



APPLE AirPods 4
for each member
of the winning team



REPLY Water Bottle
for each member of
the second and third-placed teams

ARCADE GAME

for the winning University





40
points

BEST PROJECT

Among the winners at each University

25
points

WINNER OF THE QUIZ

Play during the Workshop!

20
points

WINNER OF THE INSTAGRAM BATTLE

Vote for your University on @ReplyU Instagram Profile

15
points

BEST PICS TAKEN DURING THE EVENTS

Showing both the Reply and your University logo.
Share it from a public profile on Instagram.
Use #StudentClash and tag @ReplyU



P R O M P T I N G
T H E F U T U R E



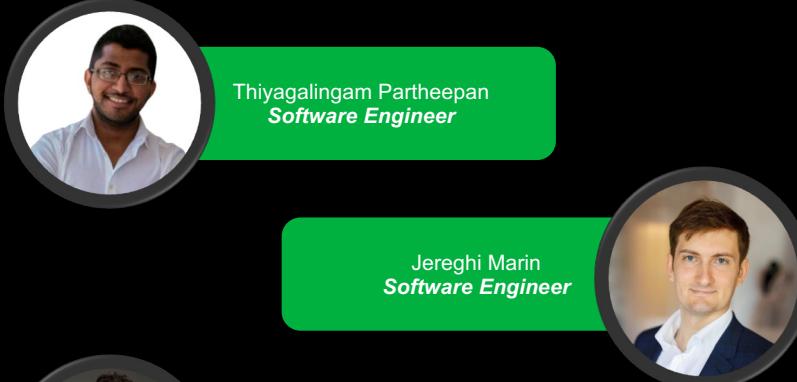
Thiyagalingam Partheepan
Software Engineer



Annaloro Giacomo
Software Engineer



Cucinella Francesco
Software Engineer



Jereghi Marin
Software Engineer



Stefano Del Col
Data Scientist



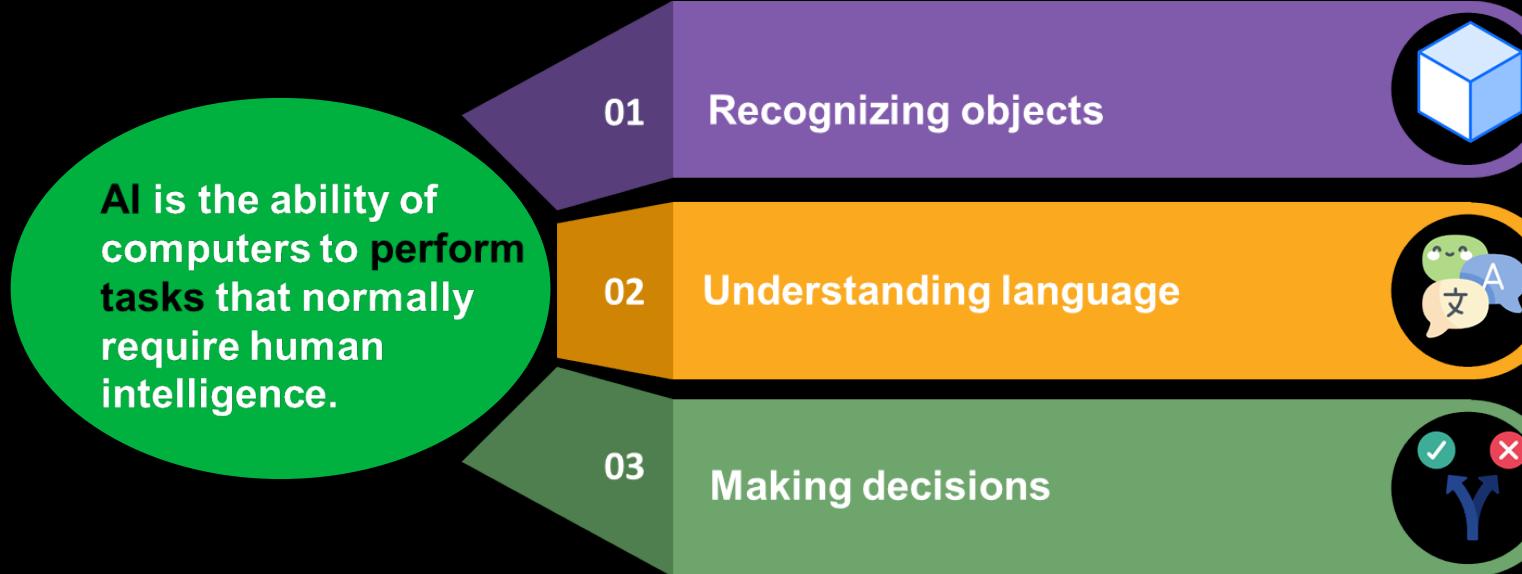
Juri Fabbri
Data Scientist



Gabriele Rapisarda
Software Engineer



WHAT IS ARTIFICIAL INTELLIGENCE?



AI is used in everyday life, like virtual assistants (e.g., Siri, Alexa), spam filters in emails, and recommendations on Netflix.



WHAT IS ARTIFICIAL INTELLIGENCE?

What do you see?



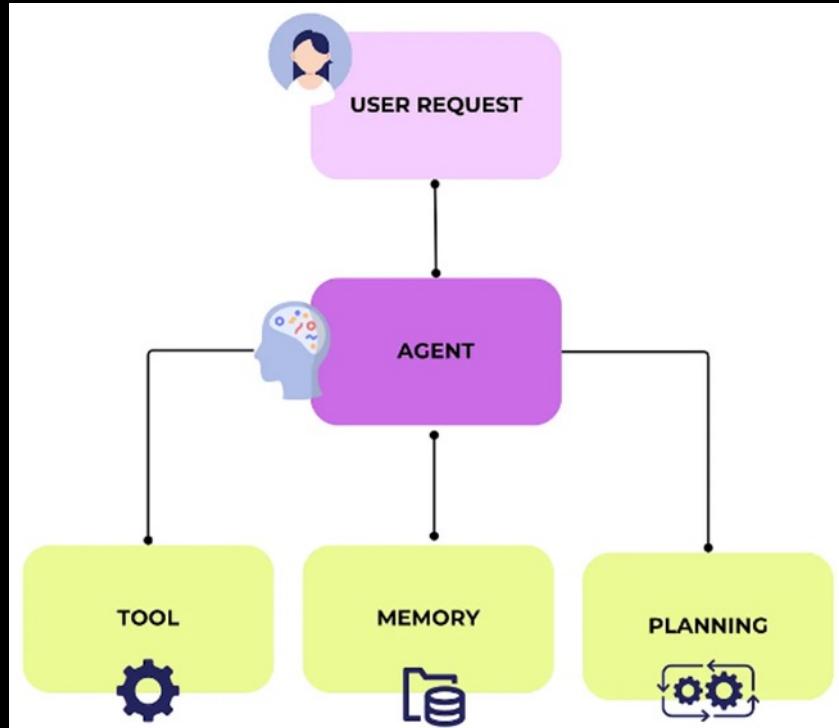
@teenybiscuit

WHAT IS AN AGENT?

An **agent** is a "virtual assistant" programmed to complete a specific task

It mimics **human decision-making** and it can be in the form of software programs, AI systems, or even autonomous machines.

Thanks to LLM's we are now in the era of **LLM – based agents**





LARGE LANGUAGE MODEL

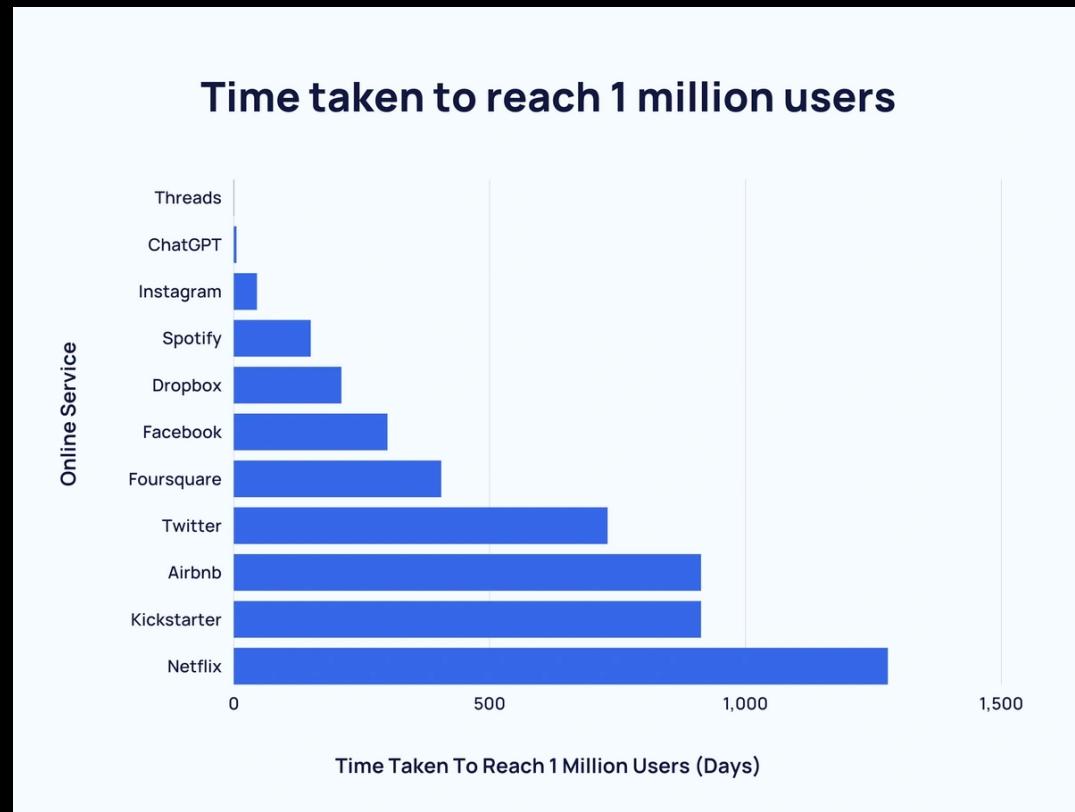
LLM

RAPID PACE EVOLUTION

ChatGPT acquired **1 million** users just **5 days** after launching in **November 2022**.

By comparison, it took Instagram approximately **2.5 months** to reach 1 million downloads.

Whereas Netflix had to wait around **3.5 years** to reach 1 million users.

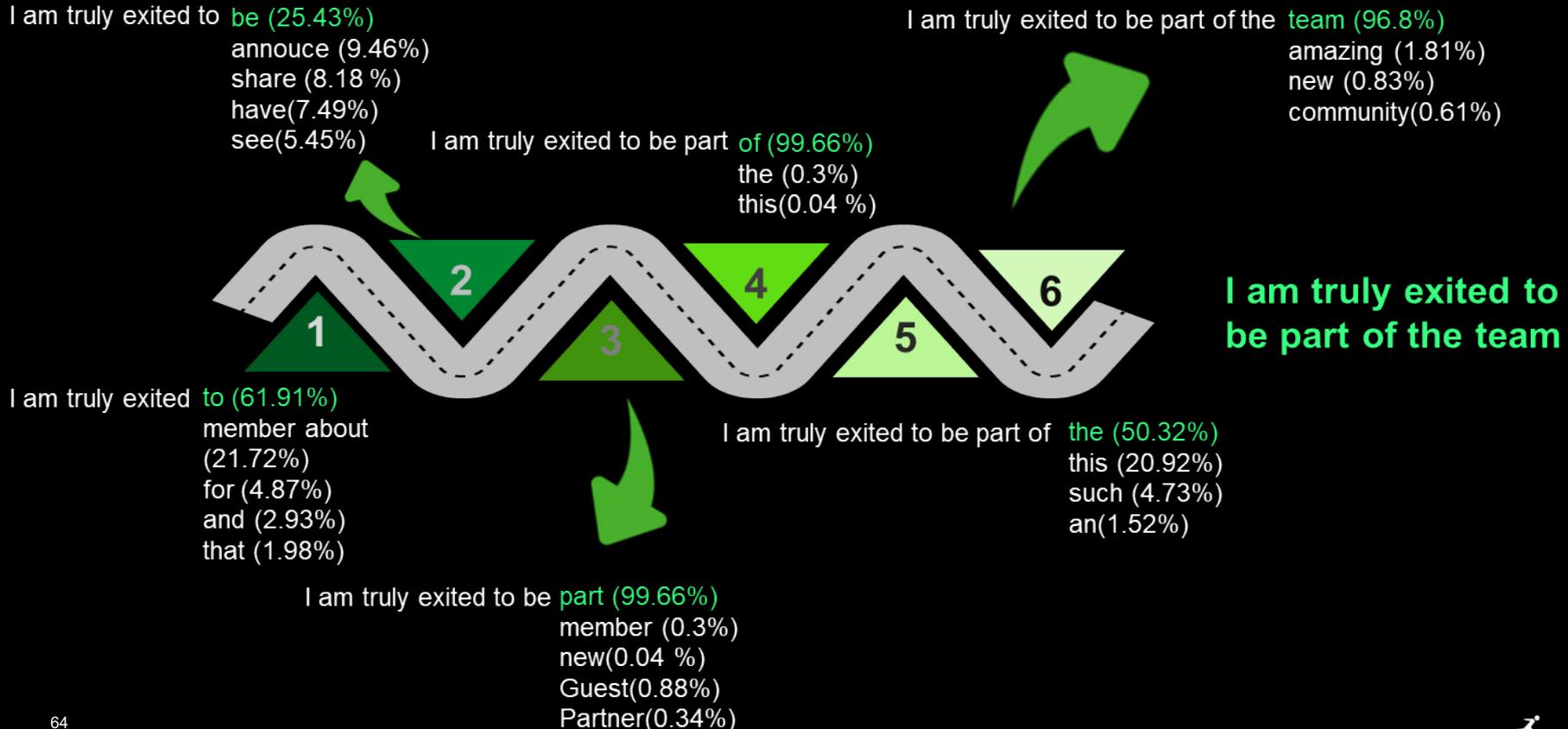


Source: Exploding Topics



HOW LLMS WORKS

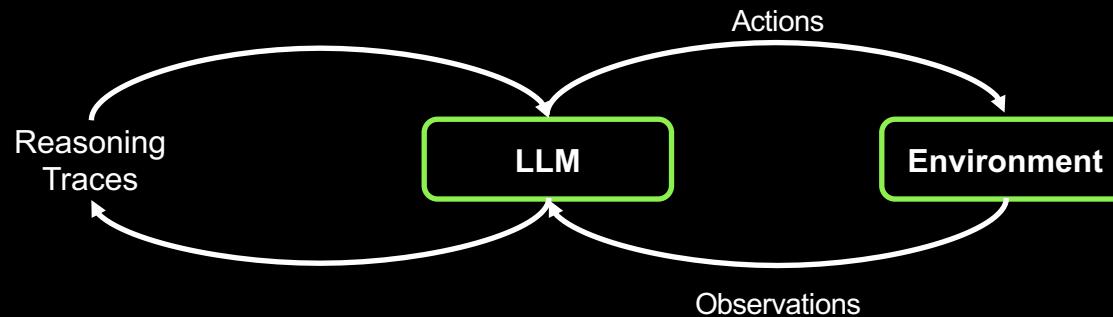
PROBABILITY



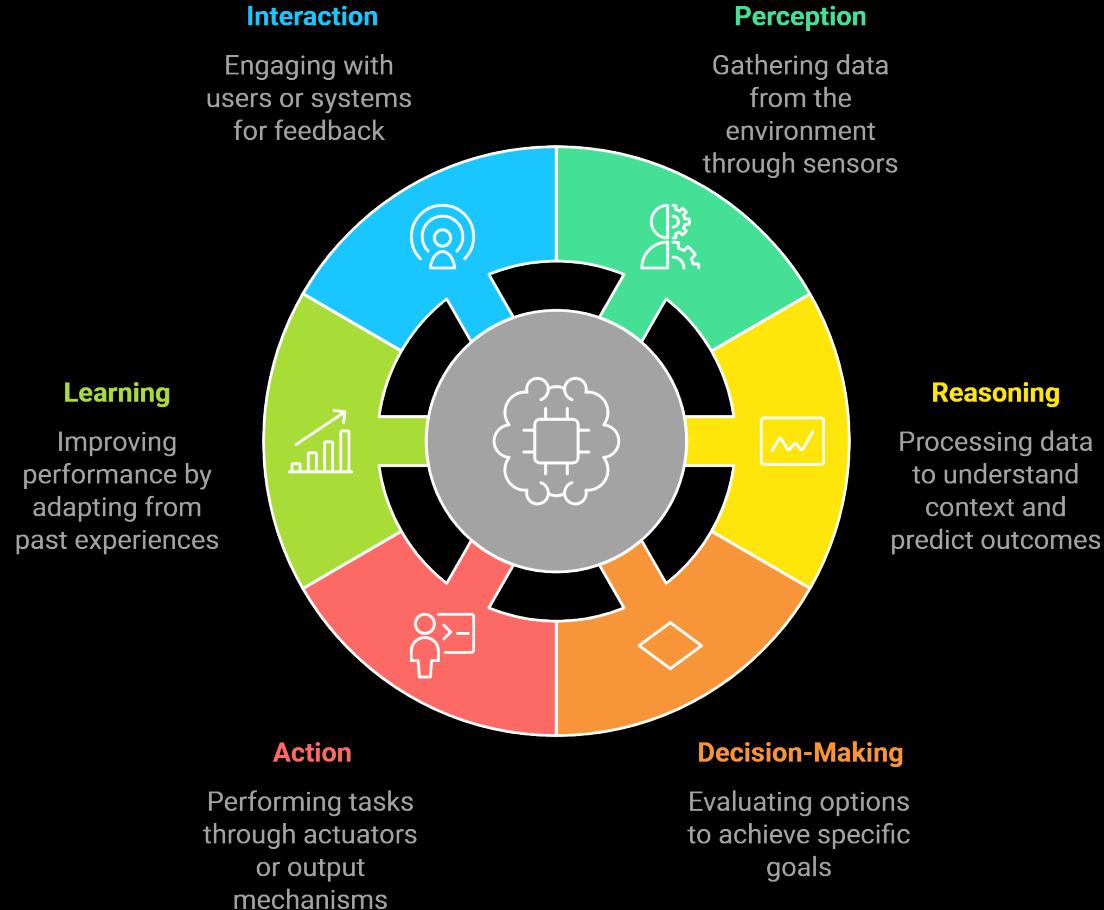
WHAT IS AN LLM-AGENT

An LLM Agent **elevates** the limited functionalities of basic **LLM** models. It's a piece of software that **uses LLMs** to judge what's the best approach to complete a task.

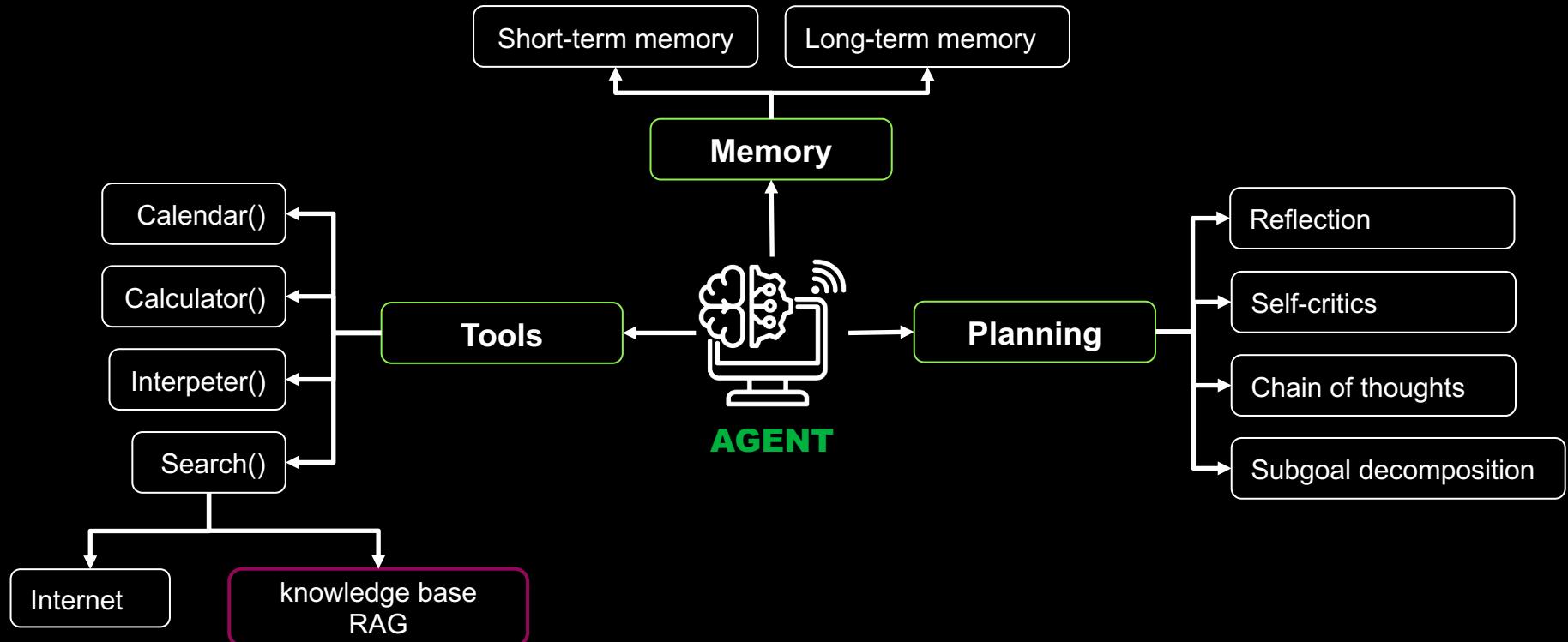
Thanks to LLMs the agent is able to use **external tools** and **evaluate** their results, even try multiple approaches and choose the best possible result.



HOW DO AGENTS WORK?



HOW DO AGENTS WORK?

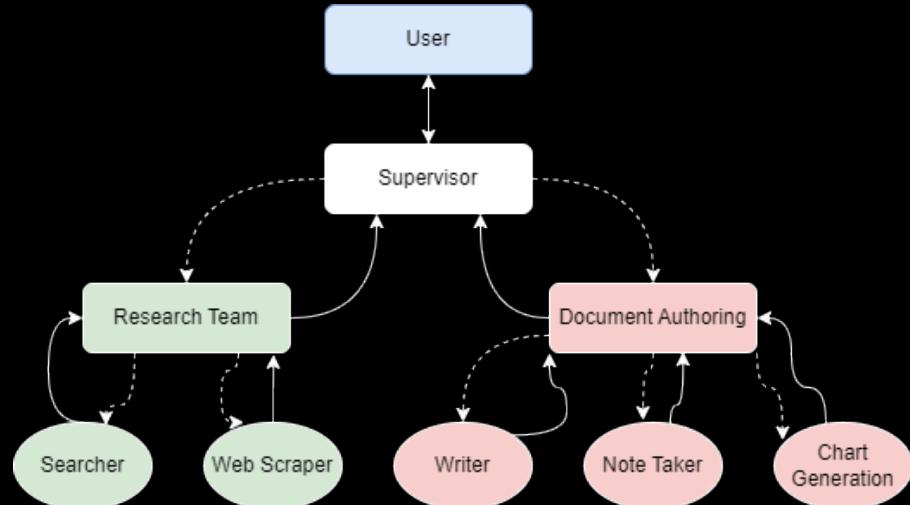


MULTI-AGENTS

A **Multi-Agent System (MAS)** is a system where multiple agents interact in a shared environment to reach their goals.

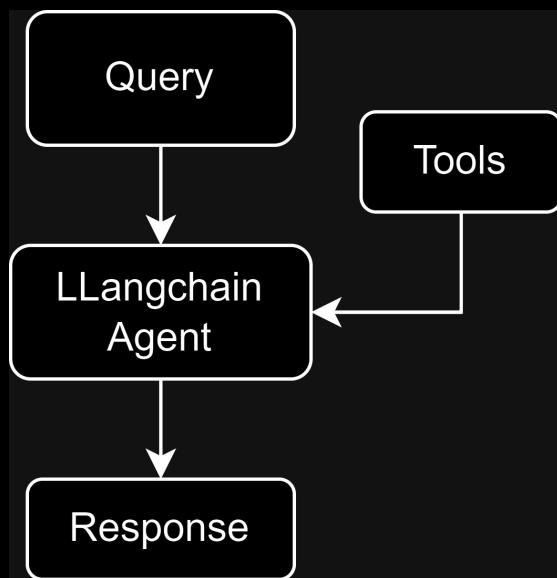
Why Multi-agents?

- **Collaboration:** lead to better outcomes
- **Scalability:** Single agents are limited in handling complex tasks and don't scale with many tools and capabilities
- **Specialization:** Different agents can specialize in different tasks



BUILDING AGENTS IN PYTHON: LANGCHAIN

LangChain is one of the most used libraries in Python for the creation of LLM agents. It streamlines the code to the bare minimum.



```
from langchain.agents import initialize_agent, load_tools
from langchain.llms import OpenAI
from langchain.memory import ConversationBufferMemory

# Initialize the LLM and load the tools
llm = OpenAI(api_key="YOUR_API_KEY")
memory = ConversationBufferMemory(memory_key="chat_history")
tools = load_tools(["ddg-search"], llm=llm)

# Initialize the agent and run a query
agent = initialize_agent(tools, llm,
agent_type="conversational-react-description", memory=memory,
verbose=True)

response = agent.run("What's the capital of France?")
```



REAL WORLD APPLICATION

CUSTOMER SERVICE

Used in virtual assistants to provide 24/7 support, and guiding users through complex processes.

FINANCE

Analyze data and execute trades autonomously, leveraging real-time information to optimize strategies.

EDUCATION

Help students practice and reinforce their knowledge providing personalized learning.



LOGISTICS

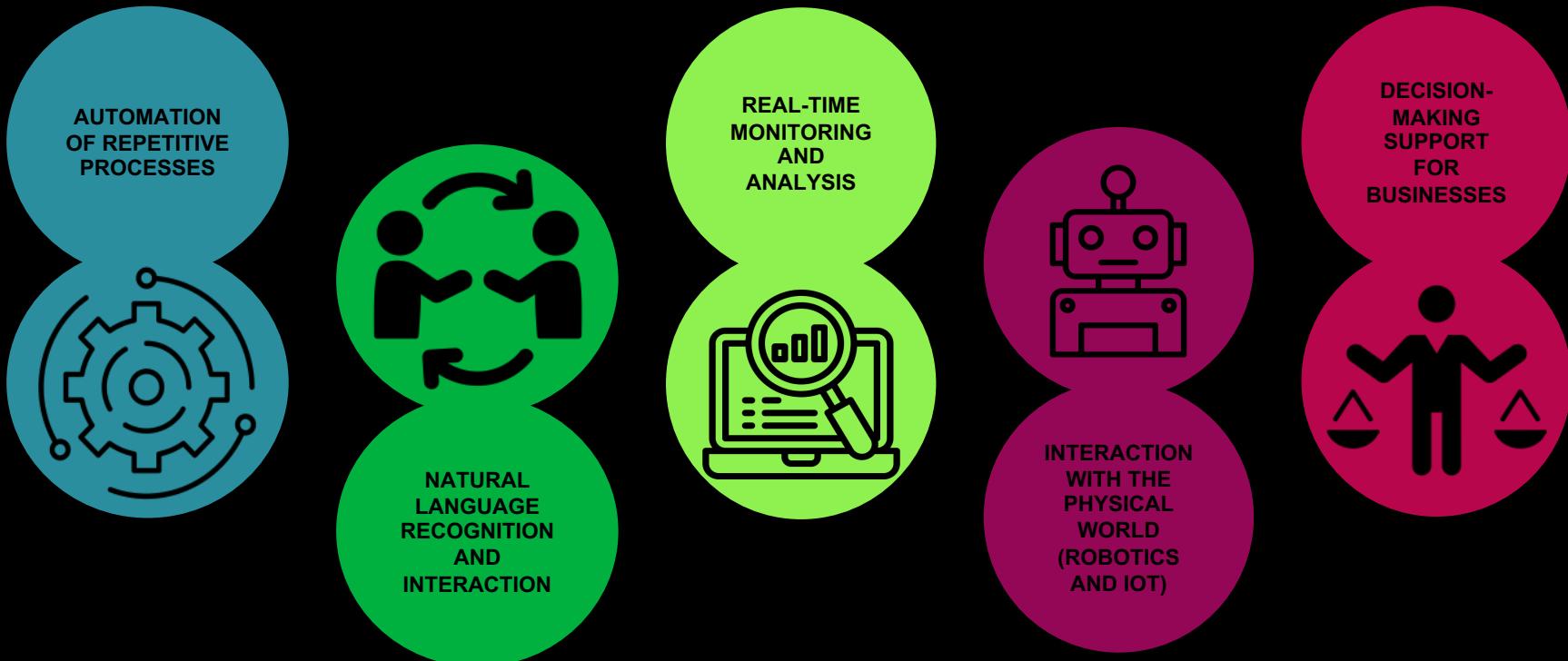
Route optimization enhancing efficiency in resource allocation and distribution

HEALTHCARE

Providing medication reminders and manage schedules improving patient care and adherence to treatment plans

MORE

POTENTIALITY



REAL USE CASES

AGENTIC SEARCH ENGINE

During this year's *Reply hackathon* in Saint Vincent, we developed an **agentic search engine**.

It's a powerful tool that enables us to perform searches on both **structured** and **unstructured databases** seamlessly.

Now, as we bring this technology to **market**, we aim to transform data access and analysis by offering a powerful, versatile solution that meets the needs of **diverse industries and use cases**.



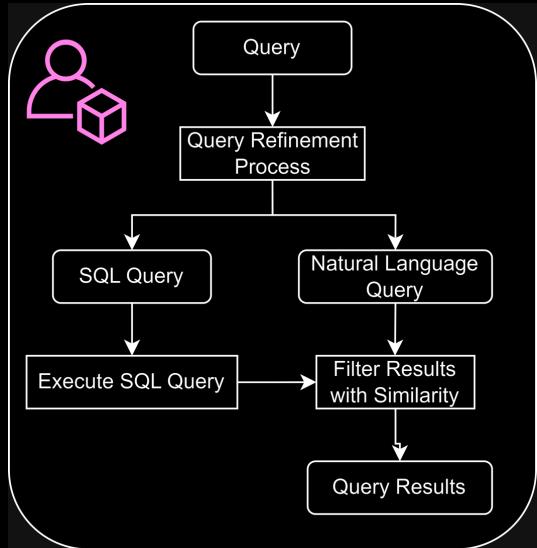
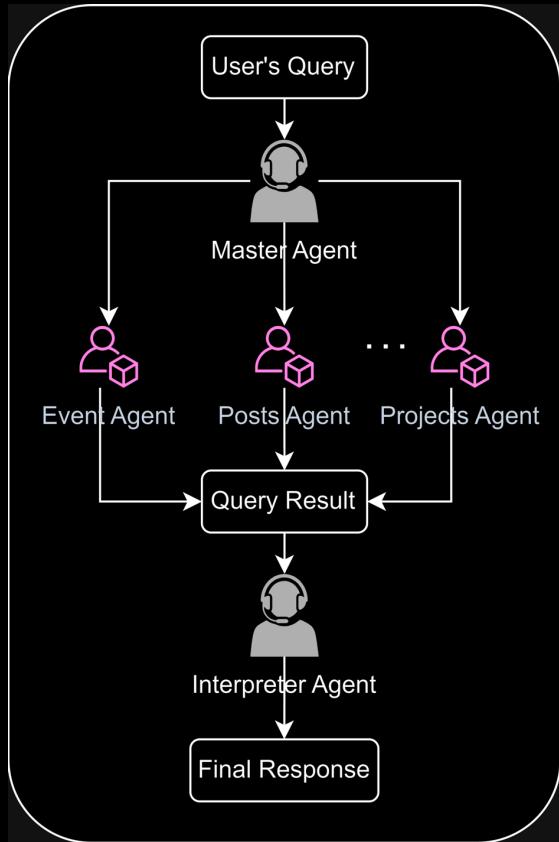
AGENTIC SEARCH ENGINE

Multi-Agent Systems

can be used to develop an intelligent search agent

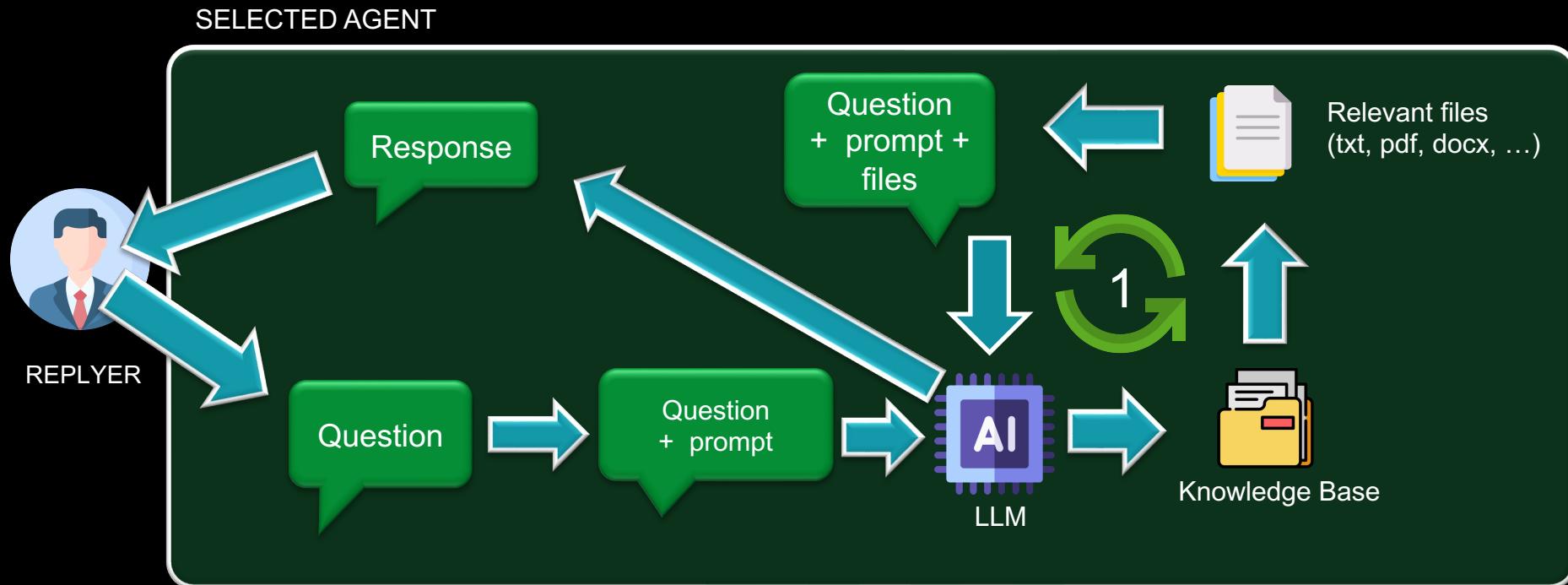
Why Multi-agents?

- **Specialization:** Each agent is specialized in querying a particular data source
- **Performance:** Queries can be executed concurrently

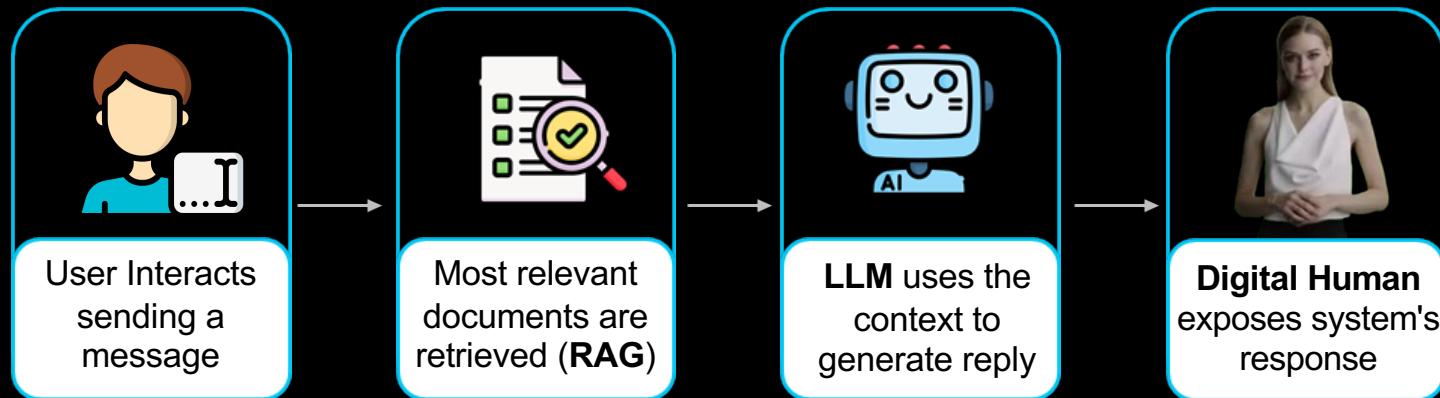


NEURONS

AI platform by Reply for experimentation and productivity.



DIGITAL HUMAN & RAG



Qdrant



Open AI



HeyGen

ST-AI-LIST: YOUR IMAGE STYLER



REPLY Neurons BETA ⓘ



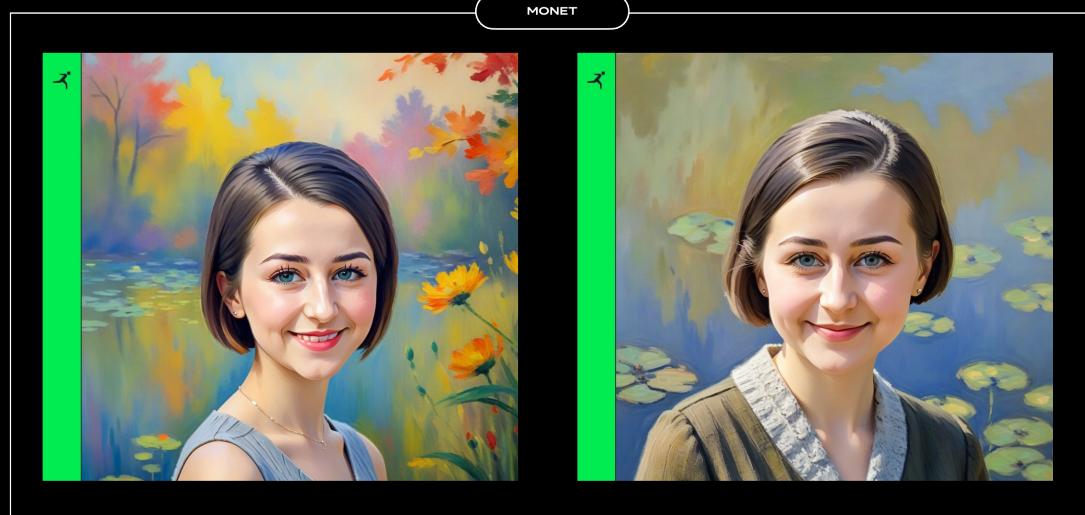
ST-AI-LIST

Take your Reply portrait in style.

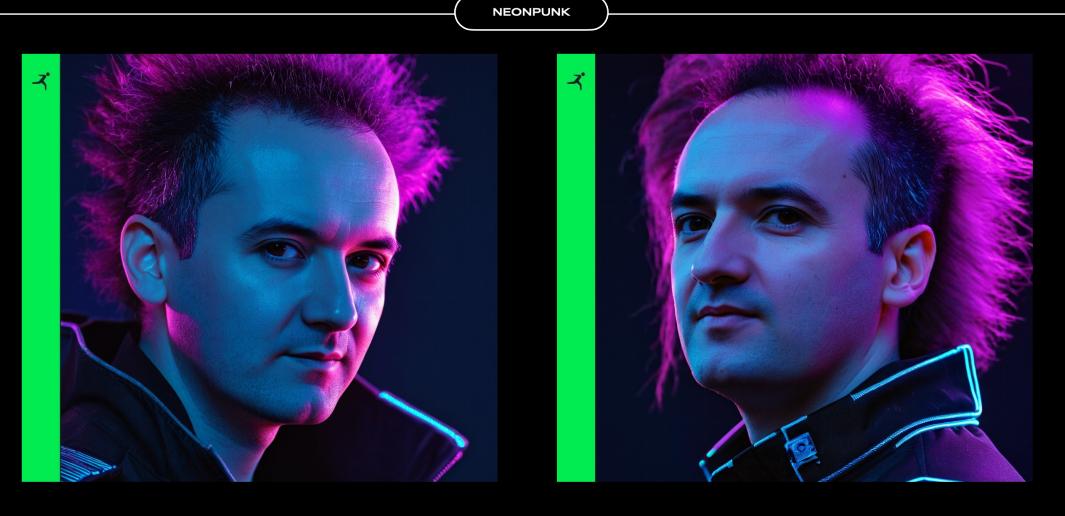
ST-AI-LIST: YOUR IMAGE STYLER



Input Image



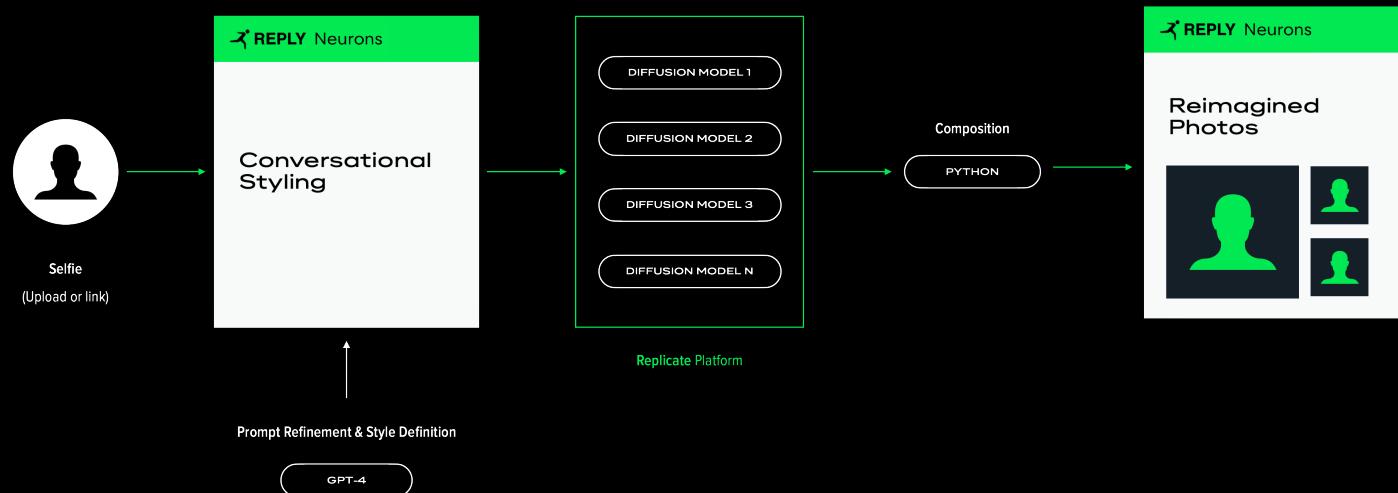
ST-AI-LIST: YOUR IMAGE STYLER



Input Image

NEONPUNK

ST-AI-LIST: YOUR IMAGE STYLER



[+ Create a new agent](#) [My Agents](#)

ST•AI•LIST [About](#)



ST•AI•LIST

Take your Reply portrait in style
by Simone Oberti
on September 25, 2024

[Copy link](#)

```
{"description":"Sara", "image_url": "https://d25m59h0ya0u4t.cloudfront.net/tamtamy/thumbmatic/oongc07goboih0imp050e6k8dq-0-0.png", "prompt":"portrait of a woman in a christmas setting with a black cat", "style":"Lowpoly"}
```

This is an external Agent of type Custom API. Double-check output accuracy and provide feedback if necessary.



- 1) URL IMMAGINE
- 2) PROMPT: portrait of a woman in a christmas setting with a black cat
- 3) STILE: Lowpoly

EINAUDI DIGITAL HUMAN

What if legendary personalities of the past could come back to life allowing users to talk to them today?

This is the aim that drives this project: let people talk about economy with Luigi Einaudi, one of the most relevant Italian personalities.

Starting from picture reference and through manual sculpting activities, the physiognomy of Luigi Einaudi has been reproduced and animated, while an LLM has been trained using his most important writing about the economy.

Through a dedicated website, people will be soon able to dialogue (both in chat and through voice) with him and [access his knowledge](#).

[Test it](#)



CASE STUDY



LEVEL 1 – THE MAIN CHALLENGE

THIS LEVEL IS MANDATORY

Think of a problem in one of the following categories and
imagine how agents can help you:

- Student life
- Hobbies
- Choose your own!

The deliverable of this challenge in a presentation.



LEVEL 2 – PRACTICE

THIS LEVEL IS MANDATORY

Implement a simple agent that simulates the use case chosen in Level 1 (or at least a part of it).

No need of coding skills needed but if you want you can show yours!



LEVEL 3 – BONUS

THIS LEVEL IS NOT MANDATORY

Create a **multi-agent system** that collaborates to solve a complex problem.



EVALUATION CRITERIA

INNOVATIVE DESIGN

✨ **CREATIVITY AND INNOVATION:** 40%

✓ **EFFECTIVE PRESENTATION:** 20%

💡 **IMPACT:** 20%

⚙️ **TECHNICAL:** 20%





Download the brief at

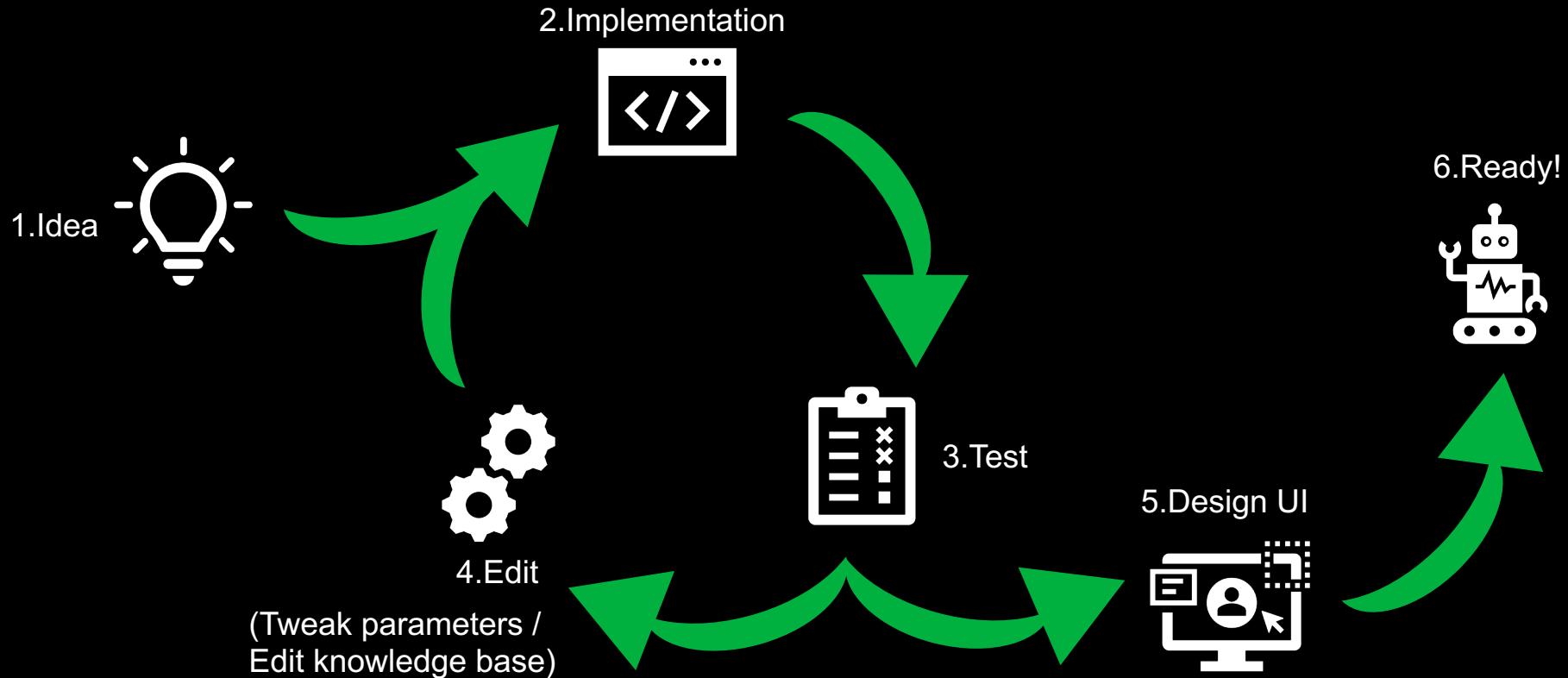
studentclash.reply.com





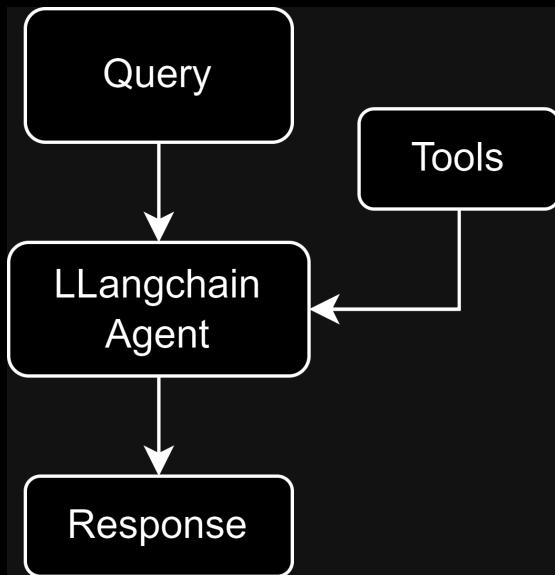
EXTRA MATERIALS

AGENT CREATION FLOW



BUILDING AGENTS IN PYTHON: LANGCHAIN

LangChain is one of the most used libraries in Python for the creation of LLM agents. It streamlines the code to the bare minimum.



```
from langchain.agents import initialize_agent, load_tools
from langchain.llms import OpenAI
from langchain.memory import ConversationBufferMemory

# Initialize the LLM and load the tools
llm = OpenAI(api_key="YOUR_API_KEY")
memory = ConversationBufferMemory(memory_key="chat_history")
tools = load_tools(["ddg-search"], llm=llm)

# Initialize the agent and run a query
agent = initialize_agent(tools, llm,
agent_type="conversational-react-description", memory=memory,
verbose=True)

response = agent.run("What's the capital of France?")
```



OPENAI AGENTS

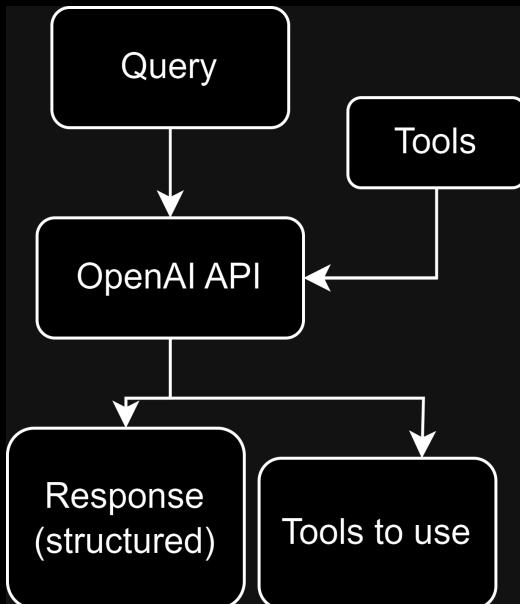
OpenAI Agents are AI-driven systems designed to streamline task automation and data processing by leveraging the power of GPT-4.

- **Advanced Language Model:** Utilizes GPT-4 to produce accurate, context-rich responses based on detailed user inputs, enabling effective interaction and decision-making.
- **Integrated Knowledge Sources:** Connects with diverse data sources, including vector databases, to enrich understanding and ensure responses are both relevant and informed.
- **Flexible Prompt Configuration:** Customizable prompts allow agents to be tailored to meet specific objectives, ideal for applications in customer support, content creation, and complex data analysis.
- **Scalable & Continuously Improving:** OpenAI Agents are built to scale across different use cases and adapt to new data, ensuring consistent performance and evolving accuracy.



BUILDING AGENTS IN PYTHON: CUSTOM AGENT USING OPENAI

An agent is a piece of code that uses LLM to perform specific tasks. At any point of your software, you can use an LLM to evaluate a string and get a response. To do this in python you can leverage the OpenAi Python library.



```
import openai
# Set your OpenAI API key
openai.api_key = "your_openai_api_key"
# Define the model and parameters for the API call
response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[
        {"role": "system", "content": "You are a helpful
assistant."},
        {"role": "user", "content": "Can you explain how to
make an API call using OpenAI's Python library?"}
    ],
    max_tokens=100, # max tokens in the response
    temperature=0.7,# response creativity (0.0 - 1.0)
)
print(response['choices'][0]['message']['content'])
```

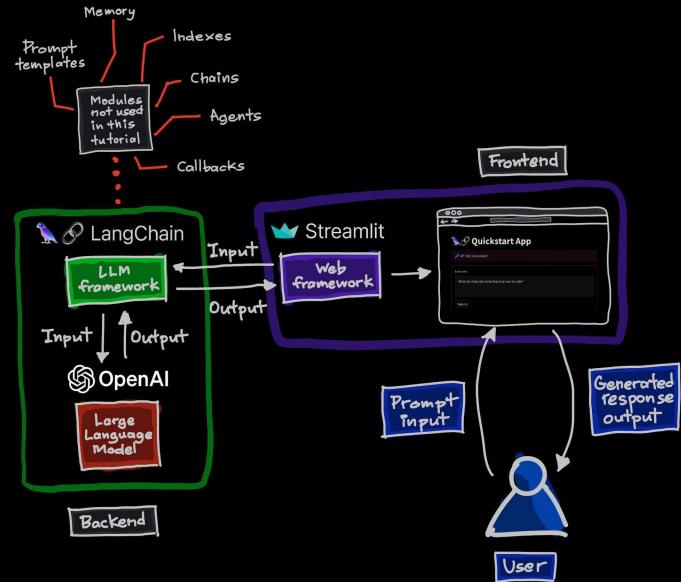


CREATE A FAST FRONT-END FOR YOUR AGENT: STREAMLIT

Streamlit is a straightforward and powerful Python library for building interactive web interfaces, ideal for rapid prototyping and data visualization. Perfect for AI agents, it enables the creation of custom, user-friendly dashboards without advanced front-end skills.

- **Speed:** Turn Python scripts into functional web applications in minutes.
- **Simplicity:** No need to learn HTML/CSS or JavaScript.
- **Interactivity:** Add inputs, charts, tables, and dynamic visualizations to make your AI agent accessible and user-friendly.

Using Streamlit as the front-end for your AI agent allows for faster testing and deployment with a professional and responsive UI.



BEDROCK AGENTS

Bedrock Agents are an advanced solution for automating specific tasks through artificial intelligence, leveraging Amazon Web Services (AWS) infrastructure. They are designed to seamlessly integrate with cloud services, facilitating the development of AI agents that can handle large amounts of data and respond in a scalable and secure manner.

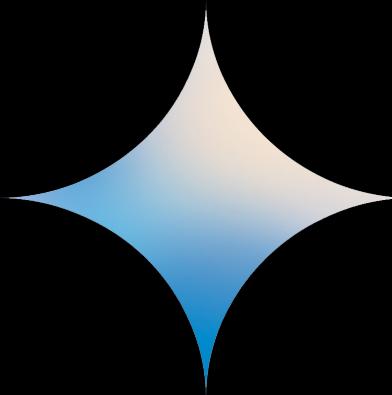
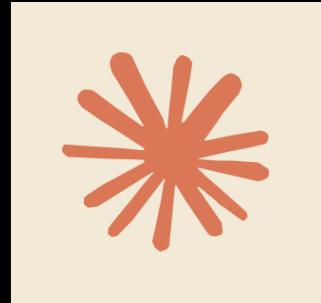
- **Language Model (LLM):** Bedrock uses powerful, AWS-optimized AI models to generate accurate, context-aware responses based on user inputs.
- **Knowledge Base:** Agents can be enriched with data imported from various formats and databases, such as **vector DBs**, to ensure deep understanding of requests and improve response relevance.
- **Prompt Customization:** Bedrock Agents enable flexible configuration of prompts and instructions, allowing the agent's responses to align with business objectives.



Amazon Bedrock



IF YOU NEED HELP, ASK AI!



SAMPLE IDEA

1. Find an Idea! Let's say we want to add a chatbot to a website
2. Download some PDF from website pages
3. Create the Agent using OpenAI or write some code using Llangchain and python:
 - o Import pdf in a Vector DB
 - o Configure a LLM Api Key
 - o Define a good Prompt
4. Test and adjust
5. Design a minimal frontend using Streamlit
6. You have created an AI Agent!



