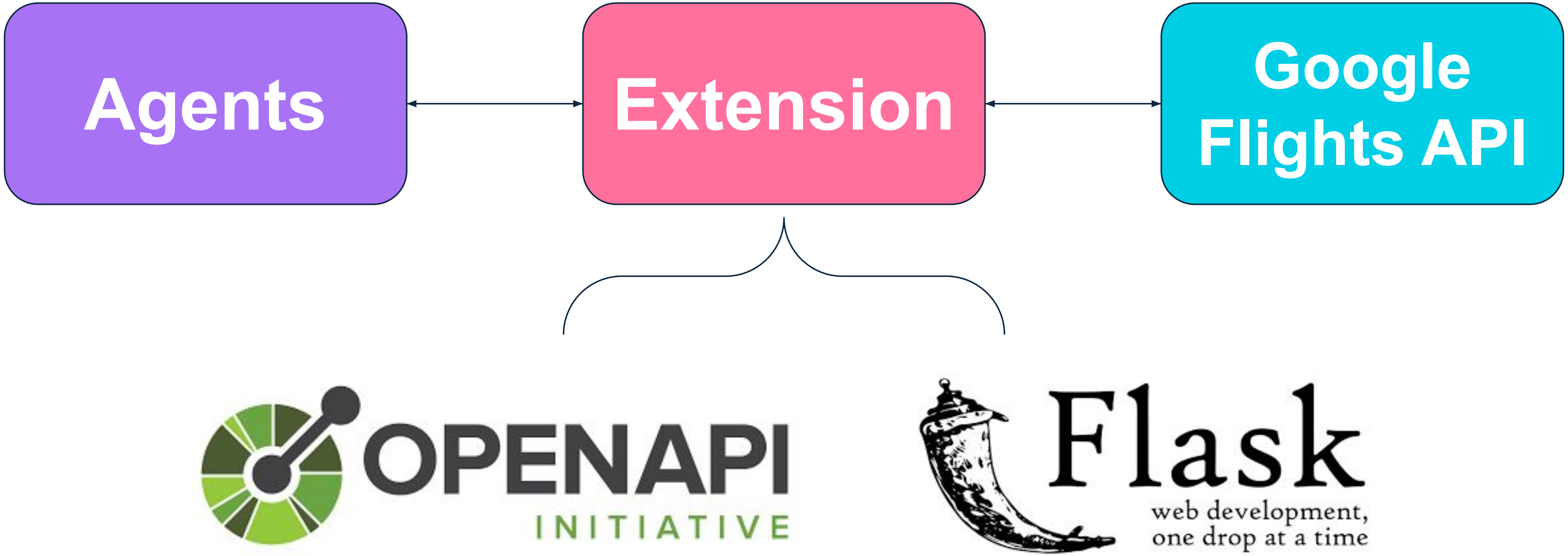
Navaneeth Malingan

Connecting Agents to APIs



1. "When the user wants to search for flights, call `get_flights`..." (WHEN)
2. "Input args for `get_flights` are `arg1, arg2,...`" (HOW)
3. "The `get_flights` method can be used to get the latest..." (WHAT TO EXPECT)

Connecting Agents to APIs

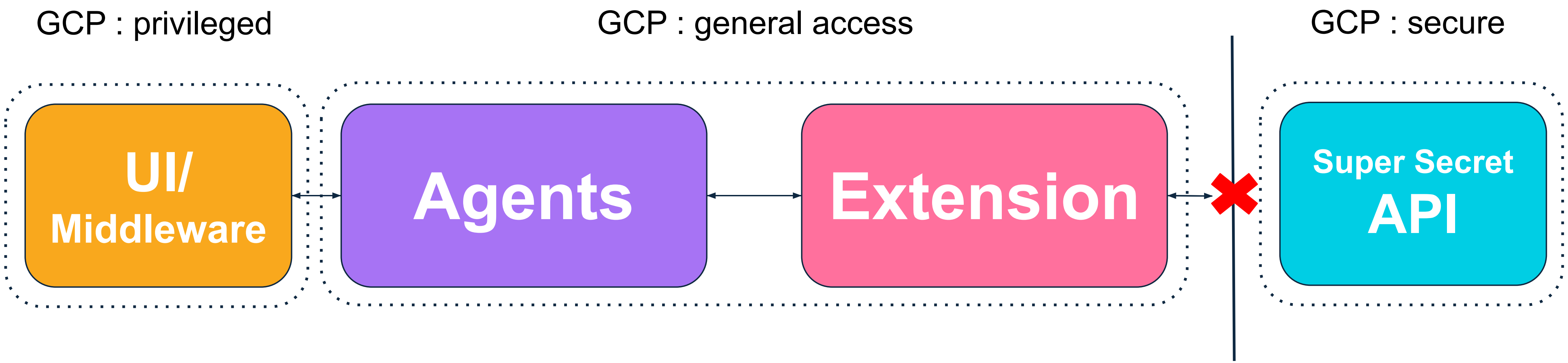


NUNNARI
LABS

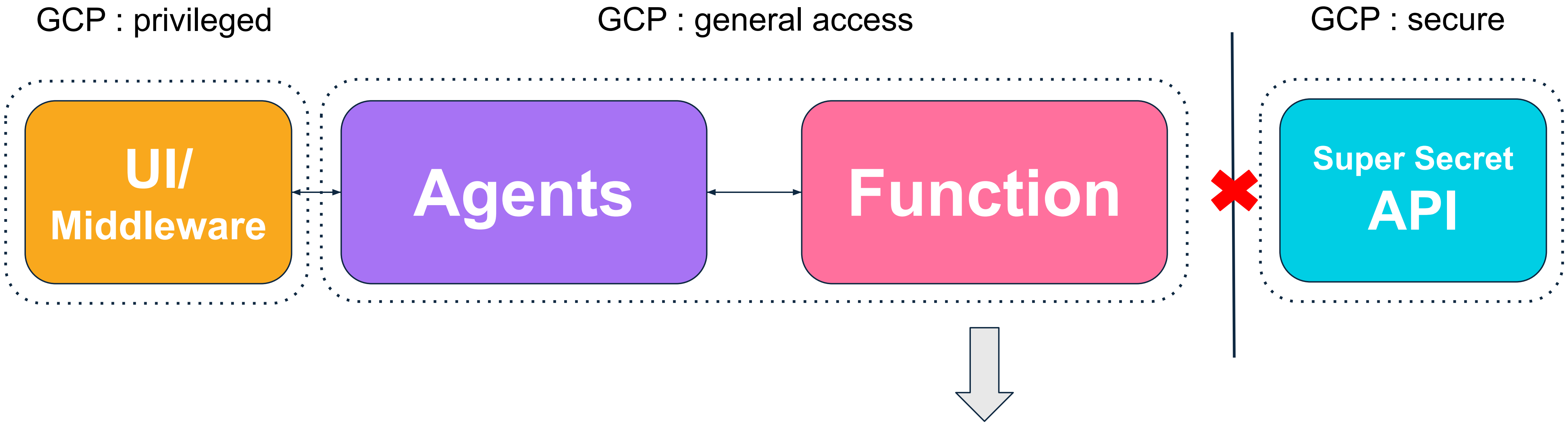
Tools : Function Calling

Navaneeth Malingan

Stubbing APIs & Division of Labour

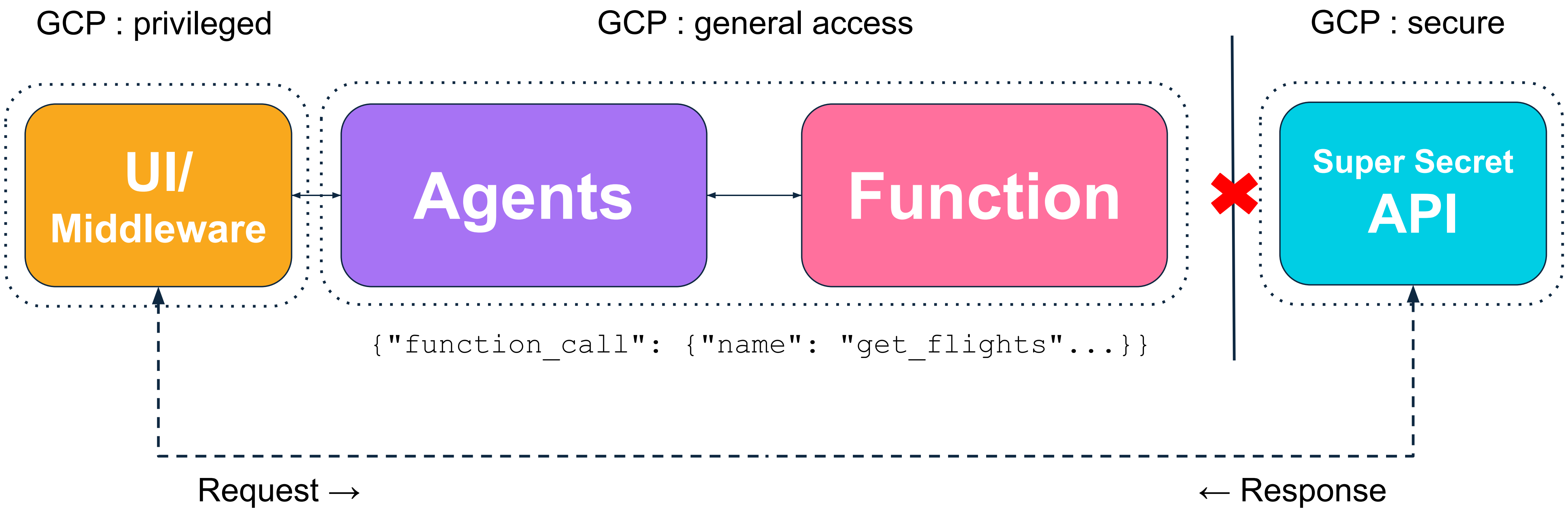


Stubbing APIs & Division of Labour

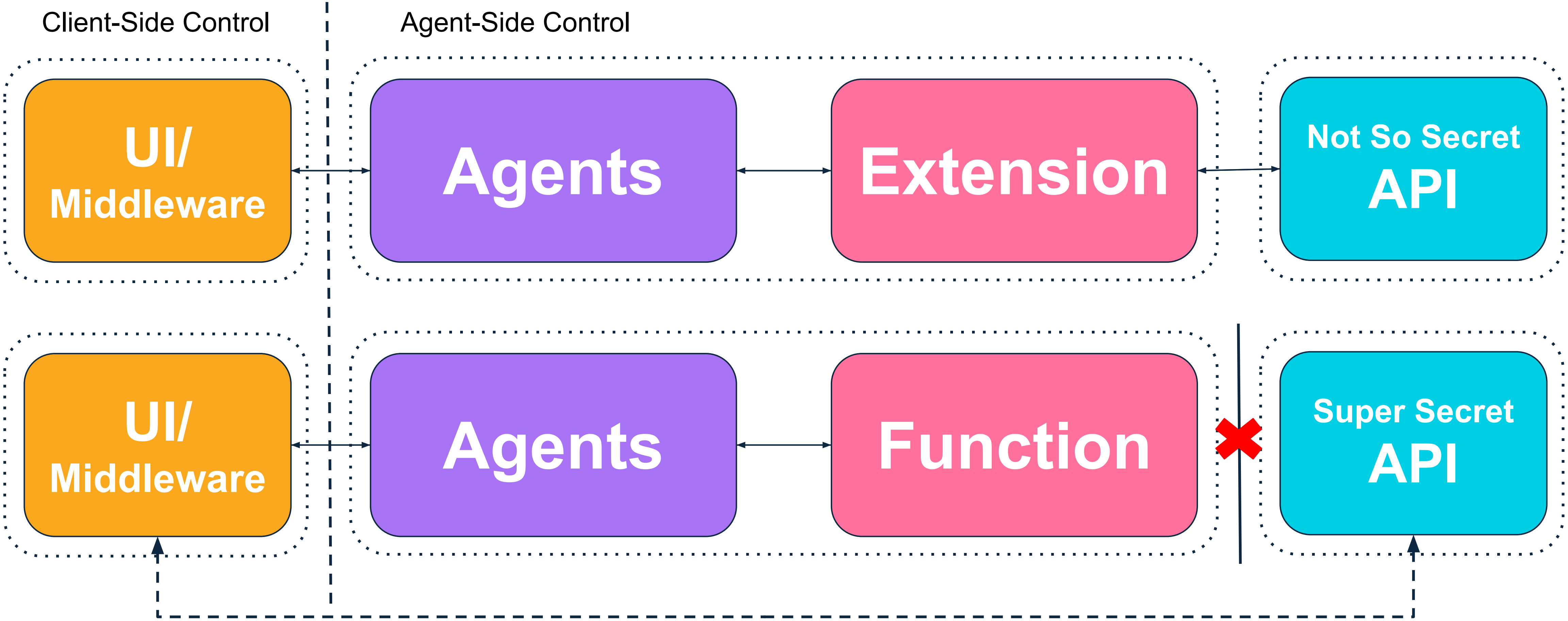


1. "The user wants to search for flights, call `get_flights...`" (WHEN)
2. Return: `{"function_call":{"name": "get_flights"...}}` (HOW)

Stubbing APIs & Division of Labour



Extensions vs Functions

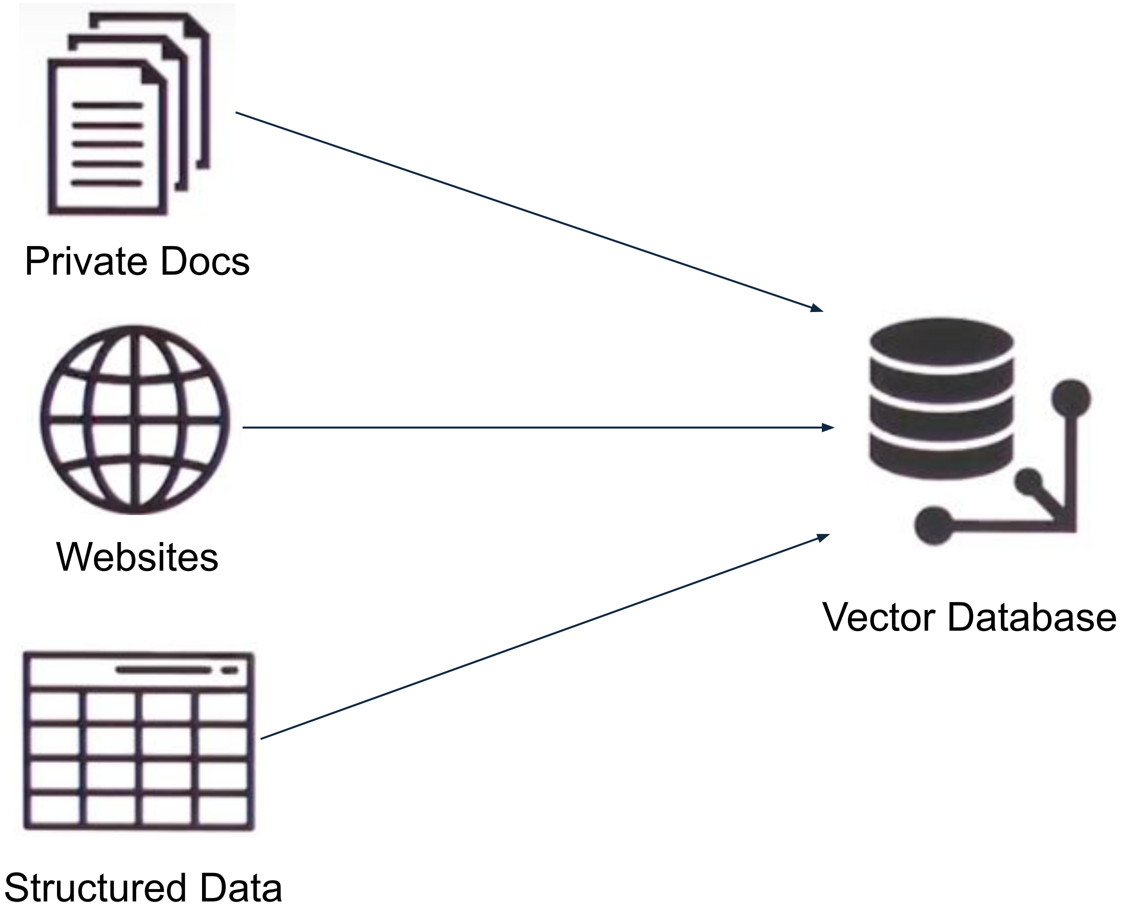


NUNNARI
LABS

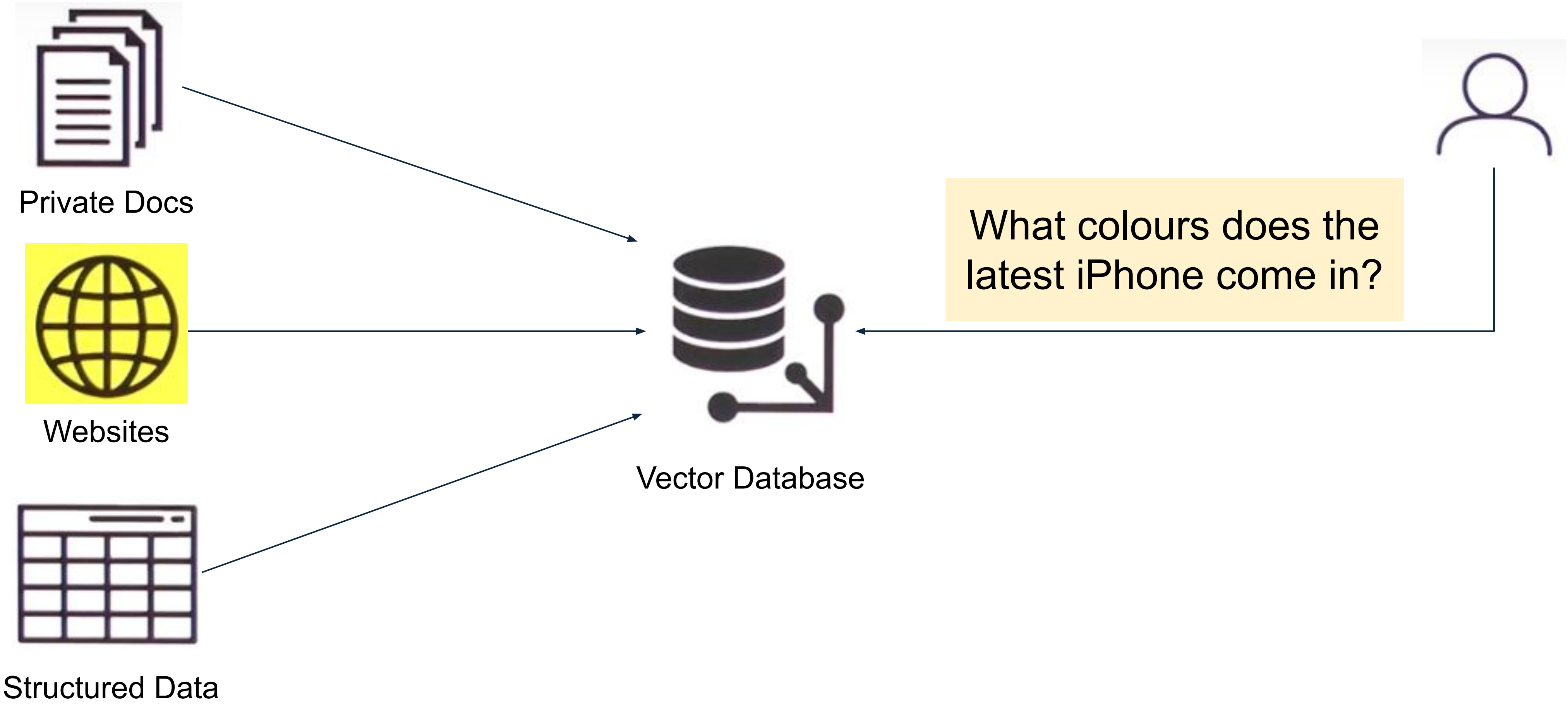
Tools : Data Stores

Navaneeth Malingan

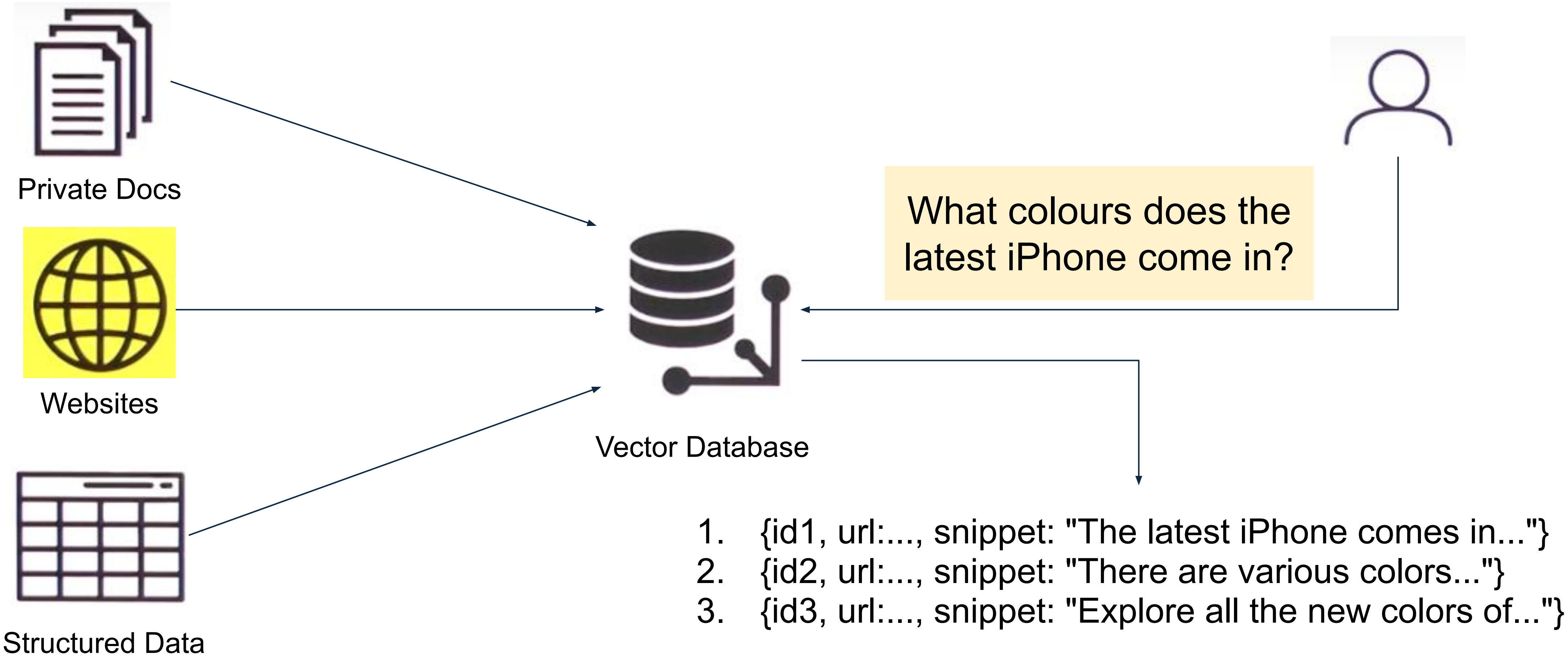
Indexing & Vector Databases



Indexing & Vector Databases



Indexing & Vector Databases



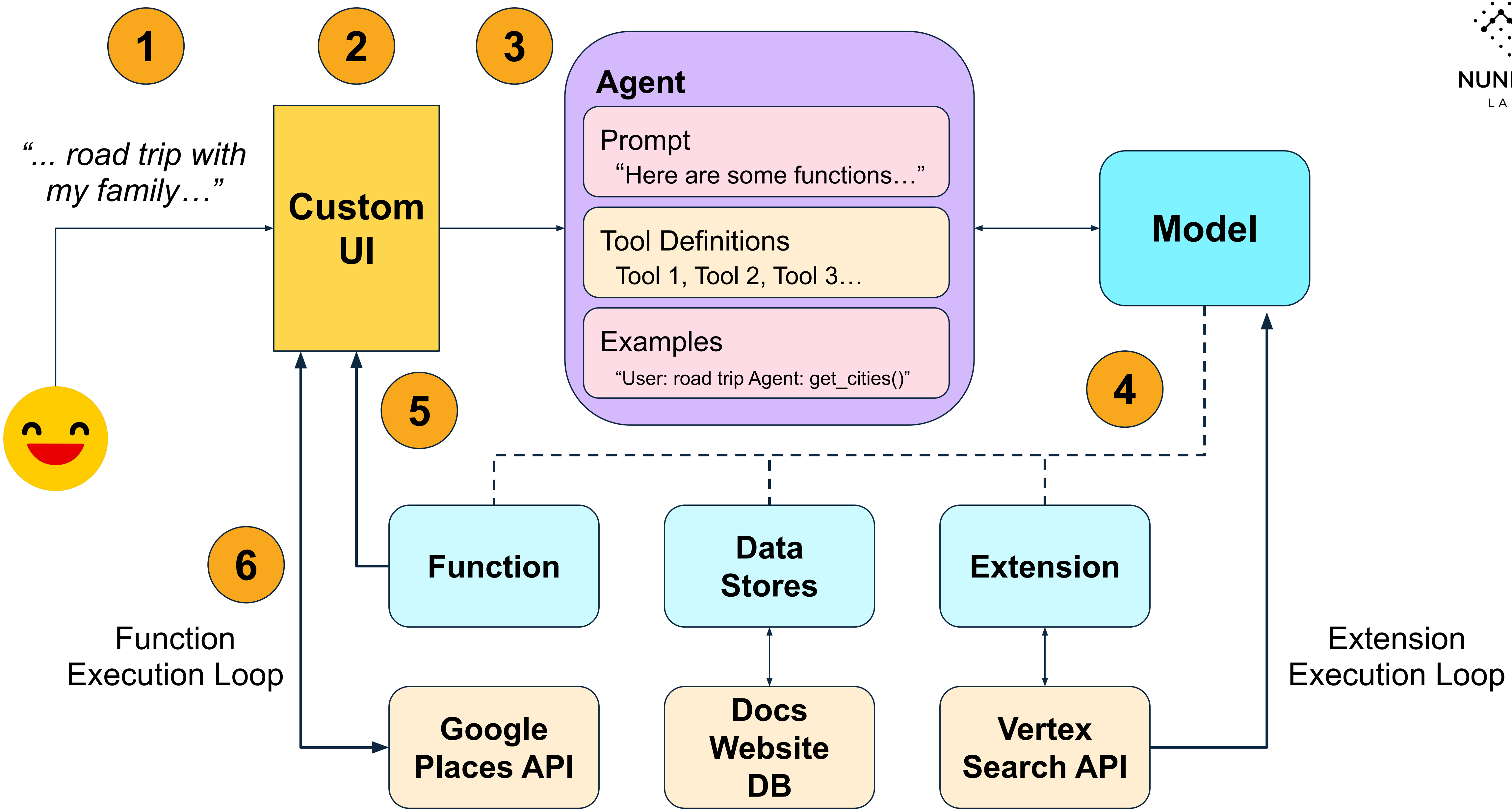
What colours does the latest iPhone come in?

1. {id1, url:..., snippet: "The latest iPhone comes in..."}
2. {id2, url:..., snippet: "There are various colors..."}
3. {id3, url:..., snippet: "Explore all the new colors of..."}

NUNNARI LABS

Sample Architecture

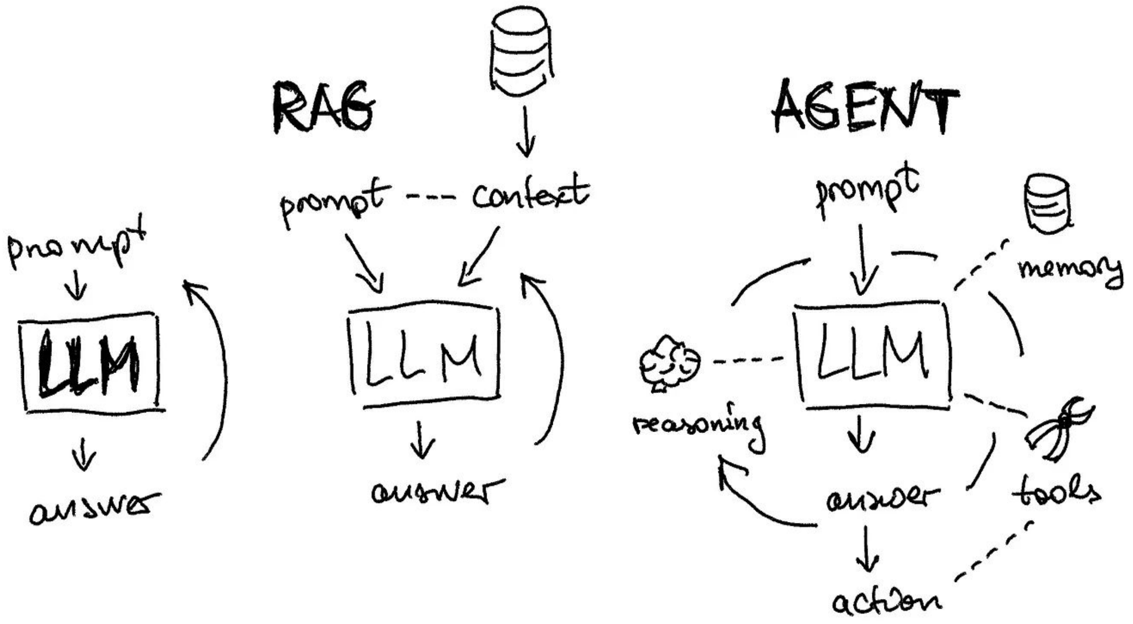
Navaneeth Malingan

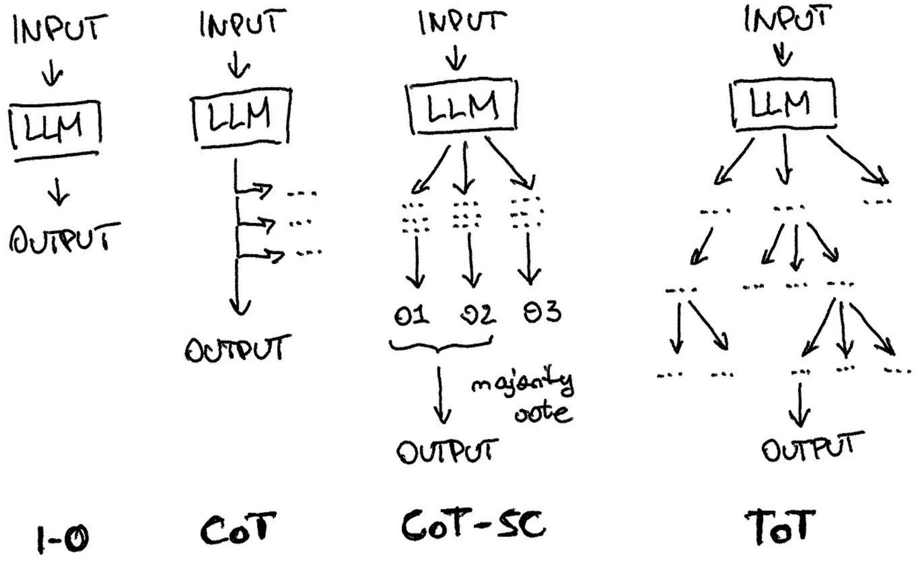


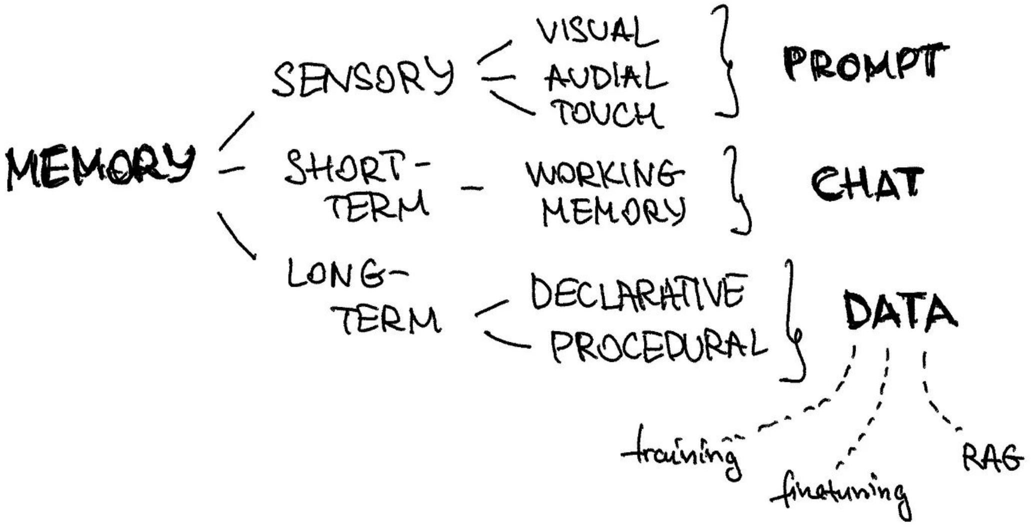
Sample Architecture

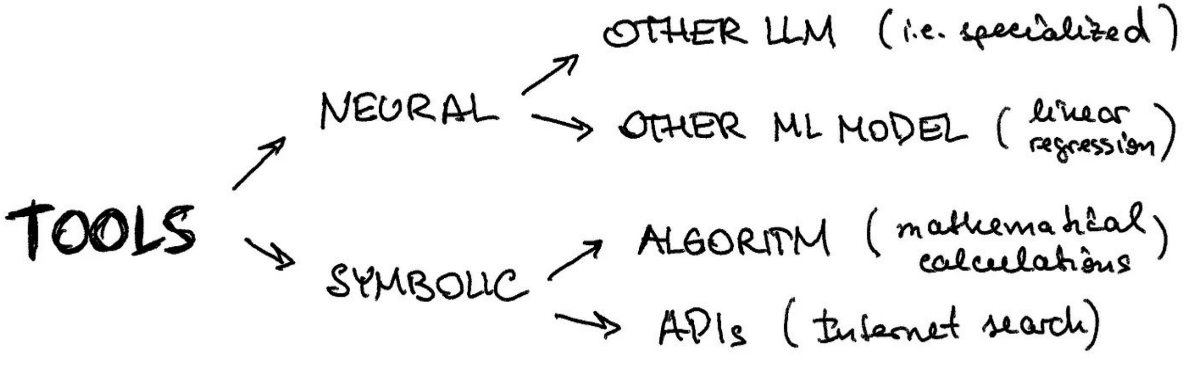
Navaneeth Malingan

Let’s Sum up Shortly!


RAG
AGENT
prompt
LLM
answer
prompt --- context
LLM
answer
prompt
memory
reasoning
LLM
answer
tools
action


INPUT
LLM
OUTPUT
I-O
INPUT
LLM
OUTPUT
CoT
INPUT
LLM
O1
O2
O3
majority vote
OUTPUT
CoT-SC
INPUT
LLM
OUTPUT
ToT


MEMORY
SENSORY
VISUAL
AUDIAL
TOUCH
PROMPT
SHORT-TERM
WORKING MEMORY
CHAT
LONG-TERM
DECLARATIVE
PROCEDURAL
DATA
training
finetuning
RAG


TOOLS
NEURAL
OTHER LLM (i.e. specialized)
OTHER ML MODEL (linear regression)
SYMBOLIC
ALGORITM (mathematical calculations)
APIs (Internet search)

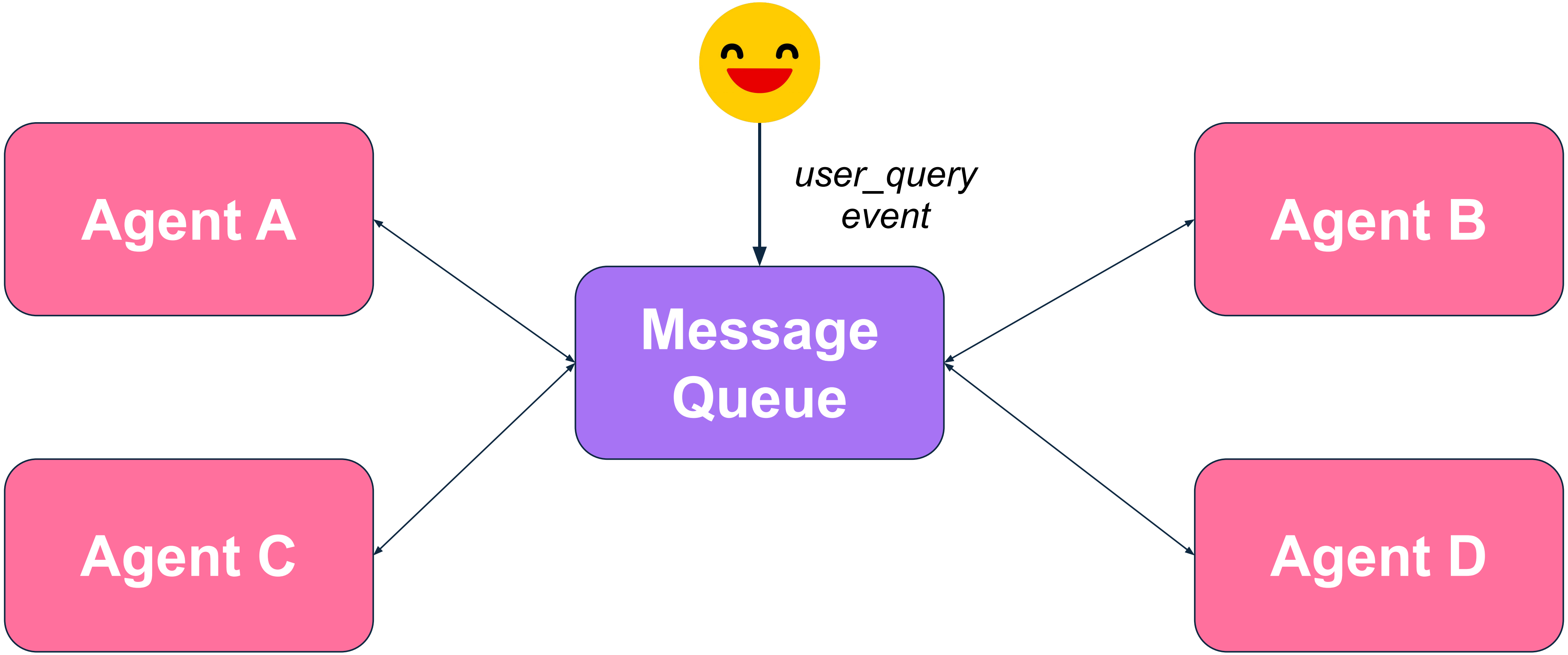
Summary

NUNNARI
LABS

Multi-Agent Architecture

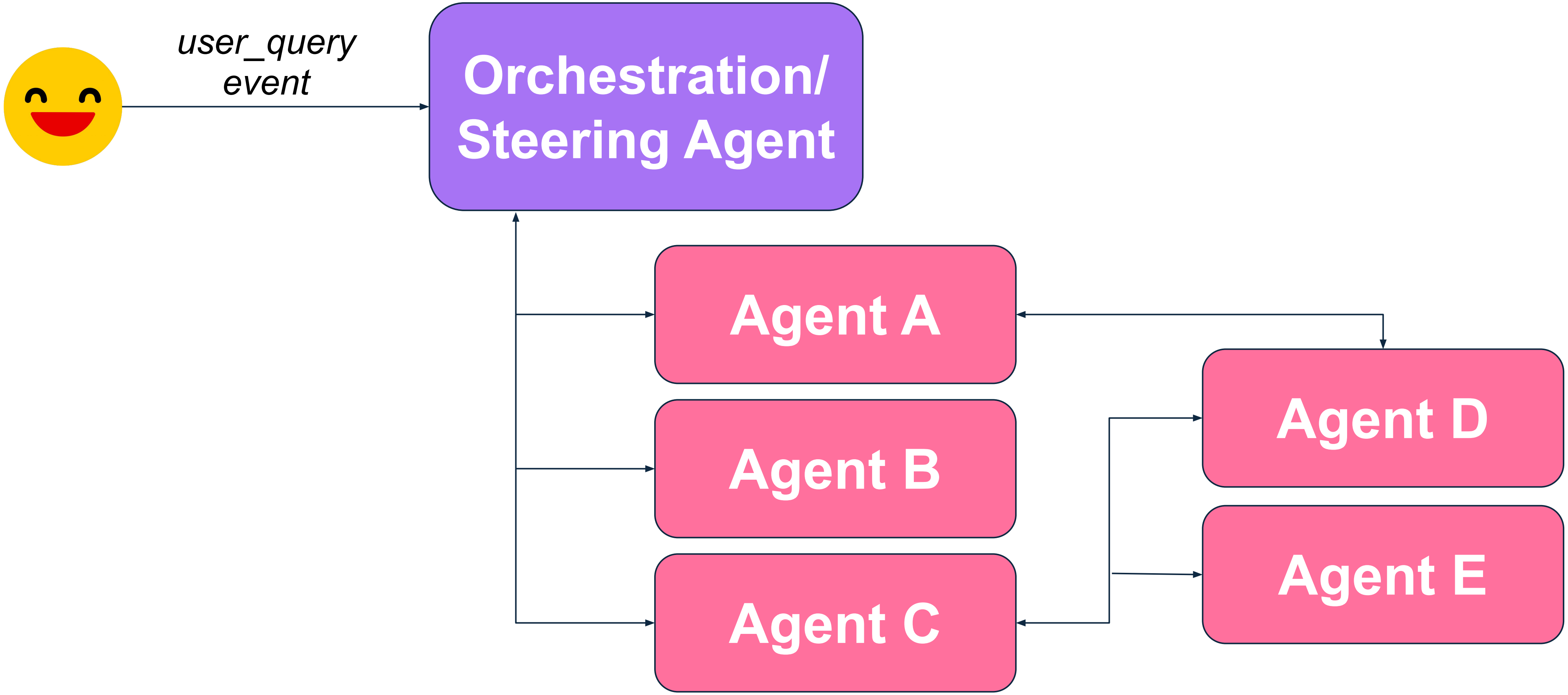
Navaneeth Malingan

Collaborative



Top Down/ Supervisory

NUNNARI LABS



NUNNARI
LABS

AI Agent vs Multi-Agent System

Navaneeth Malingan

Aspect	Agent	Multi-Agent System
Definition	Individual AI application that achieves goals by observing and acting upon the world	System of multiple agents working together to solve complex tasks
Components	Foundational model, tools, reasoning loop	Multiple individual agents, communication system (e.g., messaging queue)
Architecture	Single unit	Collaborative or Top-down/Supervisory
Complexity	Simpler, focused on specific tasks	More complex, can handle diverse or multi-step tasks
Decision Making	Independent	Coordinated or hierarchical
Specialization	Usually focused on a particular domain or task	Can involve multiple specialized agents for different aspects of a task
Scalability	Limited to individual capabilities	More scalable for complex problems
Communication	Primarily with user/environment	Inter-agent communication as well as with user/environment
Examples	Customer support agent, Service Center agent	Collaborative problem-solving system, Orchestrated workflow system
Tools Usage	Uses tools directly	Multiple agents can use different tools as needed
Typical Use Cases	Specific, focused tasks (e.g., booking a flight)	Complex, multi-step processes (e.g., comprehensive customer service)

AI Agent vs Multi-Agent System

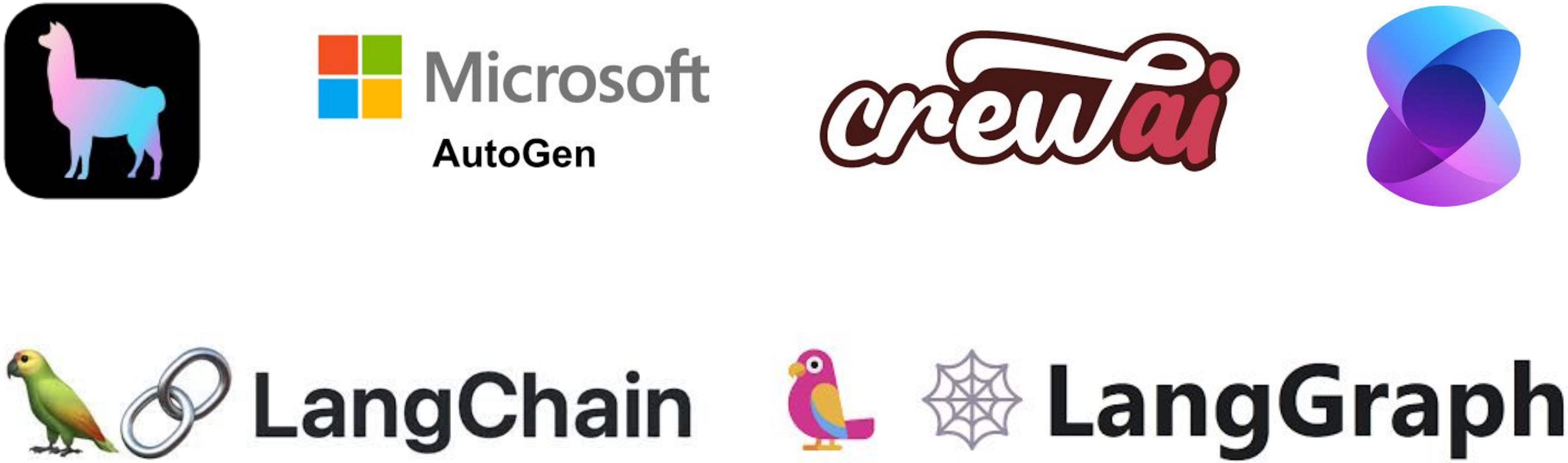
Navaneeth Malingan

NUNNARI LABS

Multi-Agentic Frameworks

Navaneeth Malingan

Popular Frameworks for Building Agents



LangGraph

LangGraph is a library for building **stateful**, **multi-actor** applications with **LLMs**, used to **create agent** and **multi-agent workflows**.

Key Advantages over other LLM frameworks

- **Cycles:** Unlike DAG-based solutions, LangGraph supports flows with cycles, crucial for developing complex agentic architectures.
- **Controllability:** Offers enhanced control over workflow processes, ensuring precise and adaptable operations.
- **Persistence:** Built-in persistence allows for advanced features such as human-in-the-loop interactions and memory retention, enabling more dynamic and responsive applications.

LangGraph - Key Features

NUNNARI LABS

- **Cycles and Branching**: Implement loops and conditionals in your apps.
- **Persistence**: Automatically save state after each step in the graph. Pause and resume the graph execution at any point to support error recovery, human-in-the-loop workflows, time travel and more.
- **Human-in-the-Loop**: Interrupt graph execution to approve or edit next action planned by the agent.
- **Streaming Support**: Stream outputs as they are produced by each node (including token streaming).
- **Integration with LangChain**: LangGraph integrates seamlessly with *LangChain* and *LangSmith* (but does not require them).

Why LangGraph?

Simplifies Development

a. State Management
b. Agent Coordination (Communication between Agents)
c. To Define Workflows and Logics

Flexibility

a. Flexibility to define custom agent logic and communication protocols.
b. Build tailored applications, from chatbots to complex multi-agent systems.
c. Provides the tools for highly customized solutions specific to your needs.

Why LangGraph?

Scalability

a. Large Scale Multi Agent Application
b. Handle high volume of Interaction and complex workflows
c. Enterprise level application (Cloud with drag and drop UI)

Fault Tolerance

a. Handle Error
b. It will not stop the flow when a agent fails

NUNNARI
LABS

It’s Time to Code!

Navaneeth Malingan

References

- **All you need to know to Develop using Large Language Models**, Sergei Savvov - towardsdatascience.com
- **Fixing Hallucinations in LLMs**, Sergei Savvov - betterprogramming.pub
- **Intro to LLM Agents with LangChain: When RAG is Not Enough**, Alex Honchar - medium.com/towards-data-science

Thank You!

Navaneeth Malingan
navaneeth@nunnarilabs.com
www.nunnarilabs.com

NUNNARI LABS

AI Tamil Nadu

TensorFlow
User Group Coimbatore

Connect with Me!