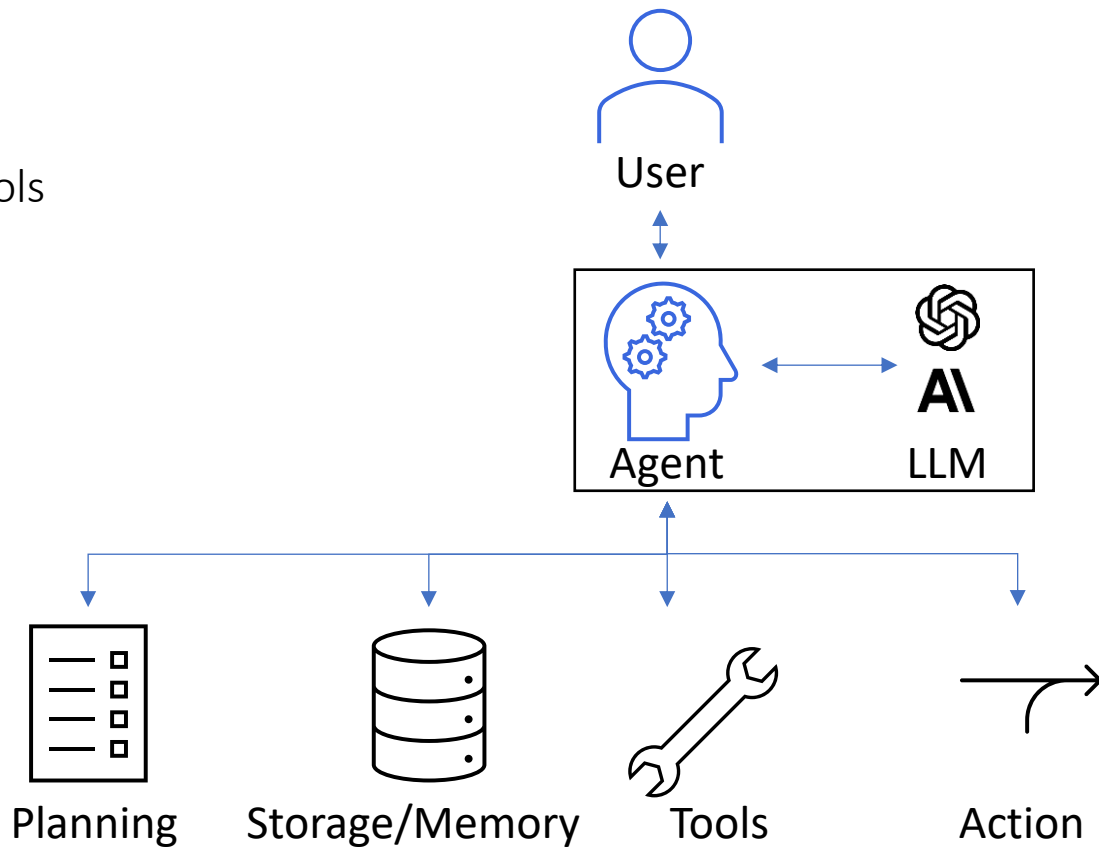


Agents

# Agents

What is an Agent?

- LLM apps execute tasks
- core element: agent
- uses planning, memory, and tools
- can perform actions
- is an expert in its field



# Agents

## Levels of AI Agents

Generality →							
Level	Techniques	Performance	Capabilities	Key Characteristics	Use Cases	Narrow Domain	General Wide-Range Domain
5	LLM-based AI + Tools (Intent + Actions + Reasoning & Decision Making + Memory + Reflection + Autonomous Learning + Generalisation + Personality (Emotion + Character) + Collaborative behaviour (Multi-Agents))	Superhuman > 100% of Skilled Adults	True Digital Persona	Agent represents the user in completing affairs, interacts on behalf of user with others, ensuring safety & reliability.	Agent acts on behalf of user to complete tasks, interacting with others while ensuring safety & reliability.	Superhuman Narrow-AI AlphaFold, AlphaZero, StockFish	Artificial Super Intelligence (ASI) Not yet achieved
4	LLM-based AI + Tools (Intent + Actions + Reasoning & Decision Making + Memory + Reflection + Autonomous Learning + Generalisation)	Virtuoso Equal to 99% of Skilled Adults	Memory & Context Awareness	Agent senses user context, understands user memory, and proactively provides personalised services at times.	A personalised virtual assistant enhances UX by understanding context & memory while acting proactively.	Virtuoso Narrow-AI AlphaGo, Deep Blue	Virtuoso AGI Not yet achieved
3	LLM-based AI + Tools (Intent + Actions) + Reasoning & Decision Making + Memory & Reflection	Expert Equal to 90% of Skilled Adults	Strategic task Automation	Using user-defined tasks, agents autonomously plan, execution steps using tools, iterates based on intermediate feedback until completion.	Agents autonomously plan and execute steps based on intermediate feedback	Expert Narrow-AI Purpose build, specific task orientated Agents	Expert AGI Not yet achieved
2	IL/RL-based AI + Tools (Intent + Actions) + Reasoning & Decision Making	Competent Equal to 50% of Skilled Adults	Deterministic Task Automation of Skilled Adults	Based on user description of deterministic task, agent auto-completes steps in predefined action.	User: "Check the weather in Beijing today".	Competent Narrow-AI Conversational AI build frameworks with LLM, RAG, etc.	Competent AGI Not yet achieved
1	Rule-Based AI + Tools (Intent + Actions)	Emerging Equal to Unskilled Humans	Simple Step Sequence	Agents complete tasks following exact steps, pre-defined by users or developers.	User: "Open Messenger" Then: "Open the first unread email in my mailbox and read its content" Then: "Call Alice".	Emerging Narrow-AI Single Rule-based systems, SHRDLU, GOFAT	Emerging AGI ChatGPT, Gemini, Llama 2, etc.
0	No AI Tools (Intent + Rules + Actions)	No AI	No AI	No AI	No AI	Narrow Non-AI UI Driven Software	General Non-AI Human-In-The-Loop Computing Mechanical Turk

Adapted From: <https://arxiv.org/pdf/2405.06643>

Source: <https://cobusgreyling.medium.com/5-levels-of-ai-agents-updated-0ddf8931a1c6>

# Agents

Which Frameworks are available?



## LangGraph

LangGraph

- built on top of LangChain (same team)
- flexible, customizable
- works with any LLM
- not intuitive for non-programmers



CrewAI

- very intuitive
- suitable for many agents
- supports many LLM providers
- not ideal for very complex tasks



## OpenAI

Swarm

- very easy to use
- suitable for beginners
- only supports OpenAI

# Agents

Which Frameworks are available?



## AutoGen

- two agents
- good for code generation
- difficult setup



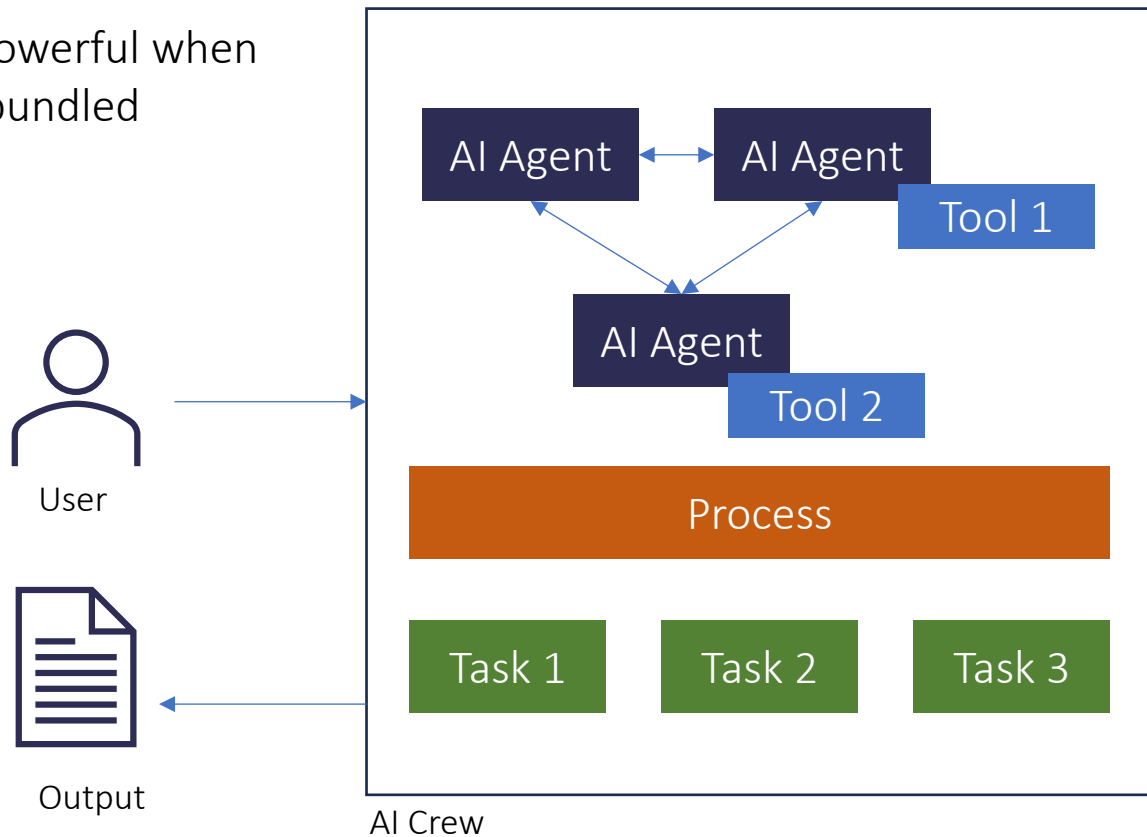
## Magentic-One

- suitable for beginners
- pre-defined 5 agents: manager, web-surfer, file-surfer, coder, terminal
- built on top of AutoGen
- limited support and documentation

# Agents

What is an AI Crew?

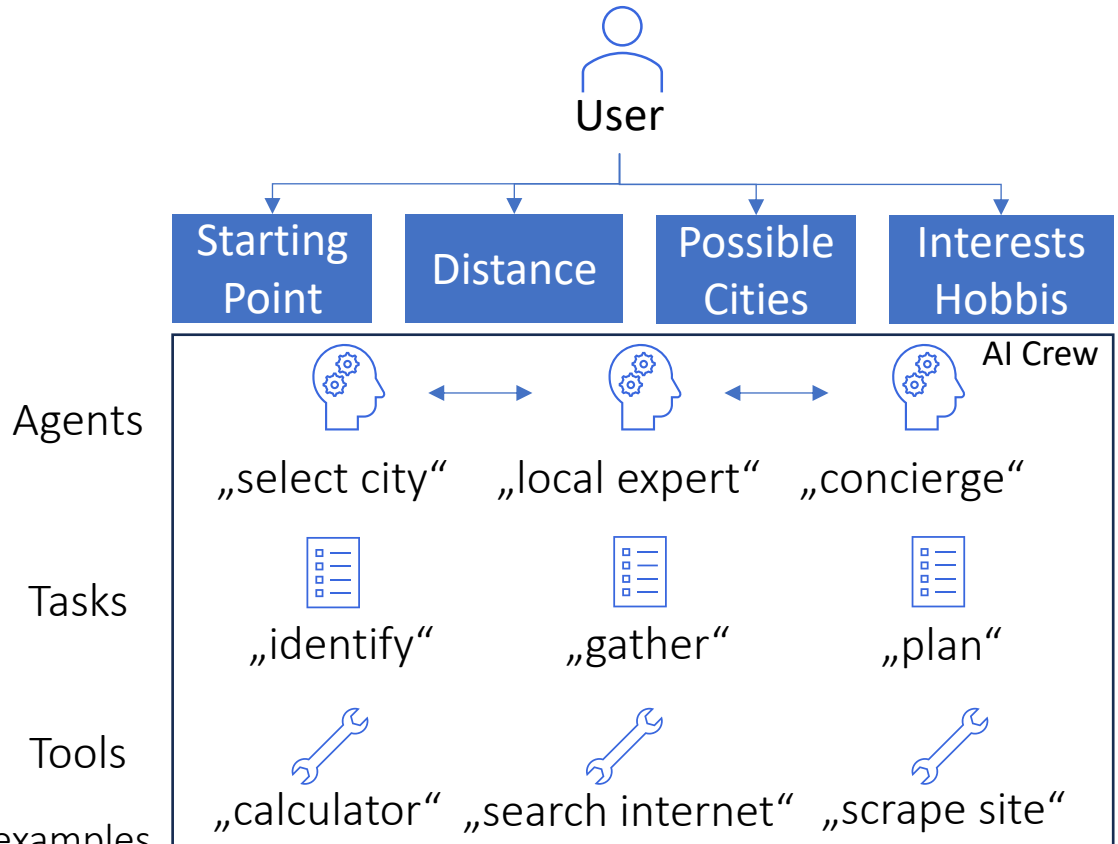
- concept extremely powerful when multiple agents are bundled
- system of experts



# Agents

Example – Plan your vacation

- 1. Define Goal
- 2. User Inputs
- 3. Set up
  - agents
  - tasks
  - if needed:
    - tools
    - process
    - ...



Idea found at  
<https://github.com/joaomdmoura/crewAI-examples>

# Agents

Example – Plan your vacation: Agents





# Agents

Example – Plan your vacation: Tasks

Tasks



„identify“

Description

...



„gather“

...



„plan“

...

AI Crew

# Agents

## Tools

- can be used by Agents for



Searching the Internet



Scraping Websites



Reading Files

Tool	Description
<code>CodeDocsSearchTool</code>	A RAG tool optimized for searching through code documentation and related technical documents.
<code>CSVSearchTool</code>	A RAG tool designed for searching within CSV files, tailored to handle structured data.
<code>DirectorySearchTool</code>	A RAG tool for searching within directories, useful for navigating through file systems.
<code>DOCXSearchTool</code>	A RAG tool aimed at searching within DOCX documents, ideal for processing Word files.
<code>DirectoryReadTool</code>	Facilitates reading and processing of directory structures and their contents.
<code>FileReadTool</code>	Enables reading and extracting data from files, supporting various file formats.
<code>GithubSearchTool</code>	A RAG tool for searching within GitHub repositories, useful for code and documentation search.
<code>SerperDevTool</code>	A specialized tool for development purposes, with specific functionalities under development.
<code>TXTSearchTool</code>	A RAG tool focused on searching within text (.txt) files, suitable for unstructured data.

...

Source: <https://docs.crewai.com/core-concepts/Tools/#available-crewai-tools>

# Agents

## Memory

- temporary storage of interactions
- enables agents to recall information to current context

### Short-Term Memory

- captures and organizes information on entities, e.g. people, places

### Entity Memory

- preserves valuable insights and outcomes
- allows agents to build up knowledge over time

### Long-Term Memory

- keeps context of interactions
- increases relevance of agent responses

### Contextual Memory

# Agents

## Memory

- implementation is pretty simple
- by default
  - memory is disabled
  - uses OpenAI embeddings

ğsộη çsêxắi iηrộstj Csêx Ağêηtj  
Tắl Rsộcêss

ηy çsêx Csêx  
ắgêηtj  
tắl  
rsộcêss Rsộcêss sêrộêηtjắl  
nêηộsỳ Tsụê  
wêscộê Tsụê

# Agents

## Memory - Benefits

### Adaptive Learning

- crews adapt to new information and refine their approach to tasks

### Enhanced Personalisation

- agents remember user preferences and historical interactions

### Improved Performance

- more informed decisions
- use past learnings and contextual insights

# Agents

Asynchronous Operation

- S

# Agents

## Callbacks

- task callback and step callback
- executed after task or step-completion
- can be used for
  - notifications
  - actions
- parameter passed inside Task

# Agents

## Collaboration

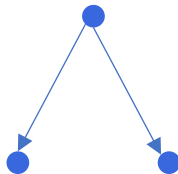
Agents can collaborate on a task to

- share information
- assist on a task
- allocate and optimize resources

Process.sequential



Process.hierarchical



```
from crewai import Agent, Task, Crew,  
Process
```

```
crew = Crew(  
    agents=[planner, writer, editor  
],  
    tasks=[plan, write, edit],  
    verbose=2,  
    manager_llm=llm,  
    process= Process.hierarchical  
)
```



# Agents

## Expected Task Outcome

- output formats can be defined in detail

```
class OutputFormat(BaseModel):  
    chapter_title: str  
    bullet_points: list[str]
```

```
Task(  
    description=(".",),  
    expected_output="A well-written slideset ...",  
    agent=editor,  
    output_format="markdown",  
    output_format_model=OutputFormat,  
    output_format_description=(  
        "The output format is a markdown file ..."  
    ),  
    output_file = "slideset.md"  
)
```

# Agents

## Use of other LLMs

- set up an llm-object
- pass it as a parameter

```
from langchain_groq import ChatGroq

llm=ChatGroq(temperature=0,
              model_name=MODEL,
              api_key=os.environ["GROQ_API_KEY"])

planner = Agent(
    role="...",
    goal="...",
    backstory="...",
    allow_delegation=False,
    llm=llm,
    verbose=True
)
```