Lecture 5:

Agentic AI Systems

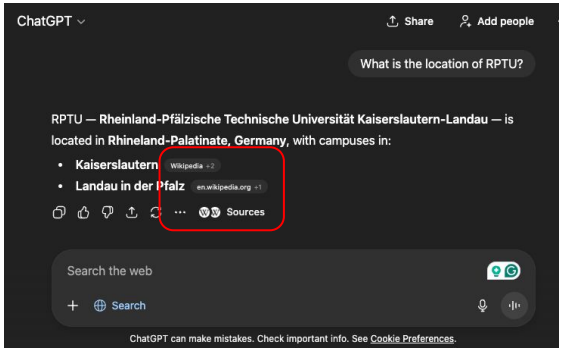
Engineering with Generative AI

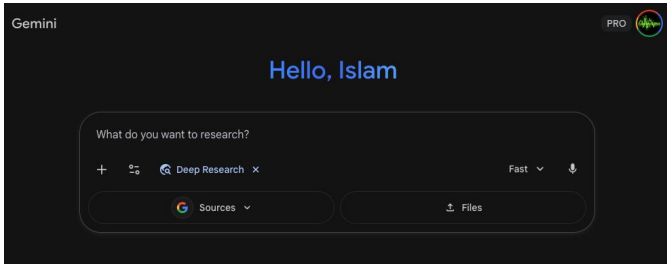
Islam Mesabah, M.Sc.

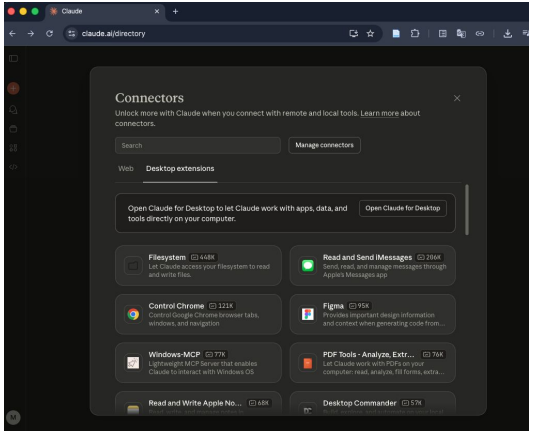
25.11.2025

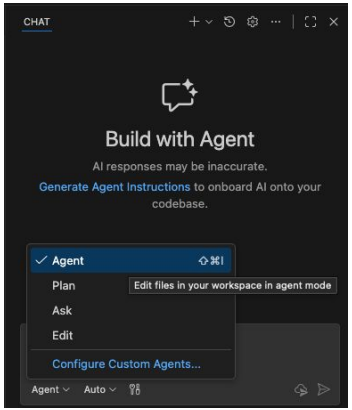
RPTU DSA dfki ai

Who has used an AI agent before?

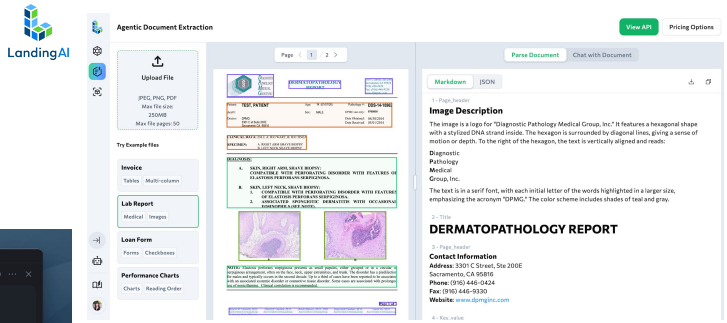


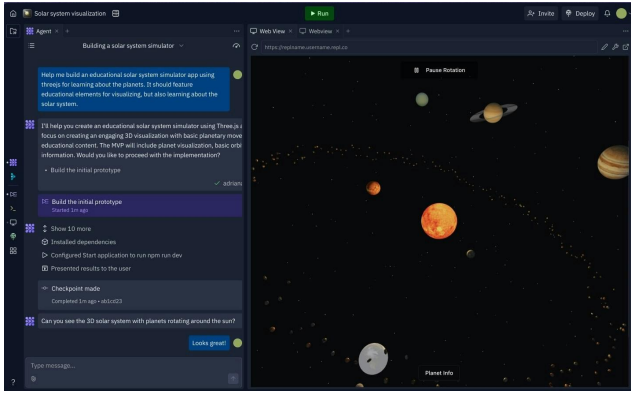


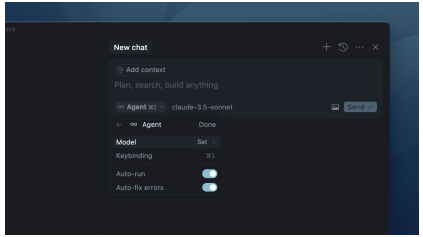


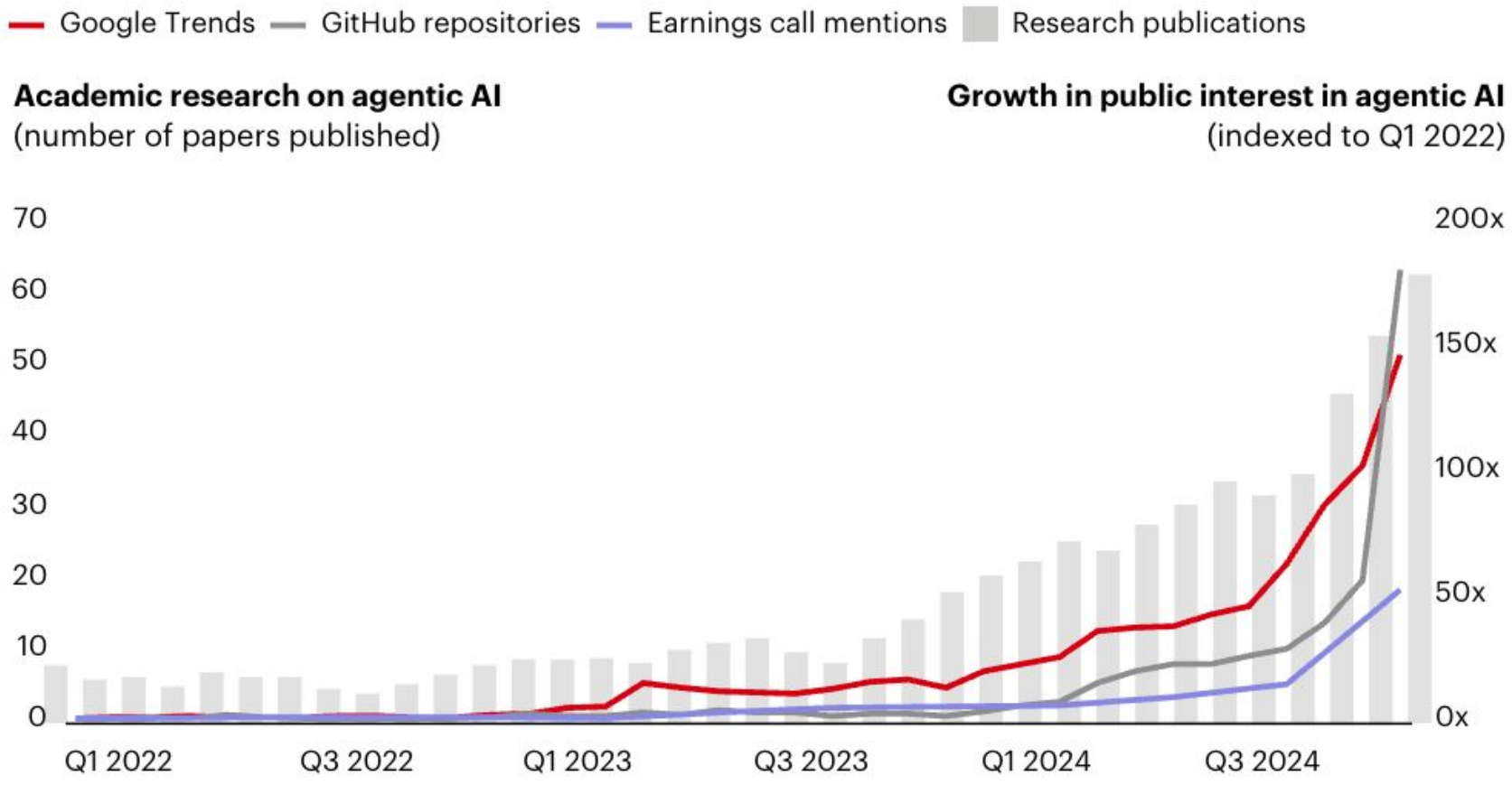


Who has used an AI agent before?









Sources: Google Trends; GitHub; CB Insights; arXiv

https://www.bain.com/insights/what-is-agentic-ai/

An artificial intelligence (AI) **agent** 🤖 is a **system** that **autonomously** performs **tasks** by **designing workflows with available tools**.

AI **agent** 🤖 can encompass a **wide range of functionalities** beyond natural language processing including **decision-making, problem-solving, interacting with external environments and executing actions**.

https://www.ibm.com/think/topics/ai-agents

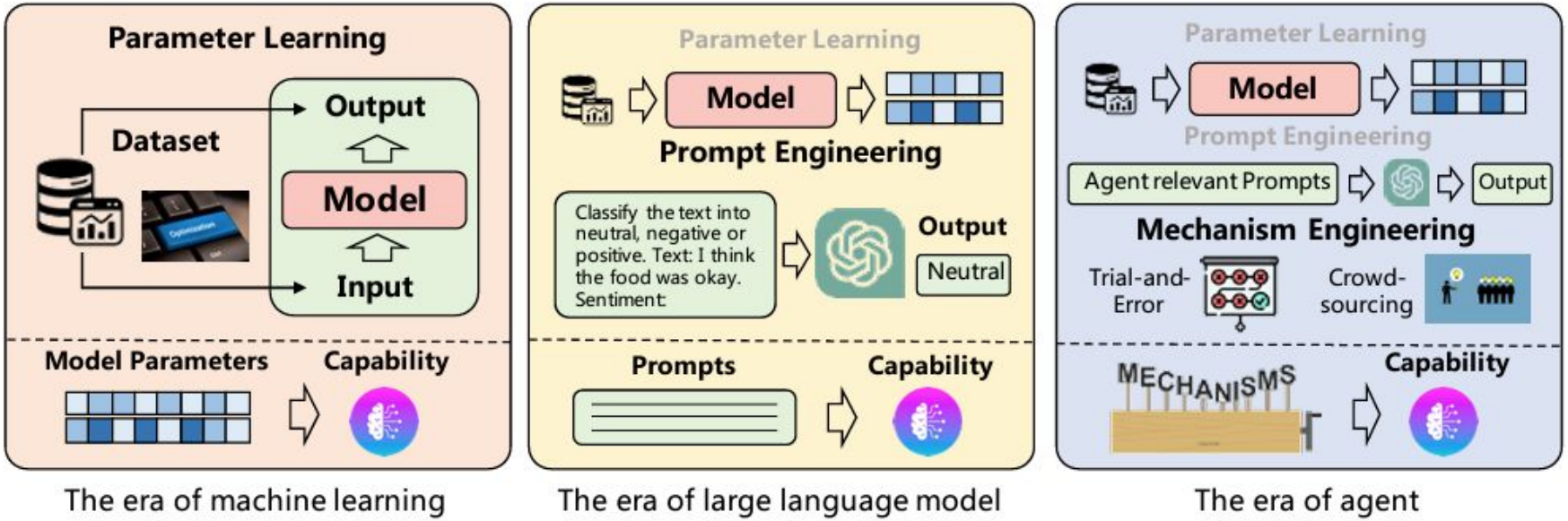
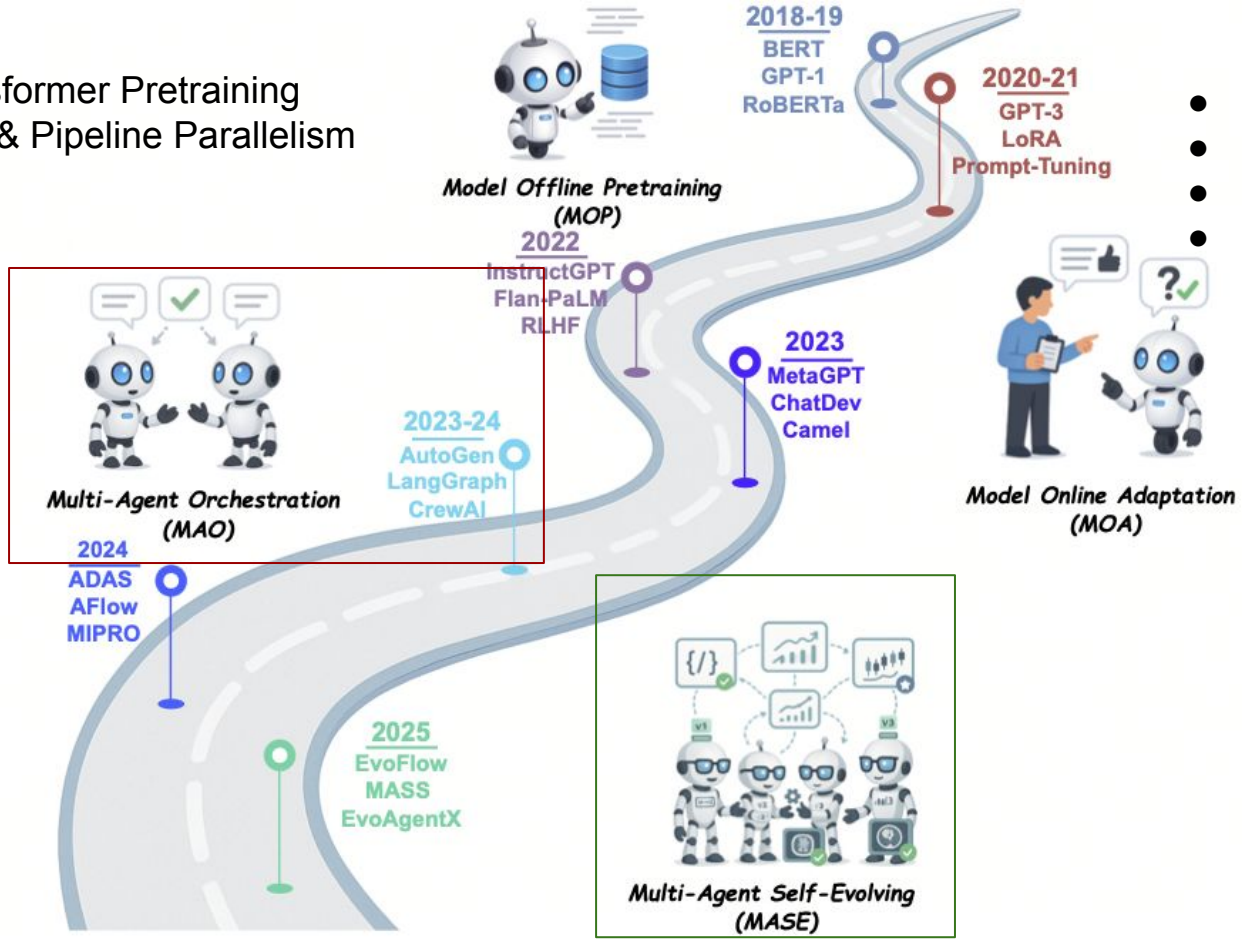


Fig. 4 Illustration of transitions in strategies for acquiring model capabilities.

- Transformer Pretraining
- MoE & Pipeline Parallelism

- Task Fine-tuning
- Instruction Tuning
- LoRA
- RLHF

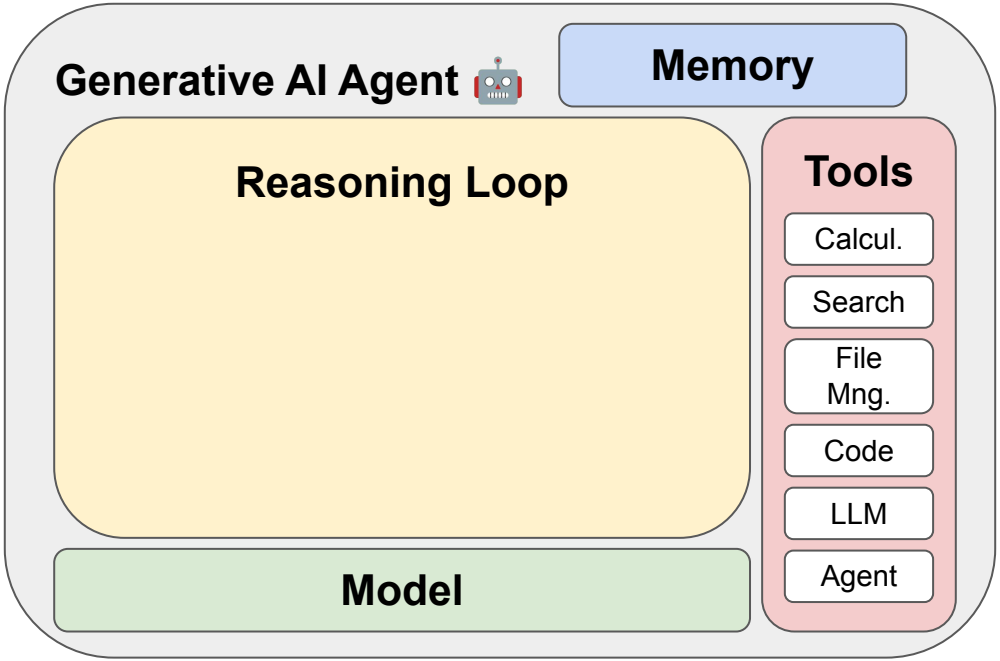
- Transformer Pretraining
- MoE & Pipeline Parallelism



- Task Fine-tuning
- Instruction Tuning
- LoRA
- RLHF

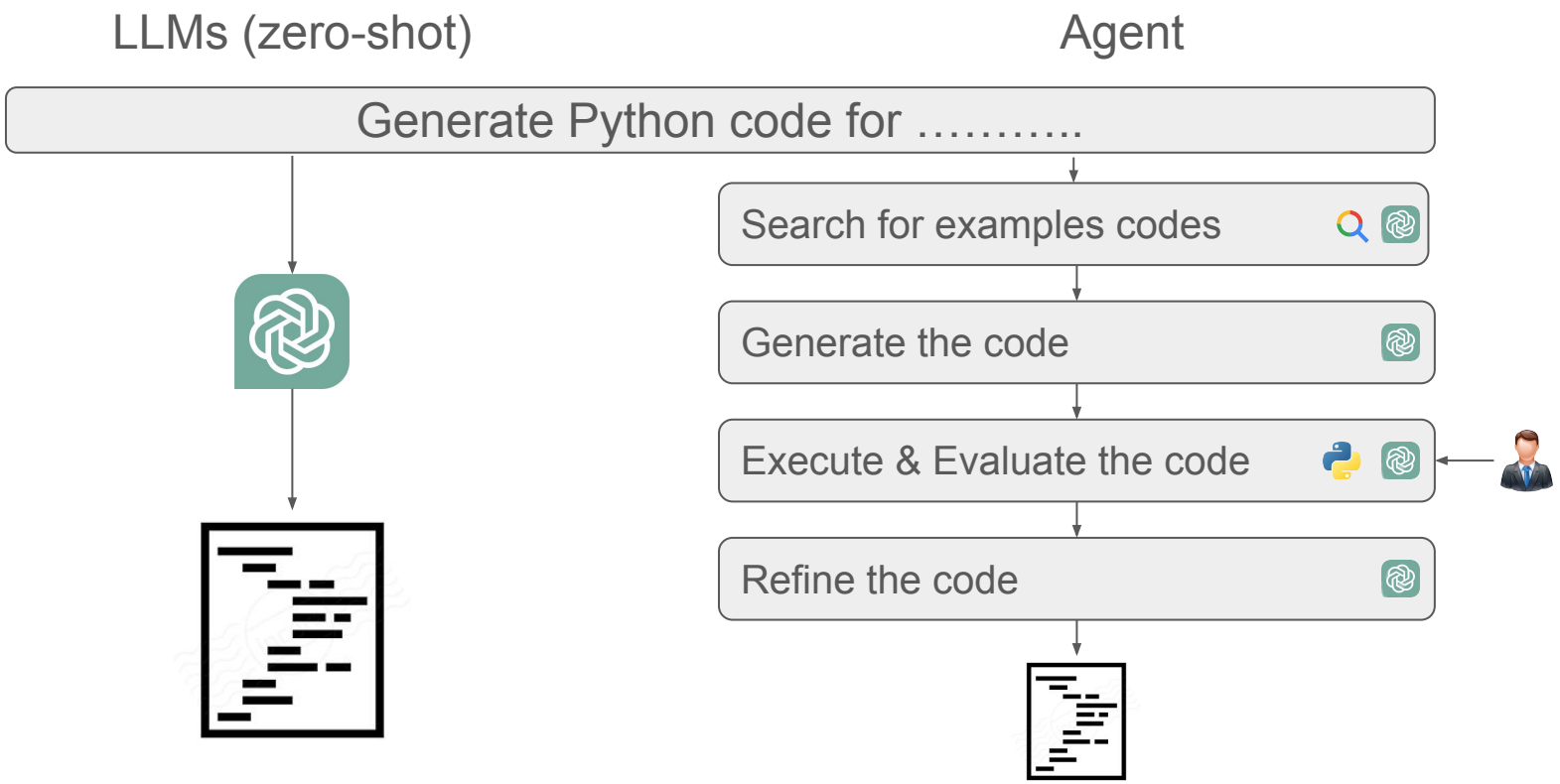
Table of contents

- LLM-based AI Agent
- Agentic AI System
- Agentic AI Frameworks
- Agentic AI Applications
- Agentic AI Evaluation
- Challenges & Limitations



LLM-based AI Agents

AI Agents vs. LLMs



AI Agent - Agent Components

Observe, Act, Achieve

An **AI Agent** 🤖 is an autonomous systems that **leverage** LLMs as the core **reasoning components** for **understanding inputs**, **planning actions**, and **generating outputs** in open-ended, real-world environments

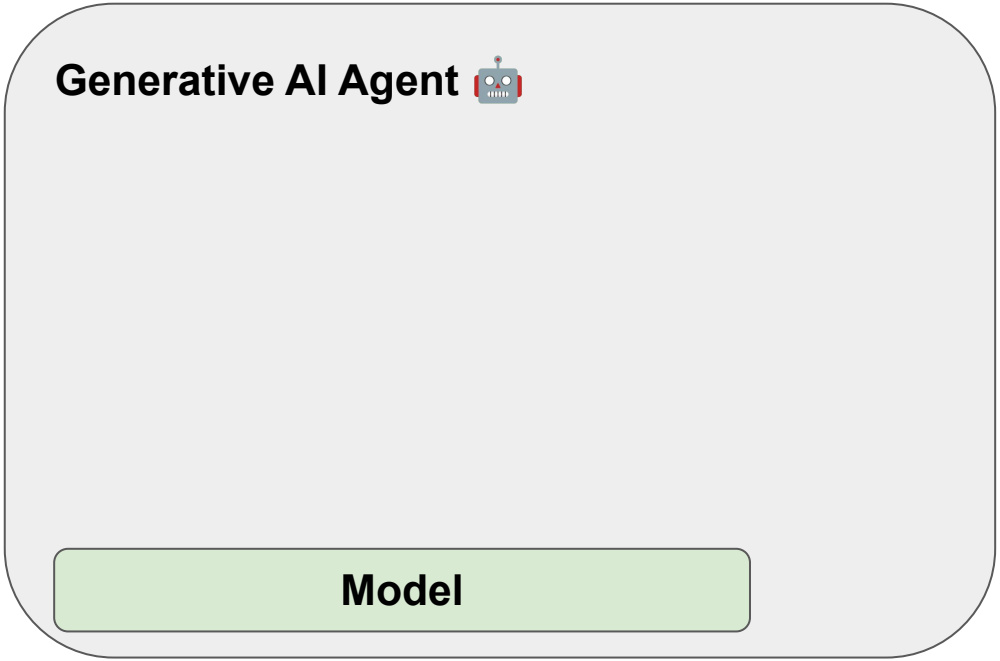
http://arxiv.org/abs/2508.07407

Agent Components

Generative AI Agent 🤖

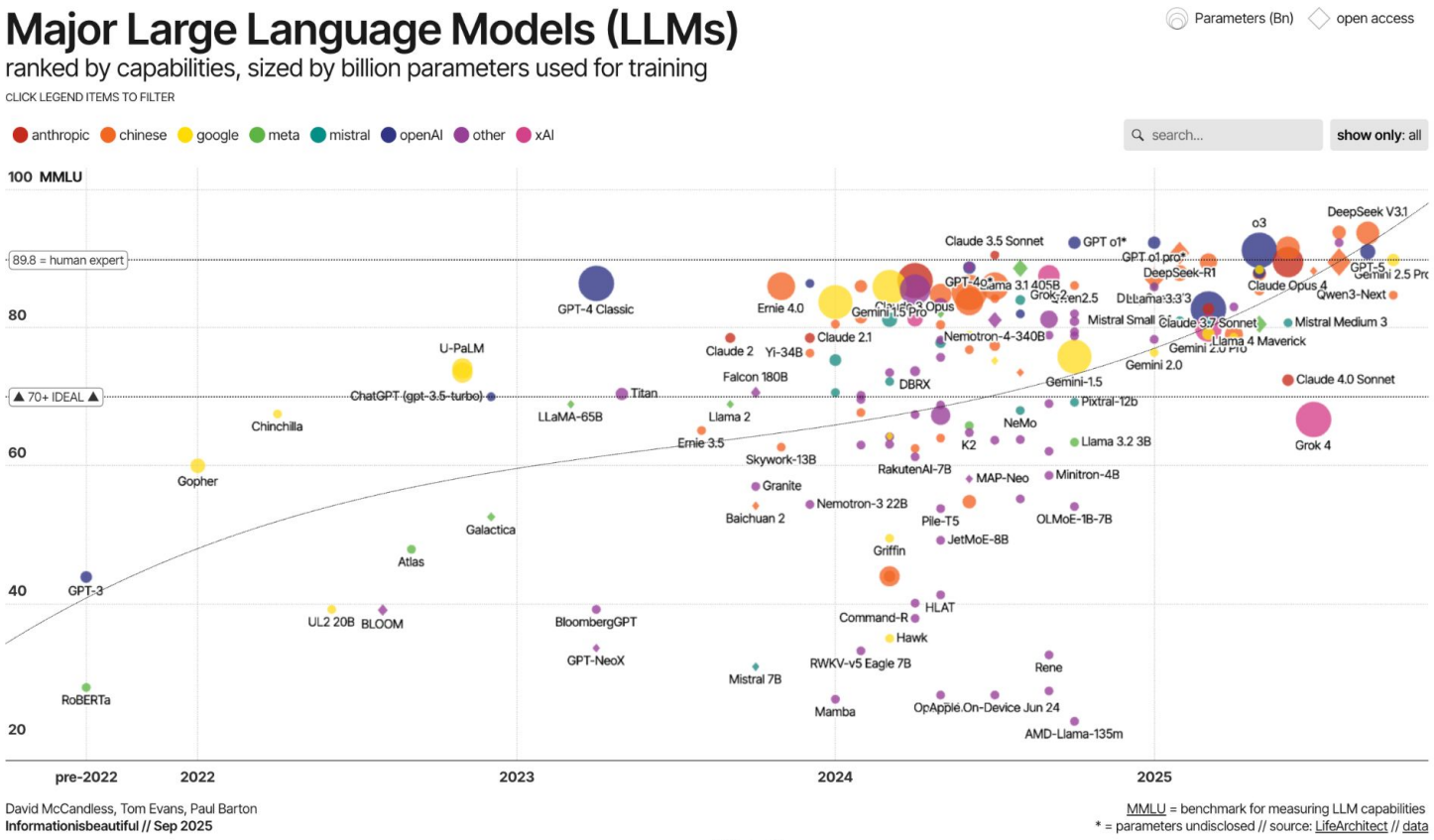
Agent Components - Model

- Coordinator (usually an LLM) acts as the main **brain** of the agent
- Variety of **open and closed source** LLMs
- Ex:
  - Ruled-Based code
  - GPT5
  - Claude
  - Llama
  - Mistral
  - Qwen
  - …



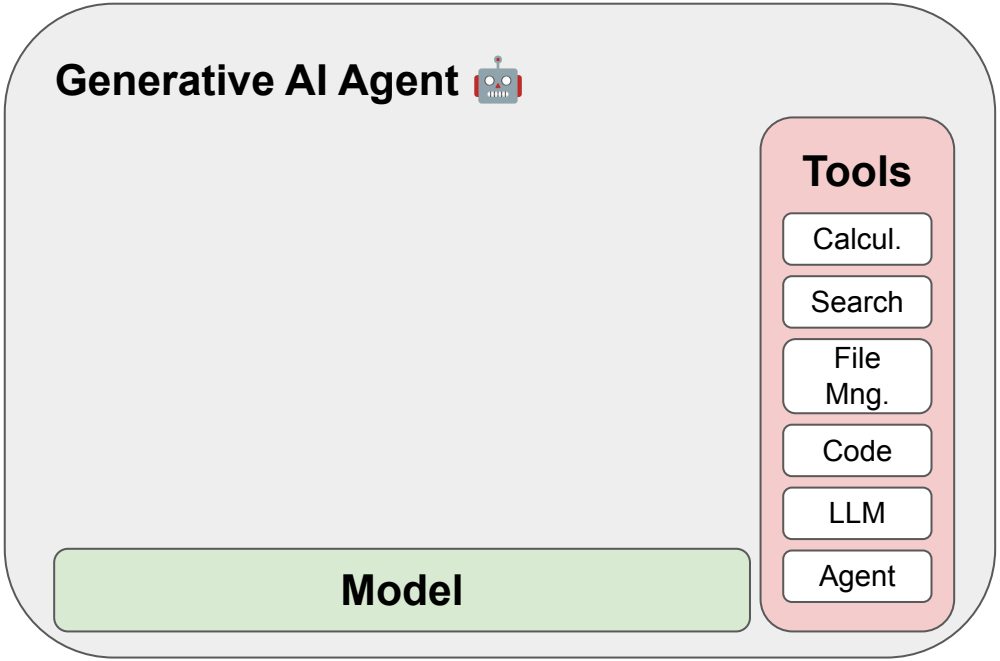
https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt

Agent Components - Model



Agent Components - Tools

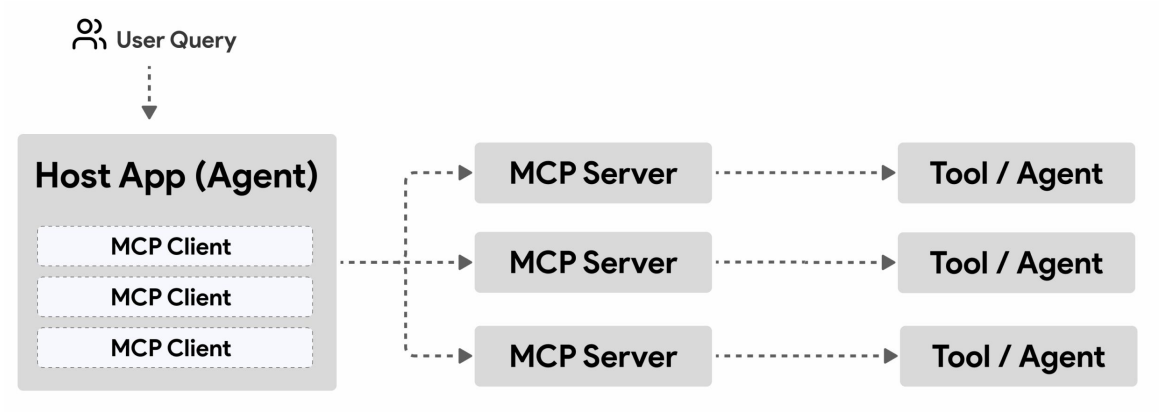
- Enable the model to **interact** with the task environment
- This interaction helps the agent **access up-to-date information** for subtasks
- Includes databases or knowledge bases.
- Includes other **LLM** models
- Best Practices is to use **Model Context Protocol (MCP)**



Agent Components - Tools - **Model Context Protocol (MCP)**

ANTHROP\C

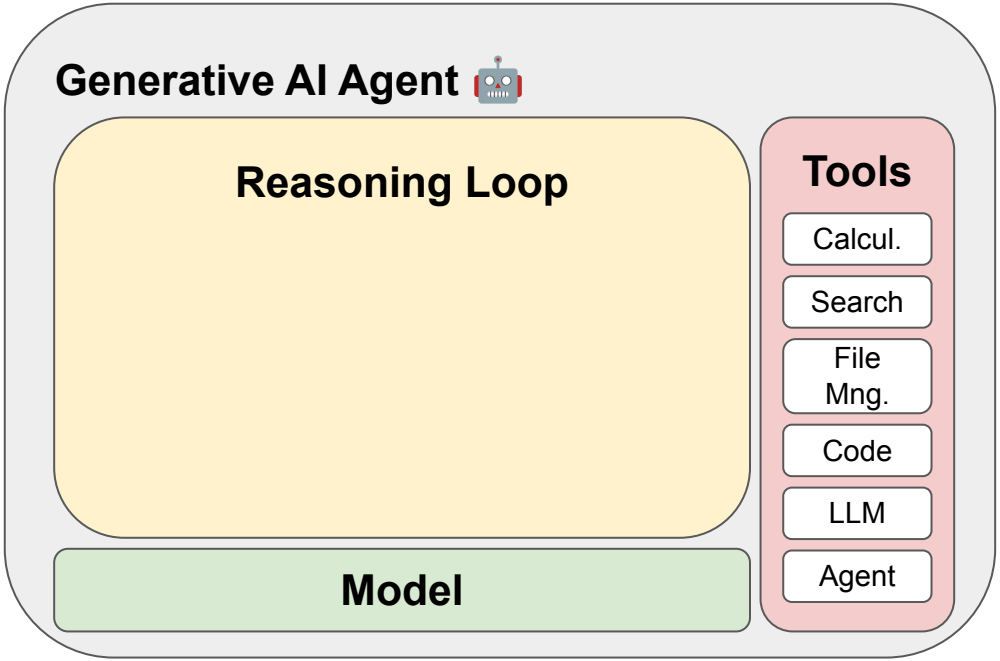
An **open standard** by **Anthropic** that enables developers to build **secure, two-way connections** between their **data sources** and **AI-powered** tools.



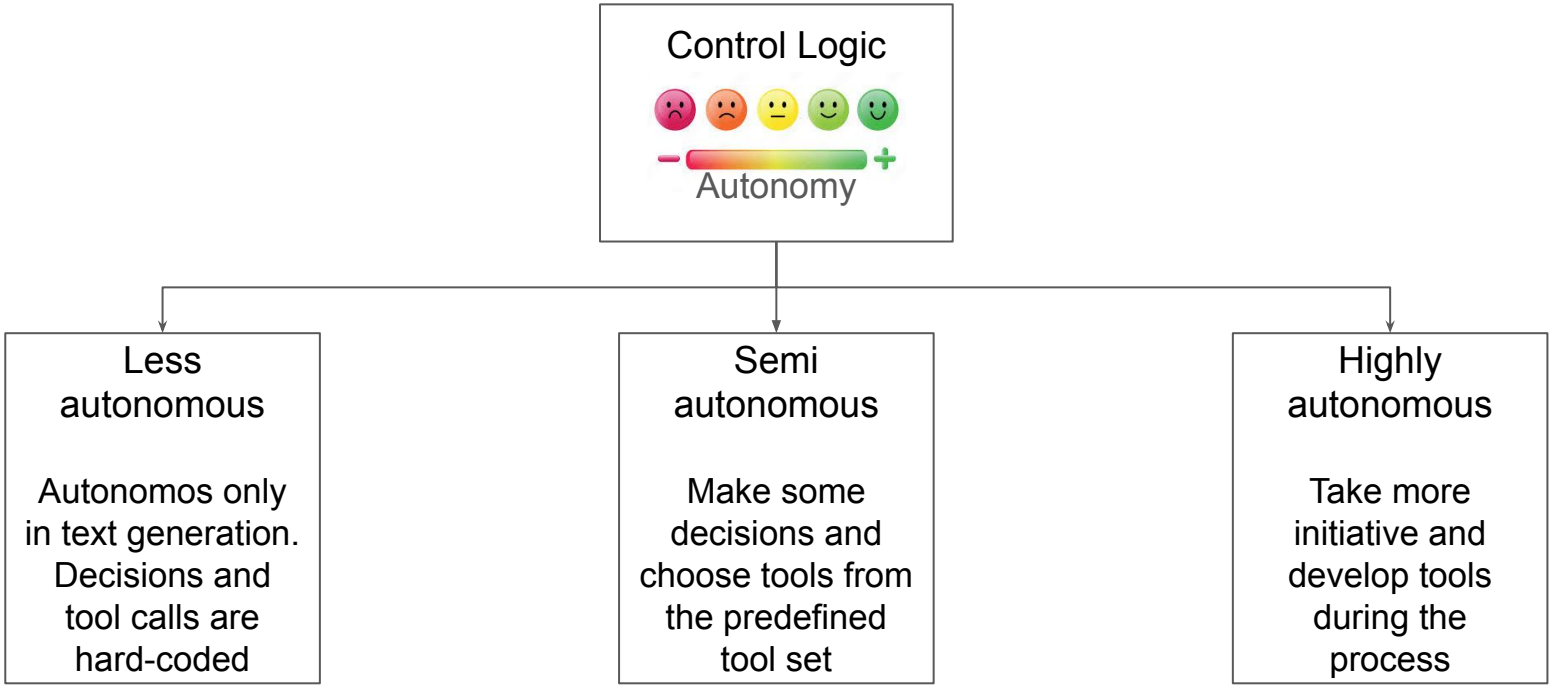
https://www.anthropic.com/news/model-context-protocol

Agent Components - Reasoning Loop

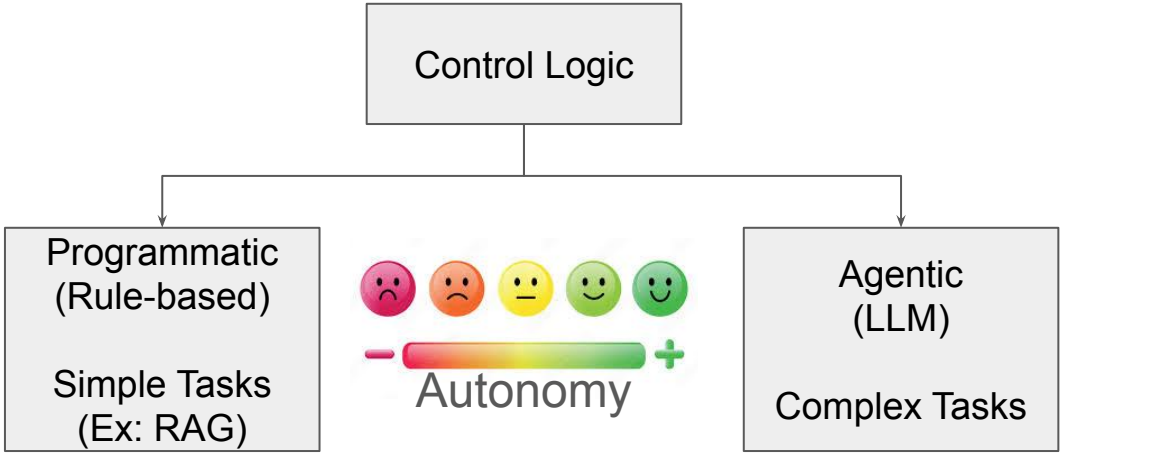
- LLM's ability to **think** about task oriented actions
- Break the task to **subtasks** for the agent to achieve its final goal
- The agent addresses each subtask **individually**.



Reasoning Loop - Autonomy Level



Reasoning Loop - Autonomy Level



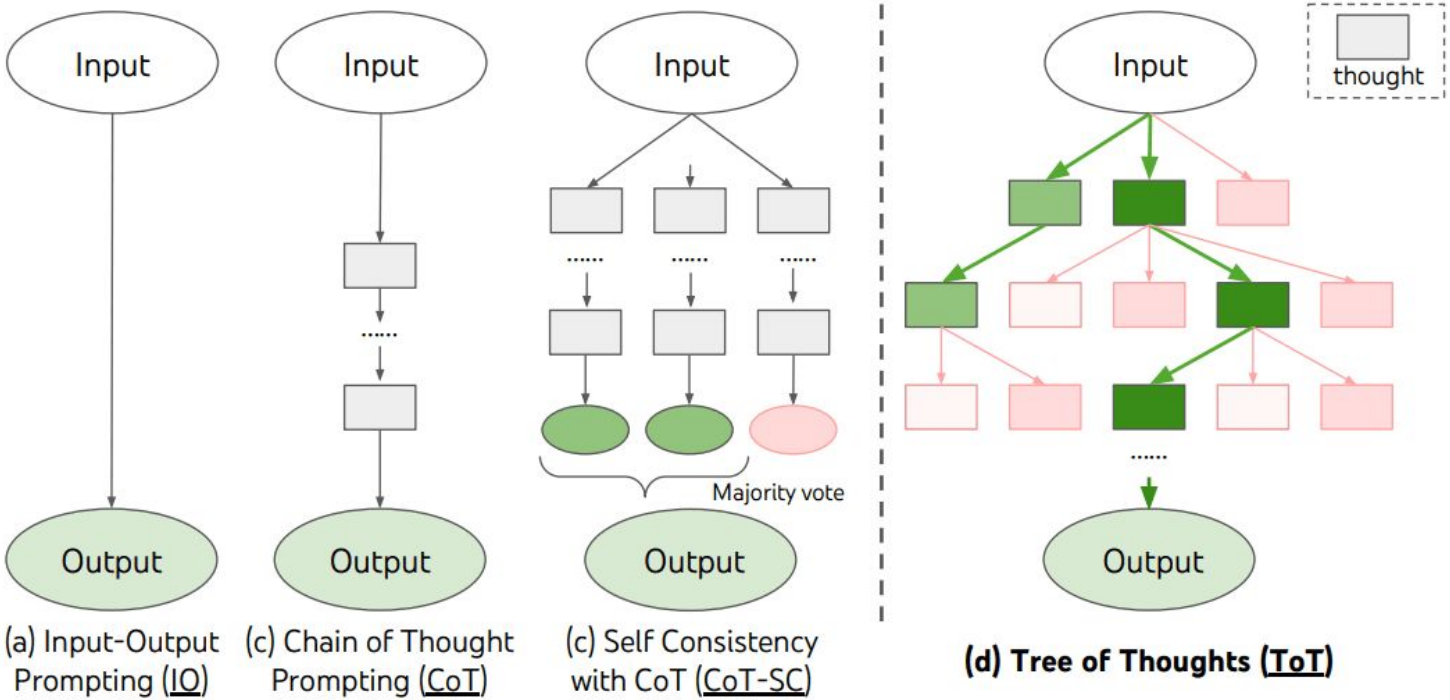
- Conditional logic
- Search algorithm

1. Chain-of-Thought
2. Tree-of-Thoughts
3. ReAct
4. Reflexion

Agentic - Reasoning Loop

- **Reasoning Without Feedback**
  - Use LLM to create a detailed plan with subtasks to achieve the final goal
  - The LLM execute the plan in steps without reflection
  - This introduce a challenge in achieve long-term goal for complex task
  - Examples:
    - Chain-of-Thought
    - Tree-of-Thoughts
- **Reasoning With Feedback**

Agentic - Reasoning Loop - ToT



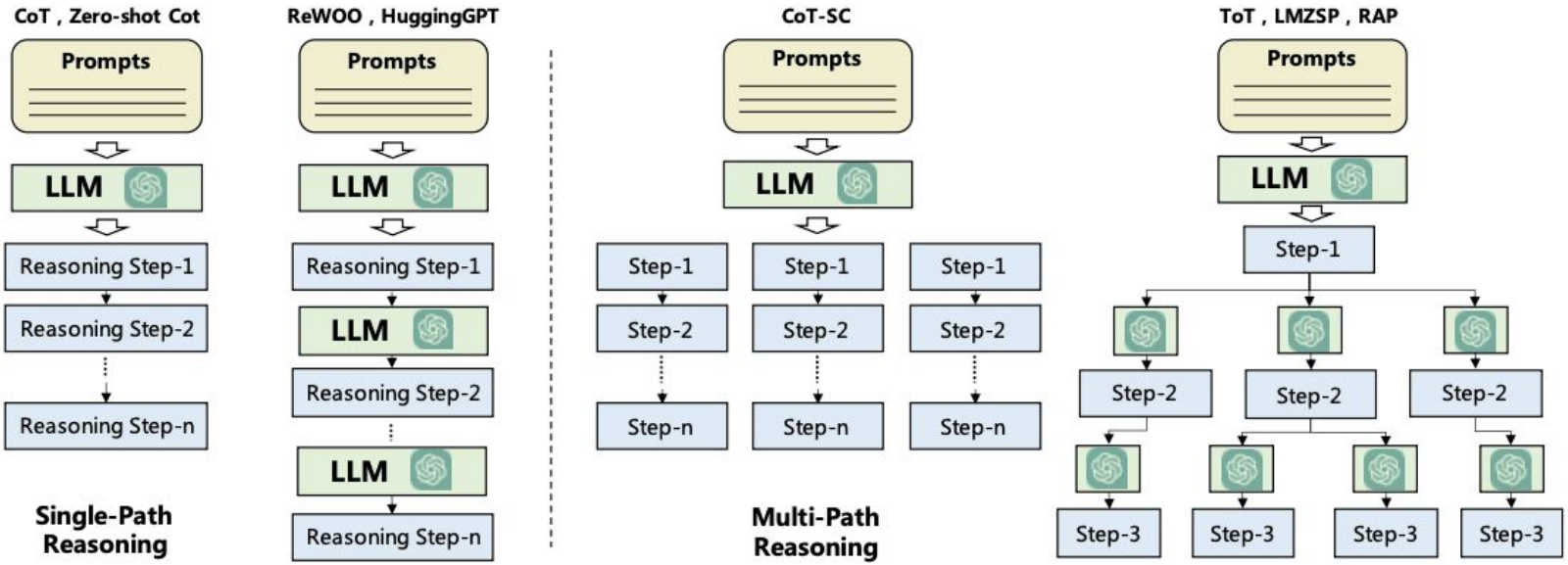
(a) Input-Output Prompting (IO)

(c) Chain of Thought Prompting (CoT)

(c) Self Consistency with CoT (CoT-SC)

(d) Tree of Thoughts (ToT)

Agentic - Reasoning Loop



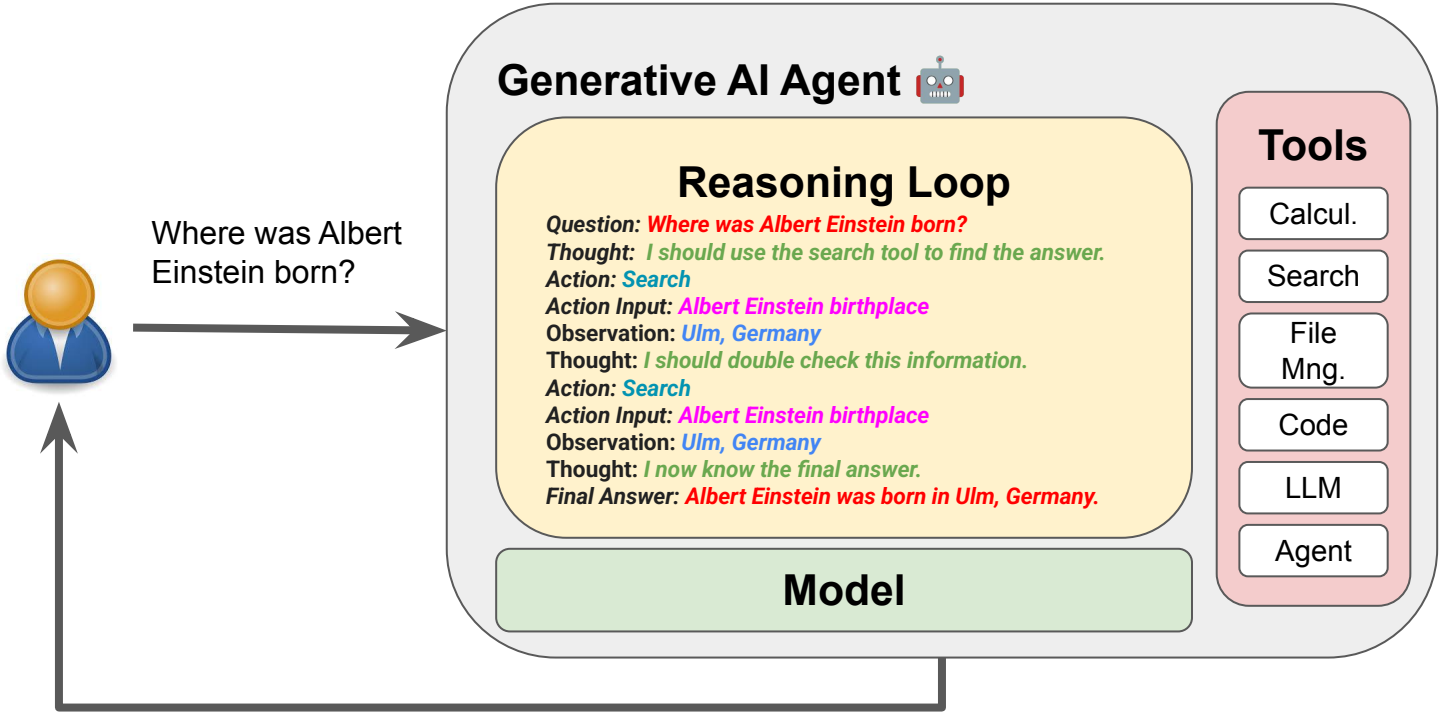
Agentic - Reasoning Loop

- **Reasoning Without Feedback**
- **Reasoning With Feedback**
  - **Iteratively** reflect and refine the **execution plan** based on **past actions and observations**
  - Correct the **mistakes** taken to enhance the quality of the final results
  - Examples:
    - ReAct
    - Reflexion

Agentic - Reasoning Loop - ReAct

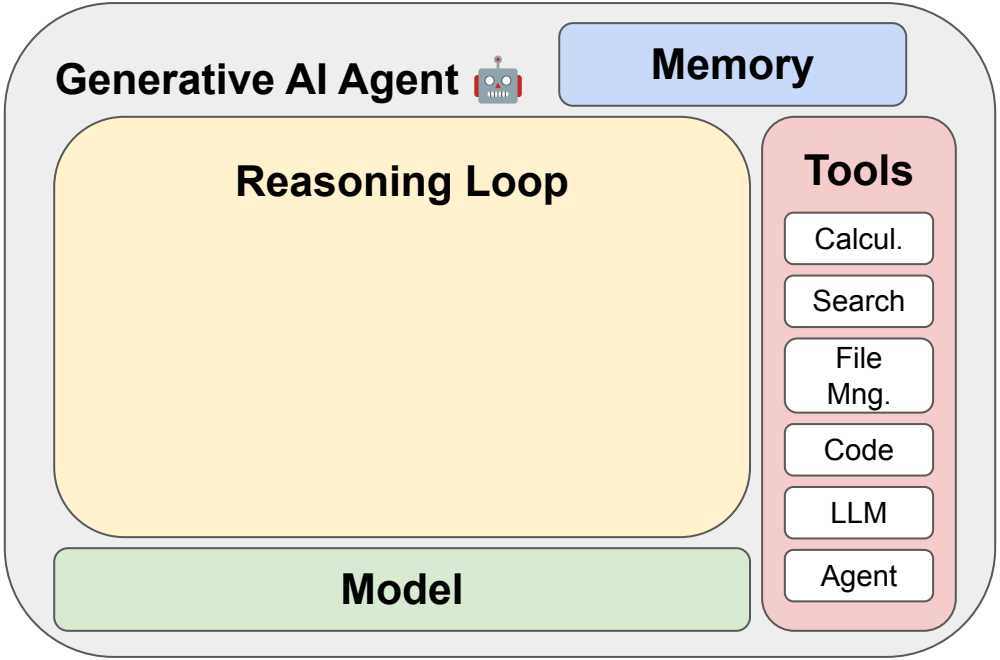
- Use **reasoning and acting** to solve complex tasks
- Repeating steps N times:
  - **Thought**: Generate verbal reasoning on the status
  - **Action**: Take actions using the set of tools available based on the generated thought
  - **Observation**: Feedback from the environment. Other types of feedback can be included as human and model feedback.

ReAct Example



Albert Einstein was born in Ulm, Germany.

Agent Components - Memory



Agent Components - Memory

- Store **internal logs** of the agent (thoughts, actions, observation and user interactions)
- Three types:
  - Short term memory:
    - **Current status** of the agent
    - Short and finit due to context window constraint of the model
    - Improve agent ability of **long range reasoning**
  - Long term memory:
    - Include **all the past behaviors** of the agent (Also past sessions)
    - Stored in **vector store** for fast relevant information retrieval for the agent
    - Improve agent ability for **accumulating experience**
  - Hybrid memory:
    - Include both short and long term memory

Agentic AI System

Agentic AI System

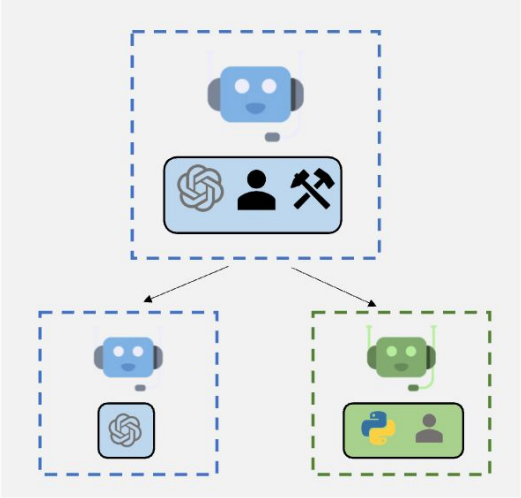
Agentic AI 🤖 is an **artificial intelligence** system that can **accomplish a specific goal** with limited supervision.

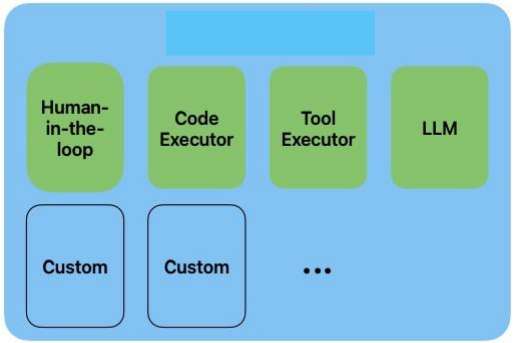
It consists of AI agents 🤖 that **mimic** human decision-making to **solve problems** in real time.

In a multiagent system, each agent **performs a specific subtask** required to reach the **goal** and their efforts are coordinated through **AI orchestration**.

Agents 🤖 Customization

- **Customize agents** to the final common goal:
  - Each agent has a specific role and specializes in performing a particular task in the system.
  - We can control the logical autonomy of the agent
  - Each agent is provided with a set of tools to serve its task
  - The agent could allow human interaction based on its task [Human-in-the-loop]

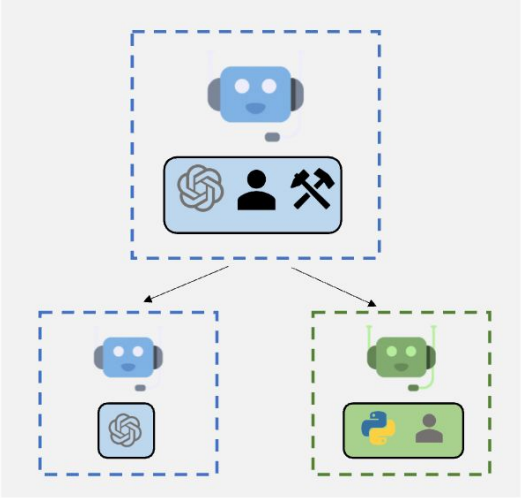


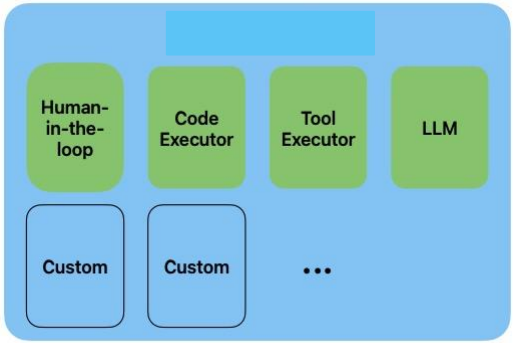


Agents 🤖 Customization

- **Customize agents** to the final common goal:
  - Each agent has a specific role and specializes in performing a particular task in the system.
  - We can control the logical autonomy of the agent
  - Each agent is provided with a set of tools to serve its task
  - The agent could allow human interaction based on its task [Human-in-the-loop]

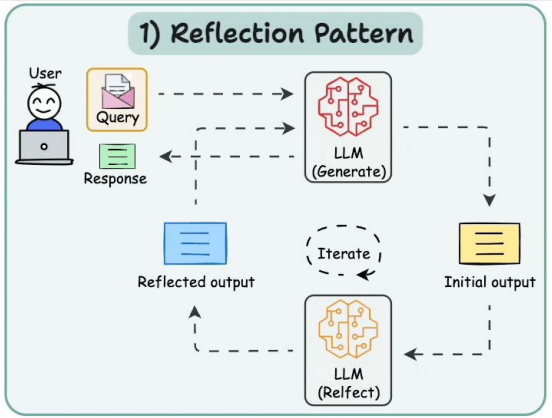
- Agents can **make decisions** based on their role in the system to achieve the common goal

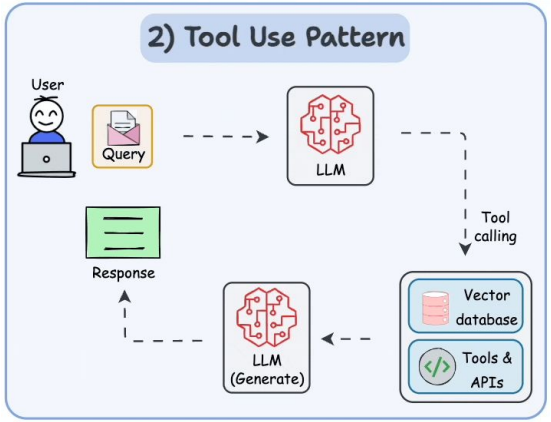


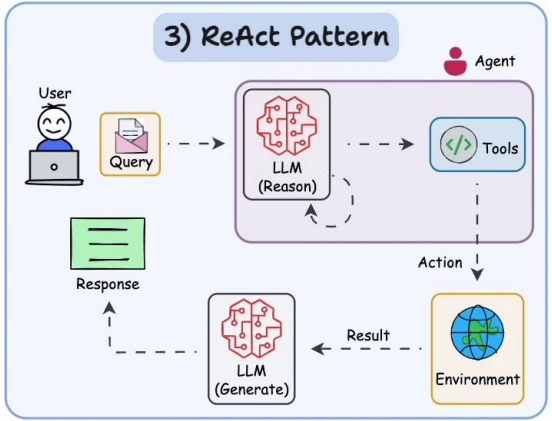


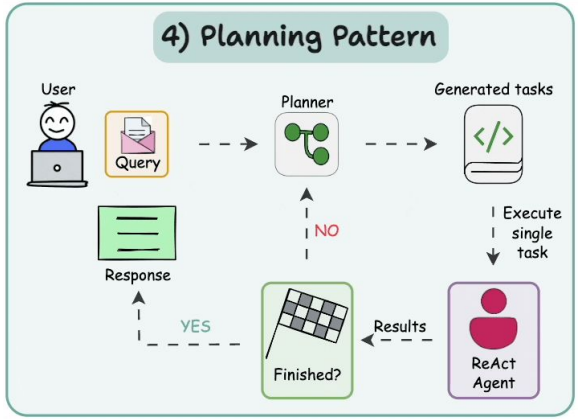
https://blog.dailydoseofds.com/p/5-agentic-ai-design-patterns

Single Agent - Design Patterns

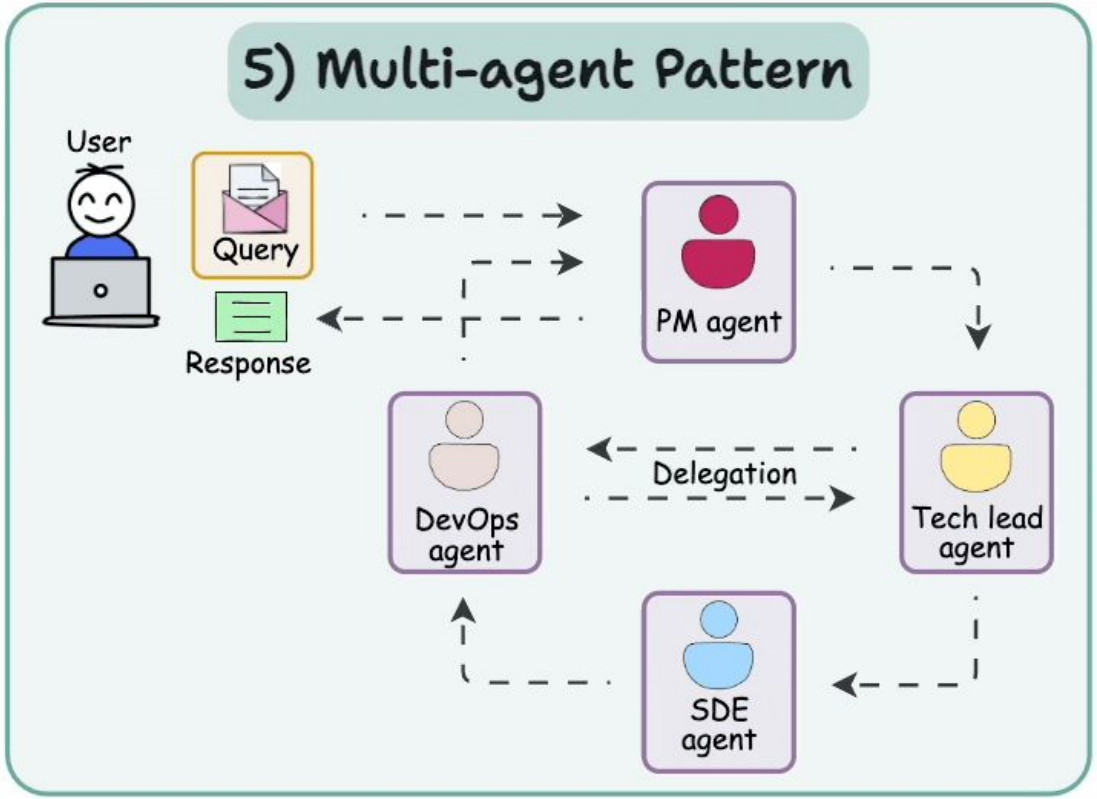








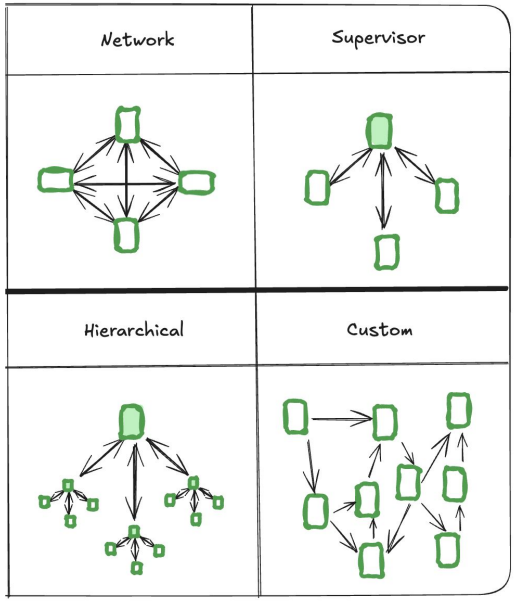
Multi-Agent - Design Pattern



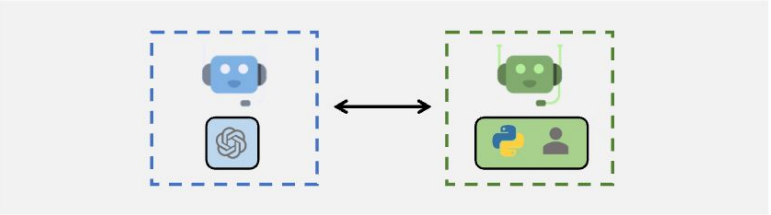
Agents 🤖 Communication

- An agent can **exchange** messages with other agents to **start or maintain a conversation.**

- The System designer could decide the **number of agents** in the system and the **pattern** they take for conversation.
  - Two agent Chat
  - Sequential Chat
  - Group Chat
    - Joint Chat
    - Hierarchical Chat

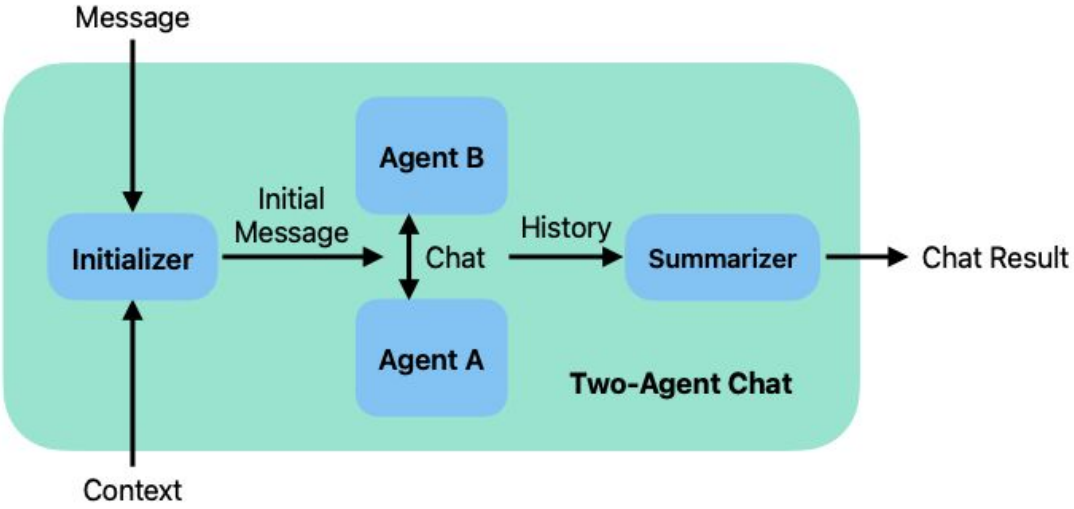


Two agent Chat



Multi-Agent Conversations

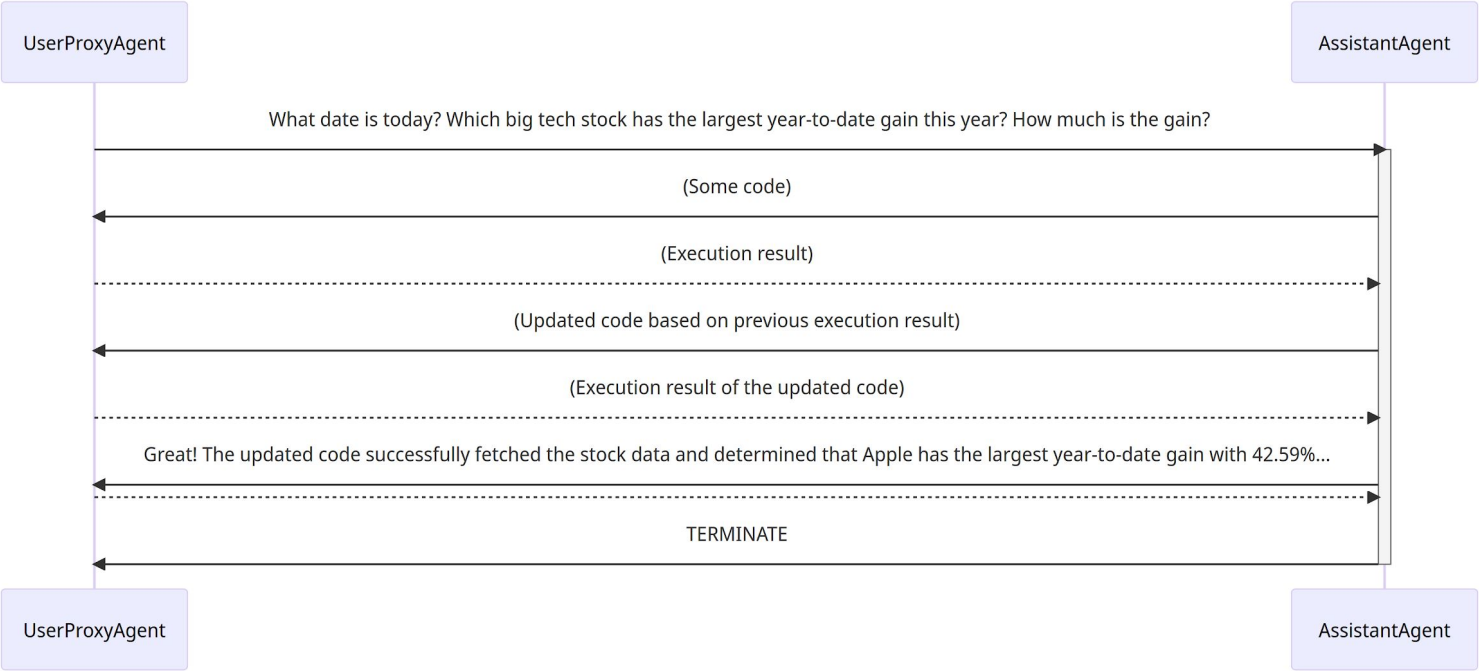
E.g., "What is triangle inequality?"



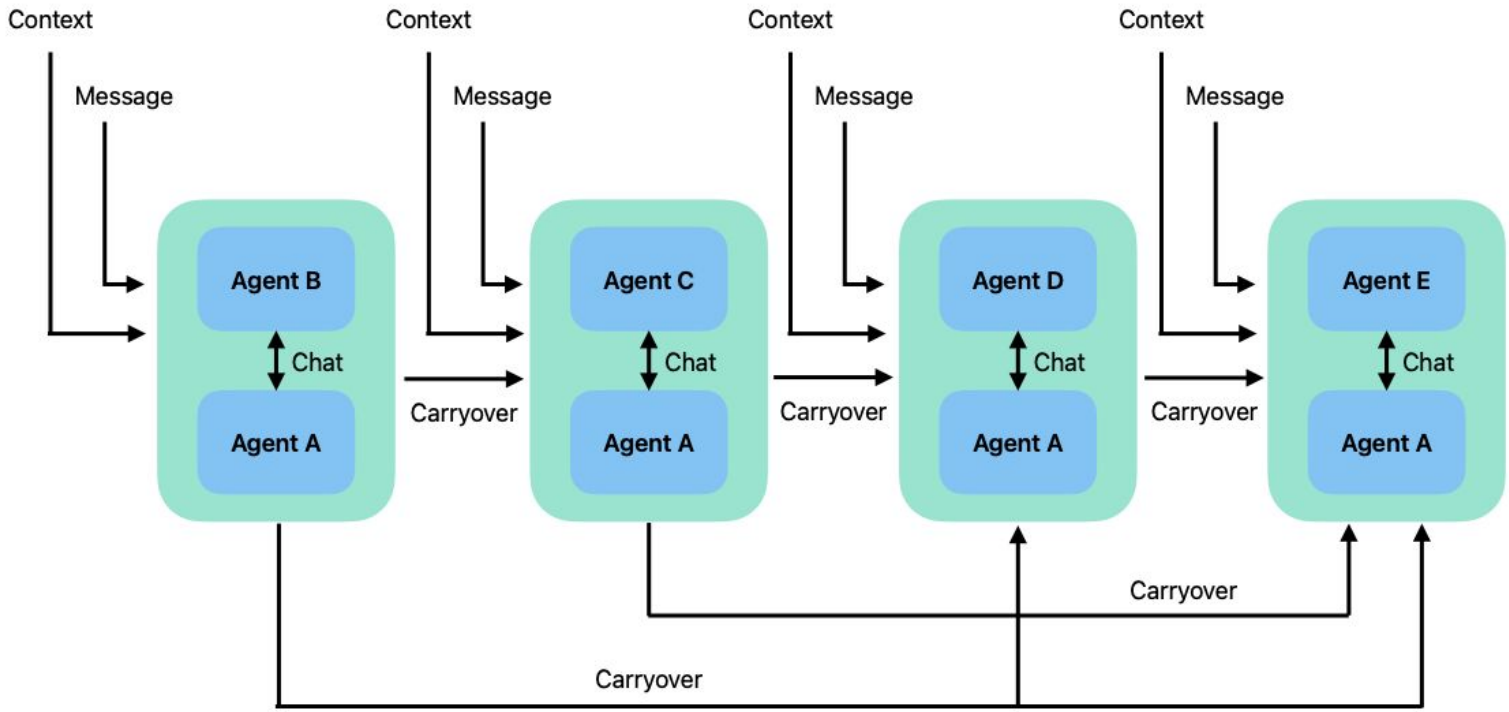
E.g., summary_method: reflection_with_llm

Two agent Chat

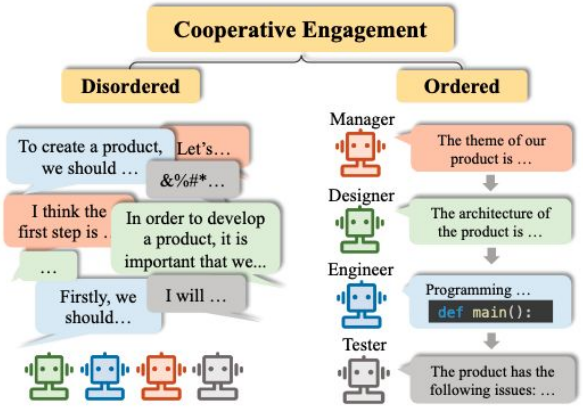
- An example conversation between two agents

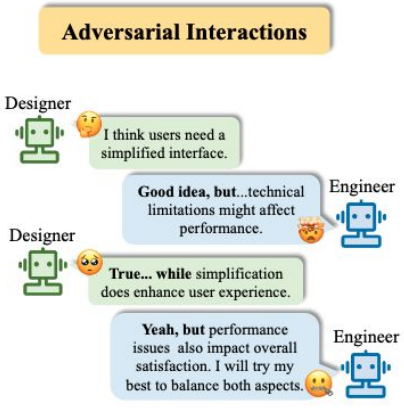


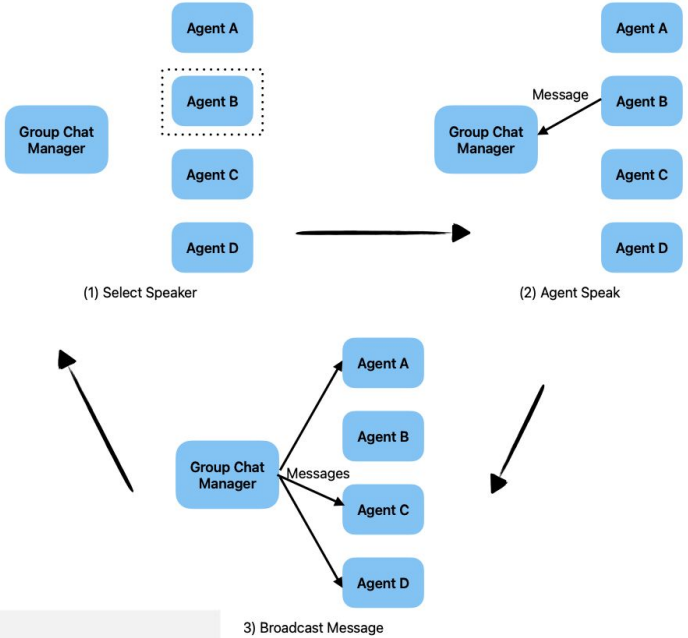
Sequential Chat

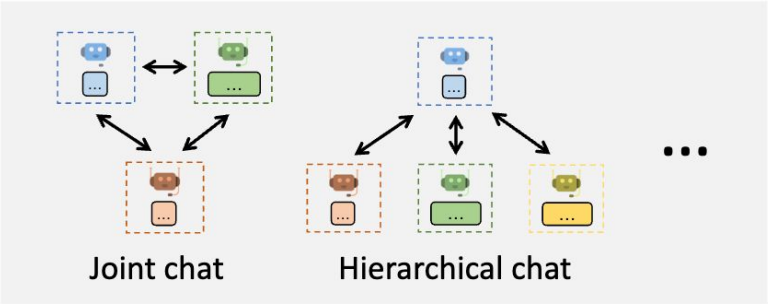


Group Chat





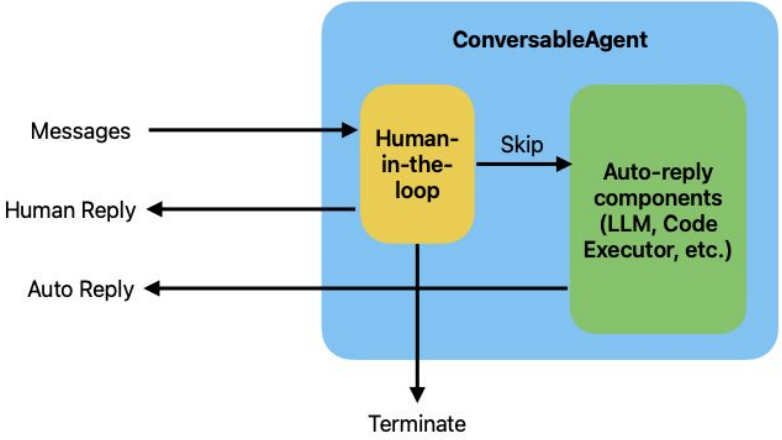




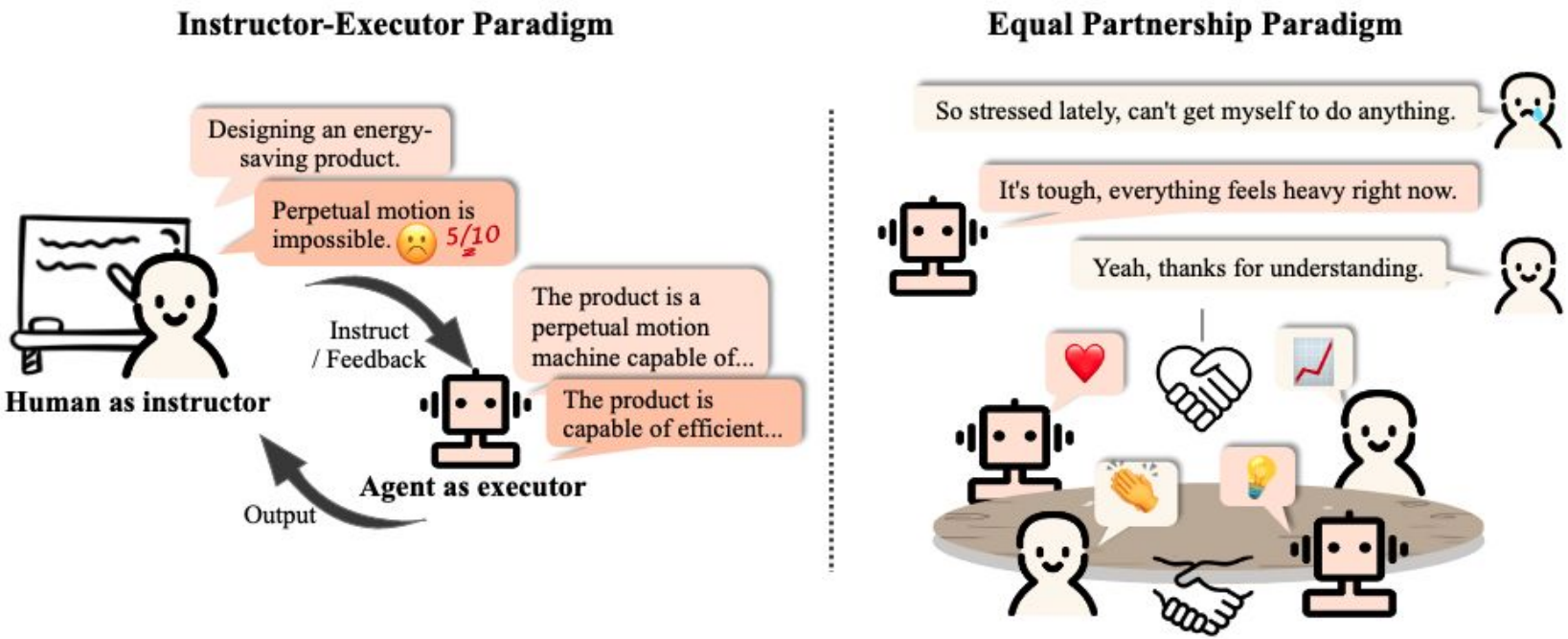
Flexible Conversation Patterns

Human-in-the-loop

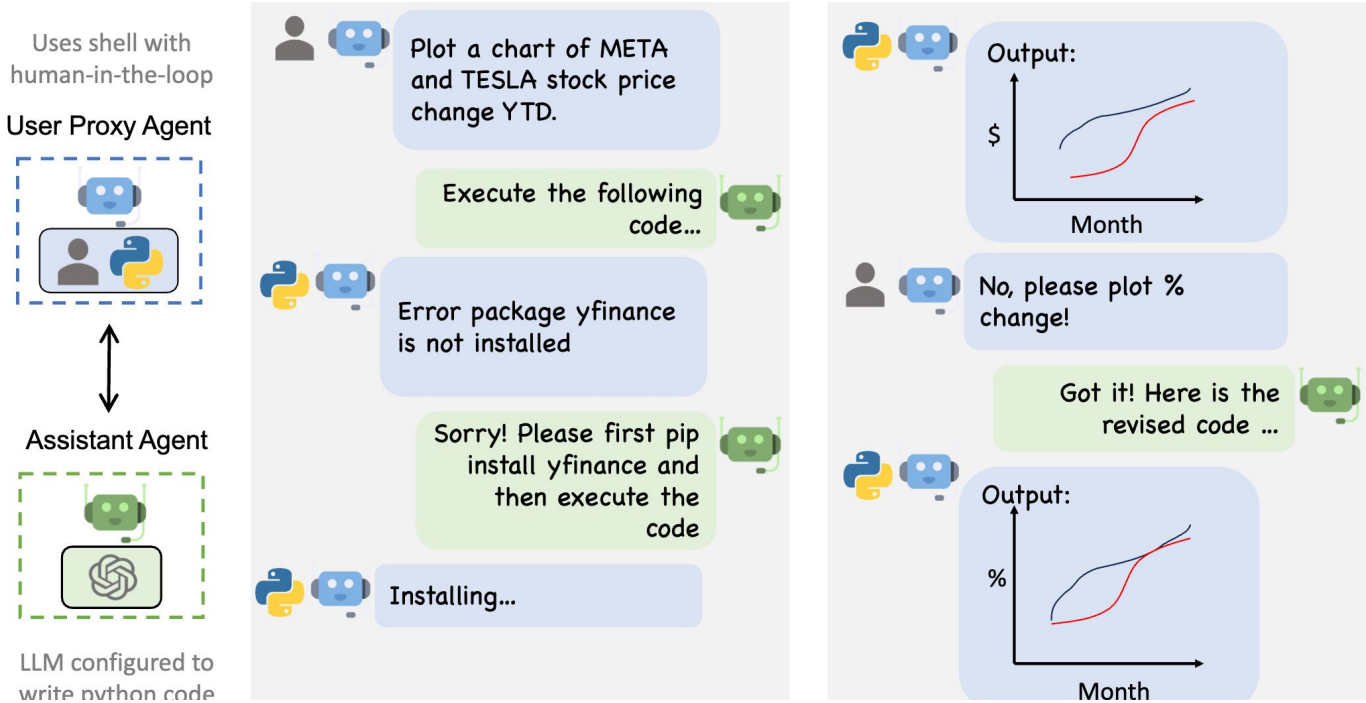
- Autonomous Agent can accomplish complex applications
- Many applications require putting **humans in the loop** with agents
  - To allow human feedback on intermediate results to help agents
  - Specify new goals
- Humans in the loop come in between two auto-replay agents.
- It takes the coming message and decides whether to pass it to the agent or provide human feedback



Human-in-the-loop



Human-in-the-loop



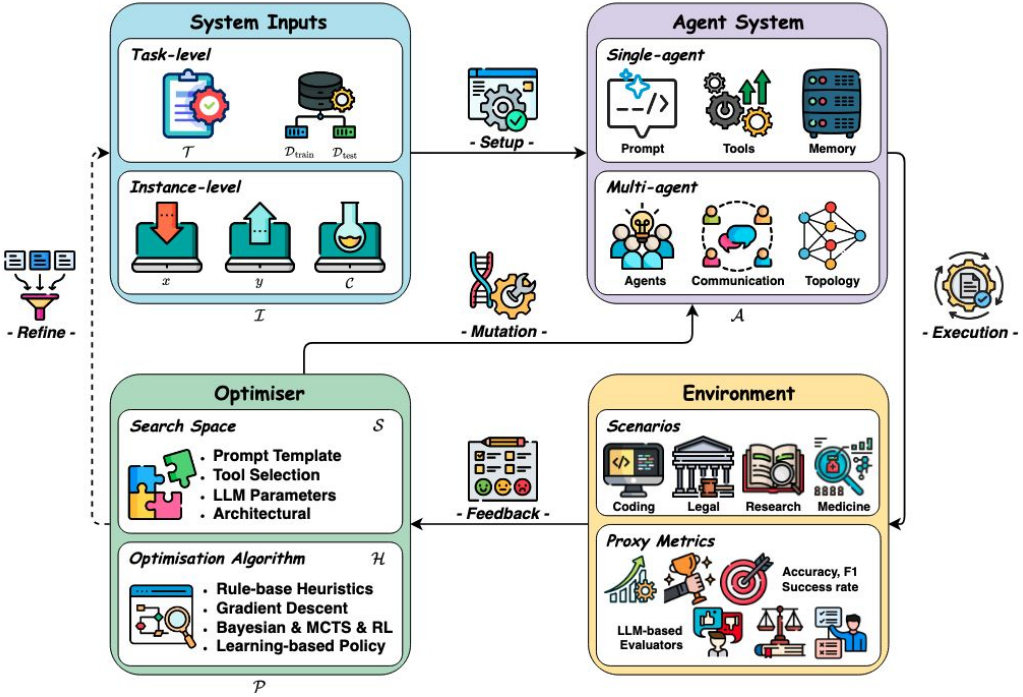
http://arxiv.org/abs/2508.07407

Self-evolving AI agents

Self-evolving AI agents are **autonomous systems** that **continuously and systematically optimise their internal components** through **interaction with environments**, with the goal of **adapting to changing tasks, contexts and resources** while preserving **safety and enhancing performance**.

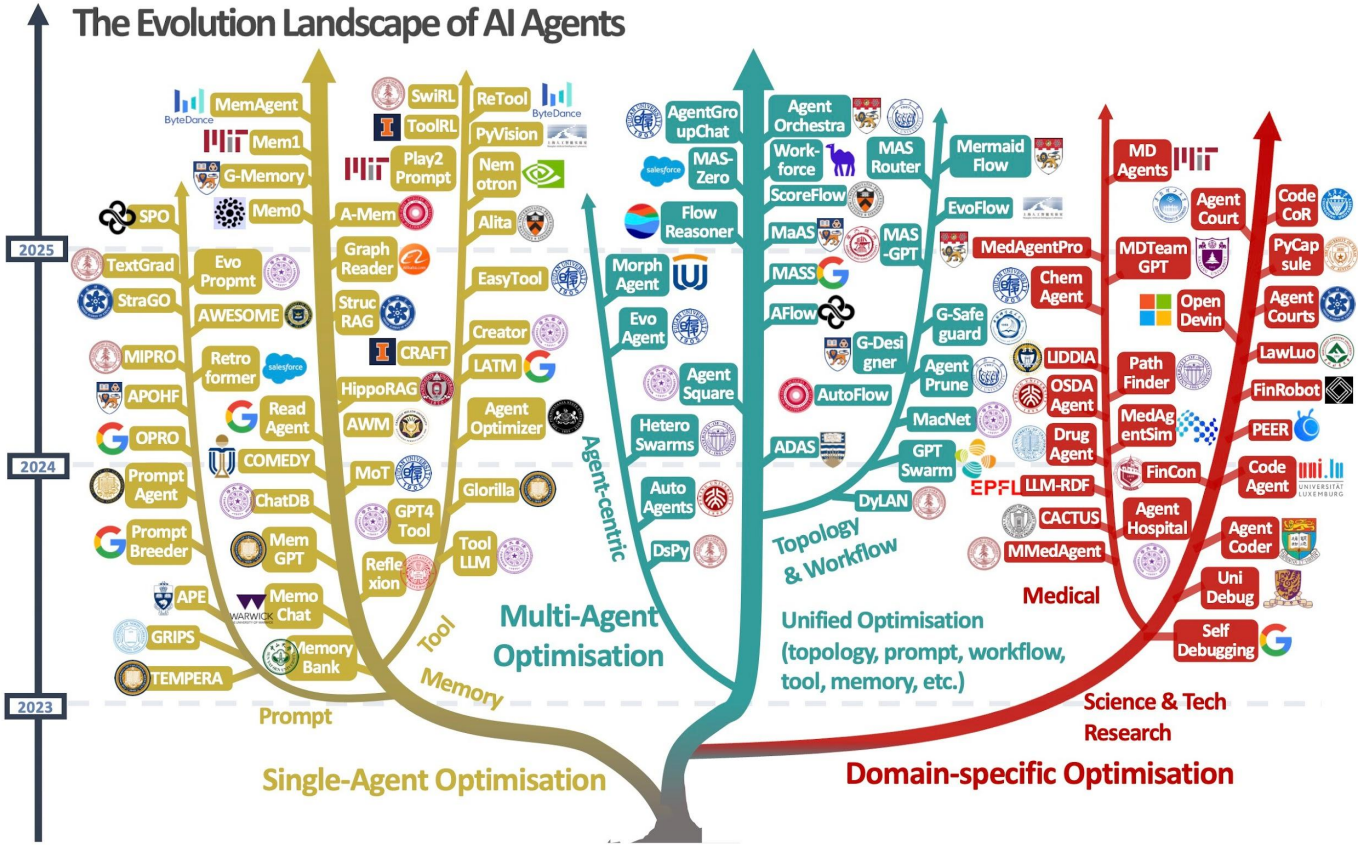
The optimization include:

- Behaviour Optimisation
- Prompt Optimisation
- Memory Optimisation
- Tool Optimisation
- Agentic Workflow Optimisation



http://arxiv.org/abs/2508.07407

Self-evolving AI agents



LANGROID

MULTI-AGENT FRAMEWORK
FOR LLM APPLICATIONS

LangChain

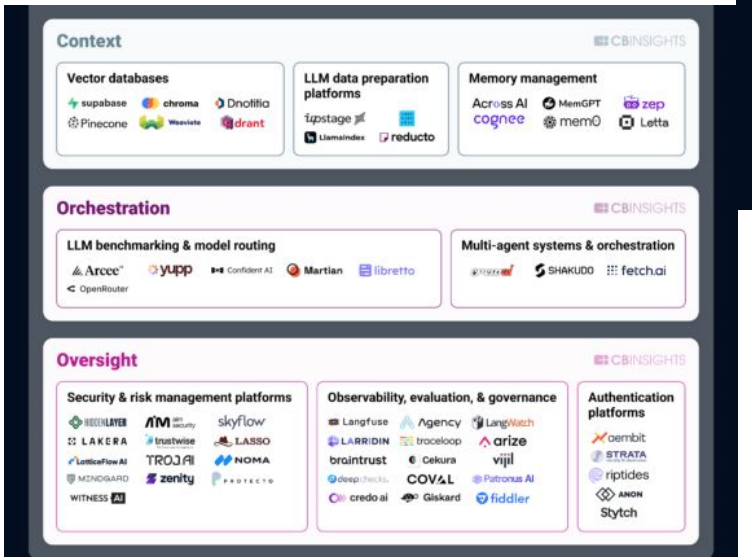
Microsoft

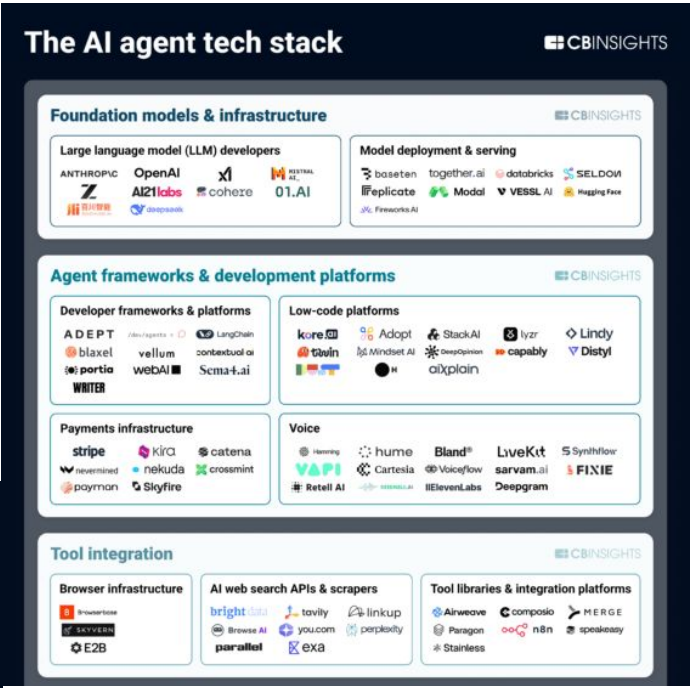
AutoGen

Agentic AI Frameworks

AI Agent Frameworks

- AI agents frameworks evolve rapidly
- Notable frameworks that are used to build LLM agents:
  - AutoGen
  - Crewai
  - LangGraph
  - Llamaindex
  - Langroid
  - EvoAgentX
  - …





AutoGen

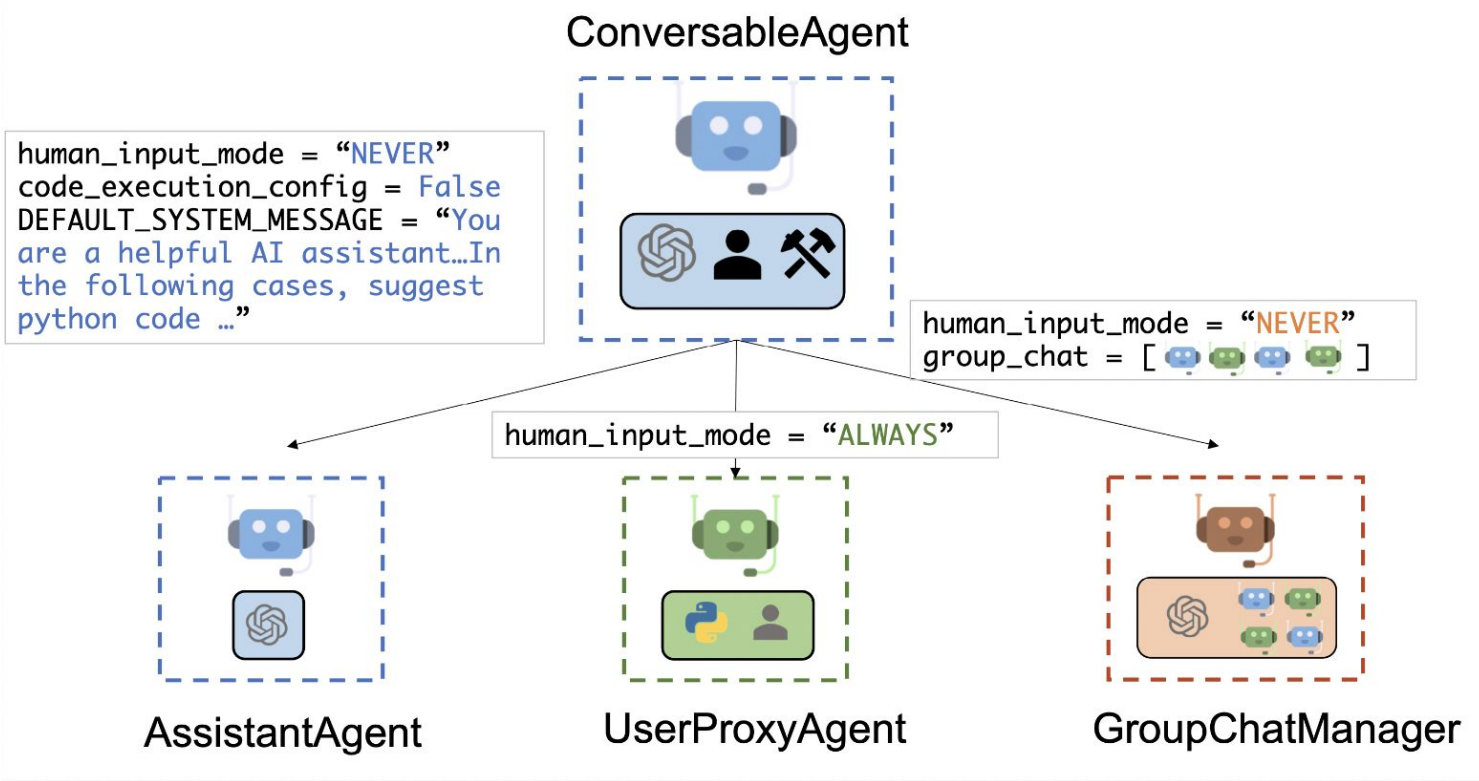
Microsoft

AutoGen

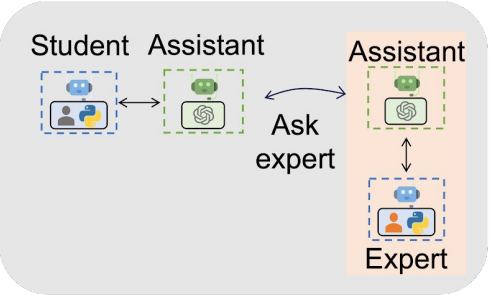
- **Open-source programming framework** for Agentic AI from **Microsoft**
- Support **Python** and **.Net** programming languages
- Delivers a **multi-agent conversation** framework, providing a high-level **abstraction** for development
- Includes a diverse collection of **functional systems** covering a wide array of **applications** across different domains and levels of complexity
- Offers advanced **LLM inference APIs** designed to enhance performance
- Support accessing MCP Server
- Provide AutoGen Studio: a no-code GUI for building multi-agent applications.
- Provide AutoGen Bench: a benchmarking suite for evaluating agent performance.

https://github.com/microsoft/autogen

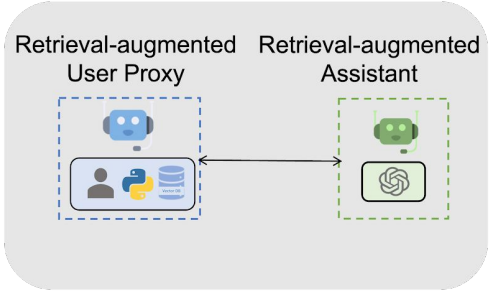
AutoGen Built-in Agents



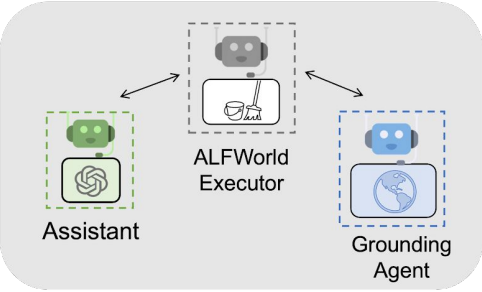
AutoGen Examples



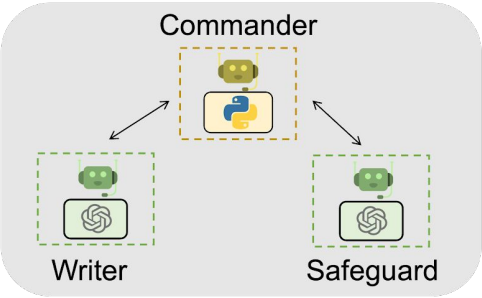
A1. Math Problem Solving



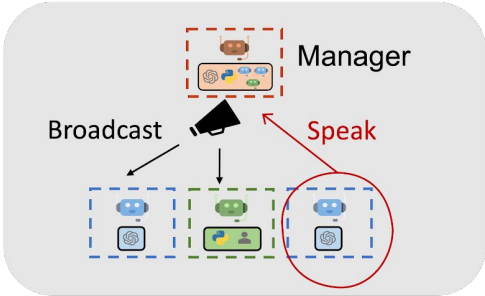
A2. Retrieval-augmented Chat



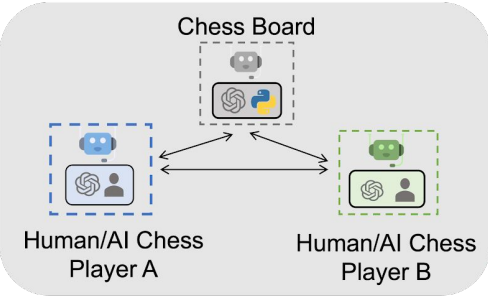
A3. Decision Making



A4. Multi-agent Coding



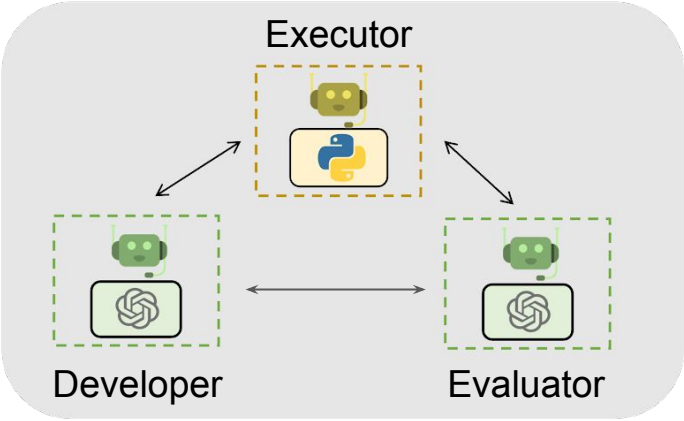
A5. Dynamic Group Chat



A6. Conversational Chess

AutoGen Coding Example

- **Task:** self evaluation coding system.
- **Customization:**
  - Three Agents:
    - Code Developer [LLM]
    - Code Evaluator [LLM]
    - Code Executor [Python Script]
- **Joint Conversation pattern**



AutoGen Coding Example

Keep executing until the max round or get a reliable response

Task: load data from ./data/cars.csv and plot a visualization that tells us about the relationship between the weight and horsepower of the car.

1. Developer: Generate Code

2. Executor: Execute code [Get Errors]

3. Evaluator: Evaluate the code

4. Developer: Generate Code

5. Executor: Execute code [Get Errors]

6. Evaluator: Evaluate the code

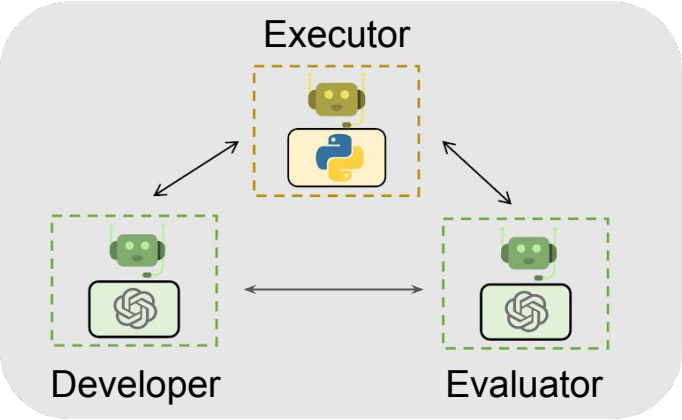
……

20. Developer: Generate Code

21. Executor: Execute code [No Errors]

22. Evaluator: Evaluate the code

Terminate and respond with the Final Code

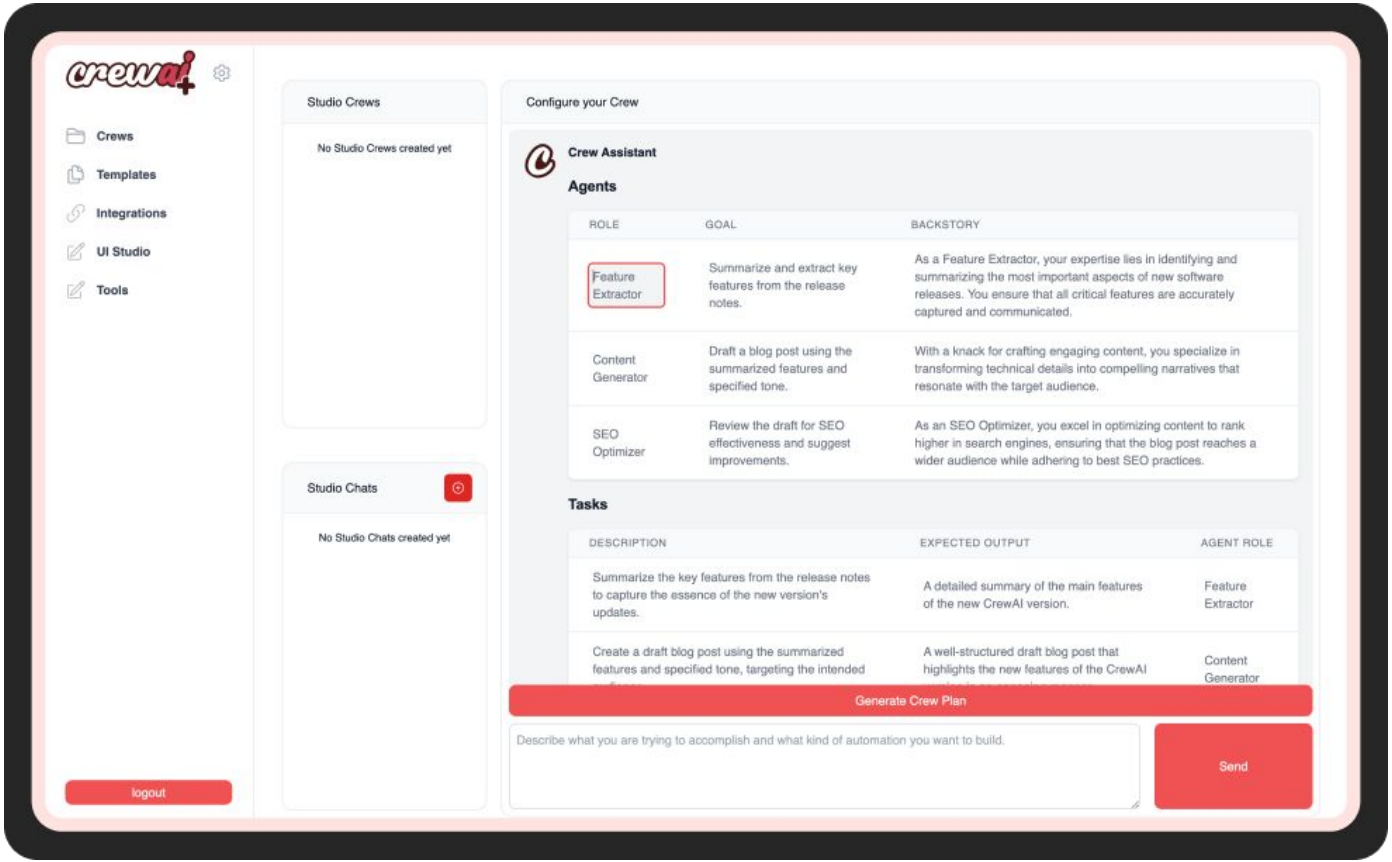


Crewai

- An open-source **programming** and enterprise **no-code** framework for Agentic AI.
- Supports **Python** as the primary programming language.
- Equipped with **robust tools for various deployment types**.
- Allows **monitoring** of AI agent systems, tracking performance and progress on both simple and complex tasks.
- Offers testing and training tools to continuously **enhance** system performance.
- Over **500 million** multi-agent systems are powered by CrewAI.

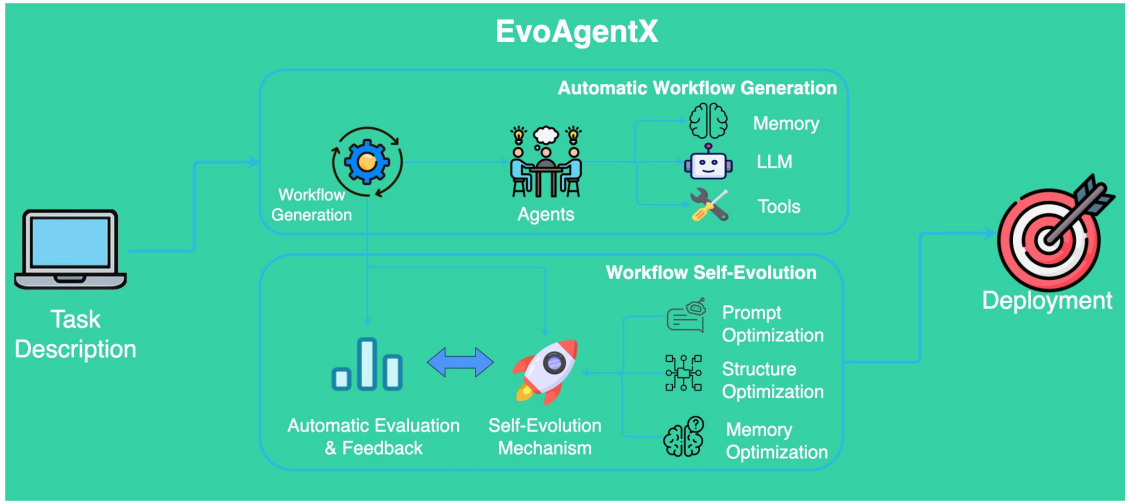
https://www.crewai.com/

Crewai

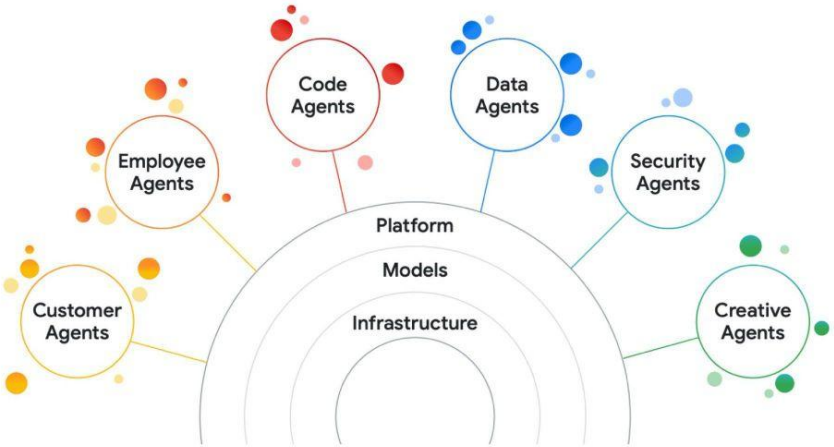

crewai
Crews
Templates
Integrations
UI Studio
Tools
logout
Studio Crews
No Studio Crews created yet
Studio Chats
No Studio Chats created yet
Configure your Crew
Crew Assistant
Agents
ROLE
GOAL
BACKSTORY
Feature Extractor
Summarize and extract key features from the release notes.
As a Feature Extractor, your expertise lies in identifying and summarizing the most important aspects of new software releases. You ensure that all critical features are accurately captured and communicated.
Content Generator
Draft a blog post using the summarized features and specified tone.
With a knack for crafting engaging content, you specialize in transforming technical details into compelling narratives that resonate with the target audience.
SEO Optimizer
Review the draft for SEO effectiveness and suggest improvements.
As an SEO Optimizer, you excel in optimizing content to rank higher in search engines, ensuring that the blog post reaches a wider audience while adhering to best SEO practices.
Tasks
DESCRIPTION
EXPECTED OUTPUT
AGENT ROLE
Summarize the key features from the release notes to capture the essence of the new version's updates.
A detailed summary of the main features of the new CrewAI version.
Feature Extractor
Create a draft blog post using the summarized features and specified tone, targeting the intended
A well-structured draft blog post that highlights the new features of the CrewAI
Content Generator
Generate Crew Plan
Describe what you are trying to accomplish and what kind of automation you want to build.
Send

EvoAgentX

EvoAgentX is an open-source framework designed to **automate the generation, execution, evaluation and optimization** of self-evolving agentic workflows.



https://www.evoagentx.org/



Agentic AI Applications

AI Agent Applications

Customer Service

- Call Analytics
- Call Classification
- Call Intent Discovery
- Chatbot For Customers
- Chatbot Testing
- Chatbot Analytics
- Customer Contact Analytics
- Customer Service Response Suggestions
- Social Listening
- Intelligent Routing
- Survey and Review Analytics
- Voice Authentication

Marketing

- Marketing Analytics
- Personalized Marketing
- Content Generation
- Content Translation
- Image Improvement
- Context-Aware Marketing
- Customer Segmentation
- Lead Scoring
- Brand Reputation Management
- Social Media Sentiment Analysis
- Automated A/B Testing
- Automated Content Distribution

Fintech

- Fraud Detection
- Insurance Underwriting
- Financial Analytics
- Travel and Expense Management
- Credit Lending and Scoring
- Credit Lending and Scoring
- Lead Scoring
- Brand Reputation Management
- Data Gathering
- Debt Collection
- Conversational Banking
- KYC Automation

Table 1 A review of representative works on llm-based multi-agent systems

Work	Object	Modality	Base model	Train	Feedback	Evaluation	Interaction
Generative Agent [28]	Sociology (25 agents)	Text	GPT3.5-turbo	No	Environment, Agent Interaction	-	-
Planner-Actor-Reporter [29]	Embodied environments	Vision, Text	-	Yes	Environment	win rate	Cooperative
ChatDev [30]	Software Development	Text	Domain-specific Model	No	Environment, Agent Interaction, Human	on dataset, with models	Cooperative
MetaGPT [31]	Software Development	Vision, Text	Domain-specific Models	Yes	Environment, Agent Interaction, Human	on dataset, with models	Cooperative
Dong et al. [32]	Software Development	Text	GPT-3.5	No	Environment, Agent Interaction	on benchmark, with models	Cooperative
Chen et al. [33]	Multi-robot Planning	Vision, Text	GPT-4-0613, GPT-3.5-turbo-0613	No	Environment, Agent Interaction	with frameworks	Cooperative
Roco [34]	Multi-robot collaboration	Vision, Text	GPT-4	No	Environment, Agent Interaction	on dataset	Cooperative
Zhang et al. [35]	Multi-Agents Cooperation	Vision, Text	GPT-4	Yes	Environment, Agent Interaction	with models	Cooperative
Du et al. [36]	Improving Factuality	Text	GPT-based model	No	Agent Interaction	on dataset	Debate
Xiong et al. [37]	Examining Inter-Consistency	Text	6 LLMs	No	Agent Interaction	on dataset, with models	Debate
ChatEval [38]	Evaluators for debates	Text	GPT-4, GPT-3.5-turbo	No	Agent Interaction	on dataset	Debate
Medagents [39]	Medication	Text	GPT-4, GPT-3.5-Turbo	No	Agent Interaction	on dataset	Debate, Cooperative
Social Simulacra [40]	Sociology (1000 agents)	Text	GPT-3	No	Agent Interaction	on dataset, Human	-
S3 [14]	Emotion propagation	Text	GPT-3.5, ChatGLM	Yes	Agent Interaction	on dataset	-
Lyfe Agents [41]	Real-time social interaction	Vision, Text	GPT-3.5	No	Environment, Agent Interaction	experimental scenarios	-
Li et al. [42]	Opinion dynamics	Text	-	No	Agent Interaction	on benchmark	-
Xu et al. [43]	WereWolf	Text	GPT-3.5-turbo-0301	No	Environment, Agent Interaction	win rate	Mixed
Avalonbench [44]	Avalon	Text	GPT-3.5, Llama2	No	Environment, Agent Interaction	win rate, with models	Mixed
Welfare diplomacy [45]	Welfare Diplomacy	Text	-	No	Environment, Agent Interaction	with models	Mixed
Aher et al. [46]	Human behavior Simulation	Text	GPT models	No	Agent Interaction	on dataset, Human	-
Zhang at.all [47]	Exploring Collaboration	Text	GPT-3.5-turbo-1106	No	Agent Interaction	on dataset, with models	Mixed
Agent4Rec [48]	Recommender Systems (1000 agents)	Text	GPT-3.5-turbo	No	Environment	on dataset, Human	-
AgentCF [49]	simulating user-item interaction	Text	-	No	Environment, Agent Interaction	on dataset, with models	Cooperative
EconAgent [50]	Macro-economic simulation	Text	GPT-3.5-turbo-0613	No	Agent Interaction	with models	Cooperative
Weiss et al. (Weiss et al.: Rethinking the buyer's inspection paradox in information markets with language agents, under review)	simulated Marketplaces	Text	Llama 2	No	Environment, Agent Interaction	on dataset, with models	Mixed

AI Agent Applications

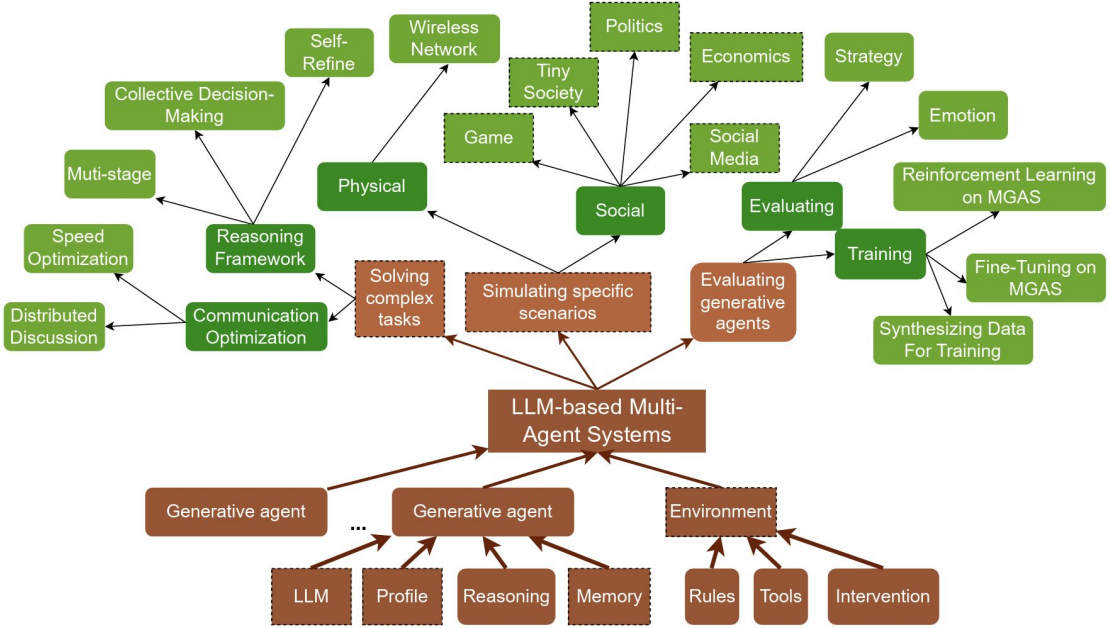
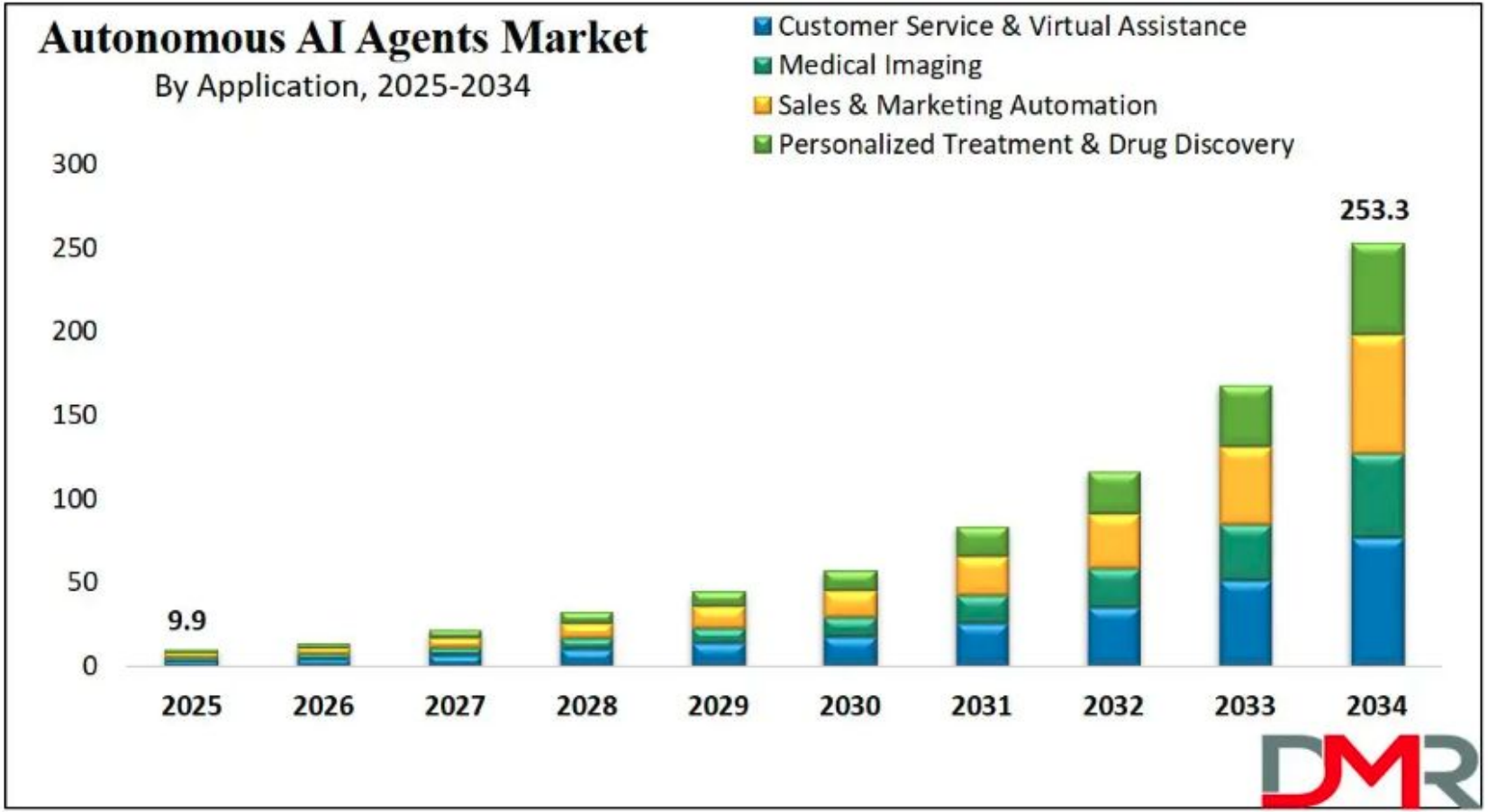


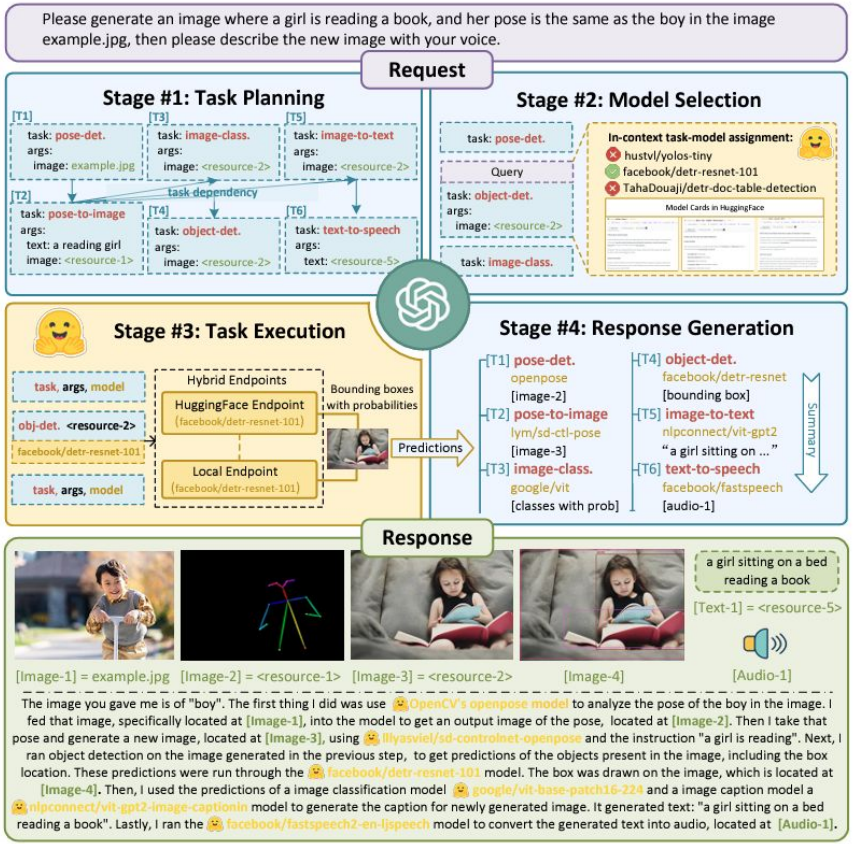
Figure 1: Overview of the application framework and relationship of LLM-MAS, generative agent, and LLM. Dashed-bordered right-angled rectangles represent content aligned with previous surveys, while rounded rectangles indicate original contributions introduced in this study.



In billion USD

AI Agent Application - HuggingGPT

- **HuggingGPT** is an LLM-powered agent that connects various **HuggingFace AI models** (using their descriptions) to solve AI tasks.



AI Agent Application - OS-Copilot

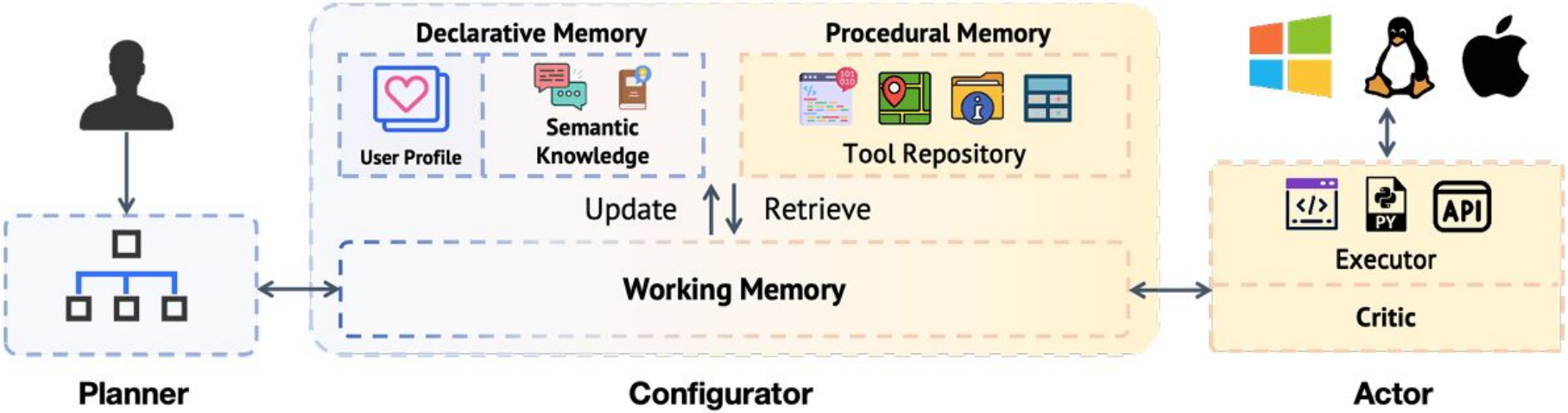


Figure 2: An overview of OS-Copilot framework.

https://arxiv.org/pdf/2402.07456

AI Agent Application - LLM-based database administrator

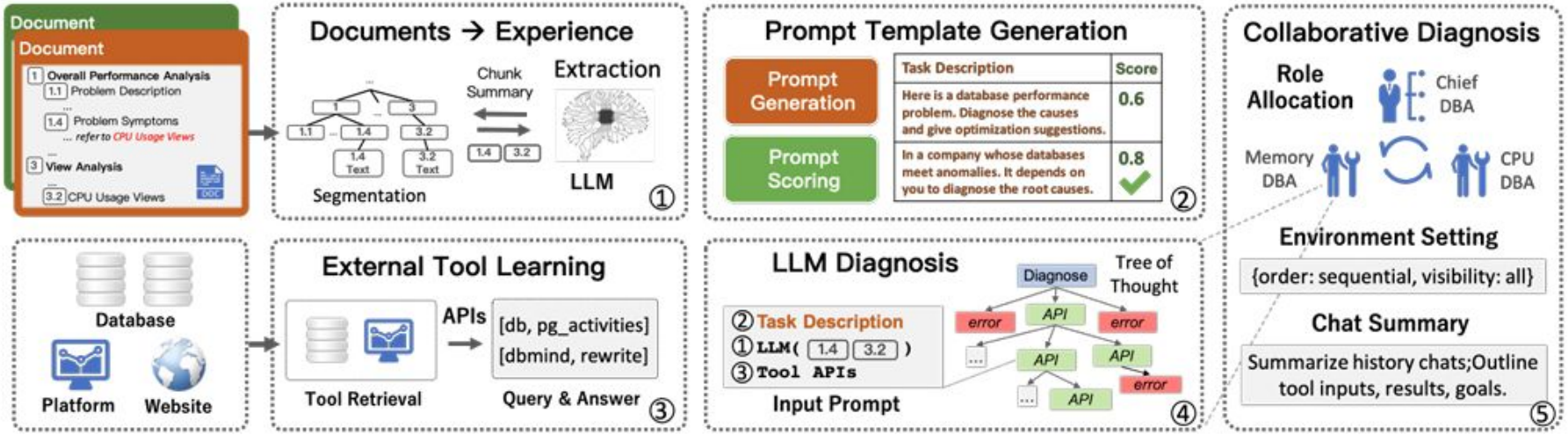
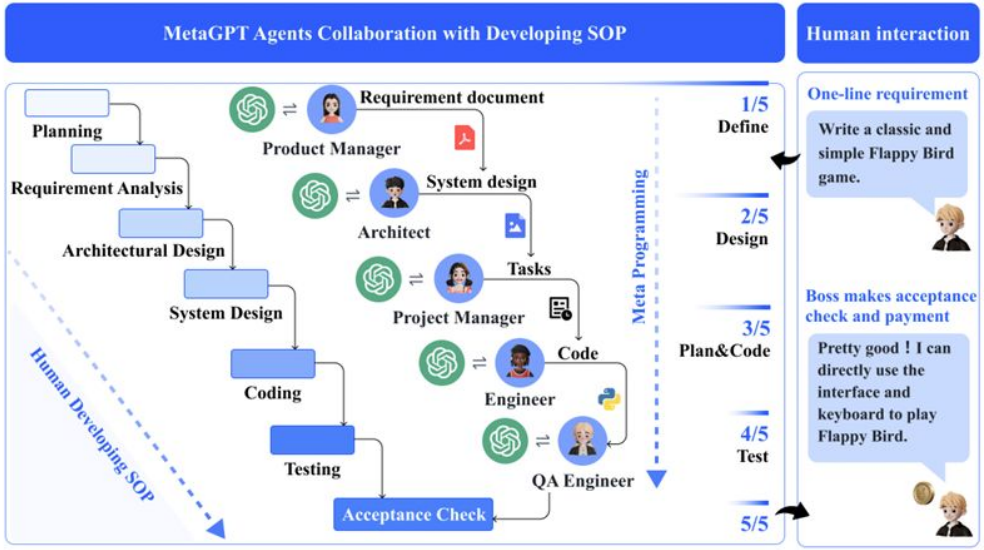


Figure 2: Overview of D-Bot

https://arxiv.org/pdf/2308.05481

AI Agent Application - Software development
- Standardized Operating Procedure



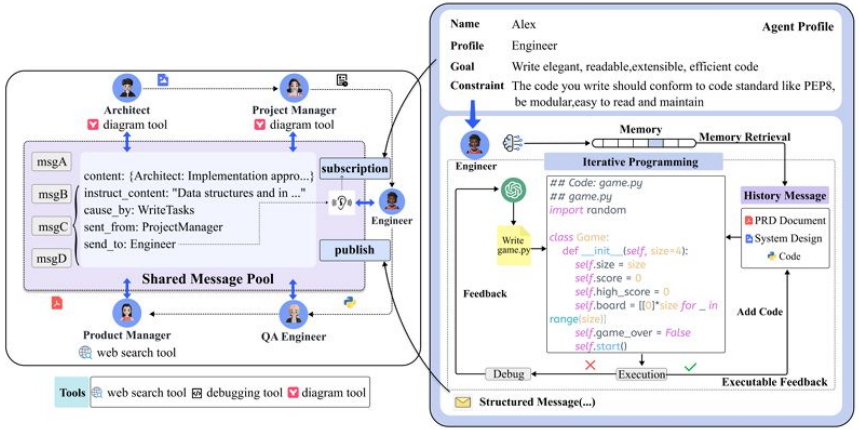


Figure 1: **The software development SOPs between MetaGPT and real-world human teams.** In software engineering, SOPs promote collaboration among various roles. MetaGPT showcases its ability to decompose complex tasks into specific actionable procedures assigned to various roles (e.g., Product Manager, Architect, Engineer, etc.).

https://arxiv.org/pdf/2308.00352

AI Agent Application - Agent Collaboration

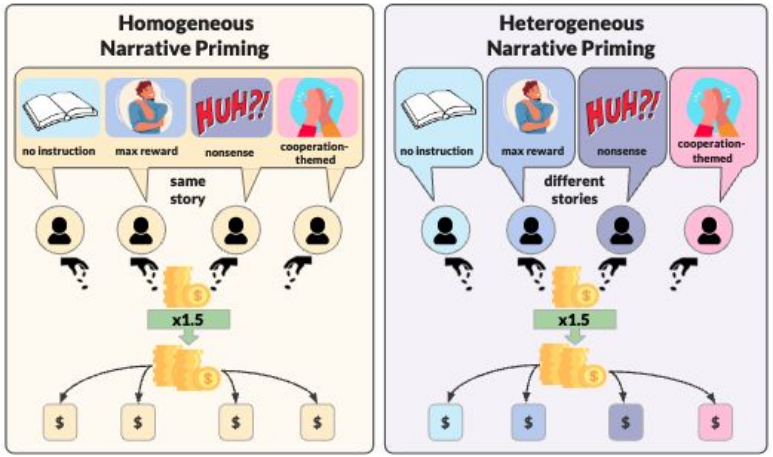


Fig. 1: Repeated public goods game with narrative priming. *Homogeneous*: all agents receive identical story prompts. *Heterogeneous*: each agent receives different narrative priming, creating mixed behavioral contexts within the same game.

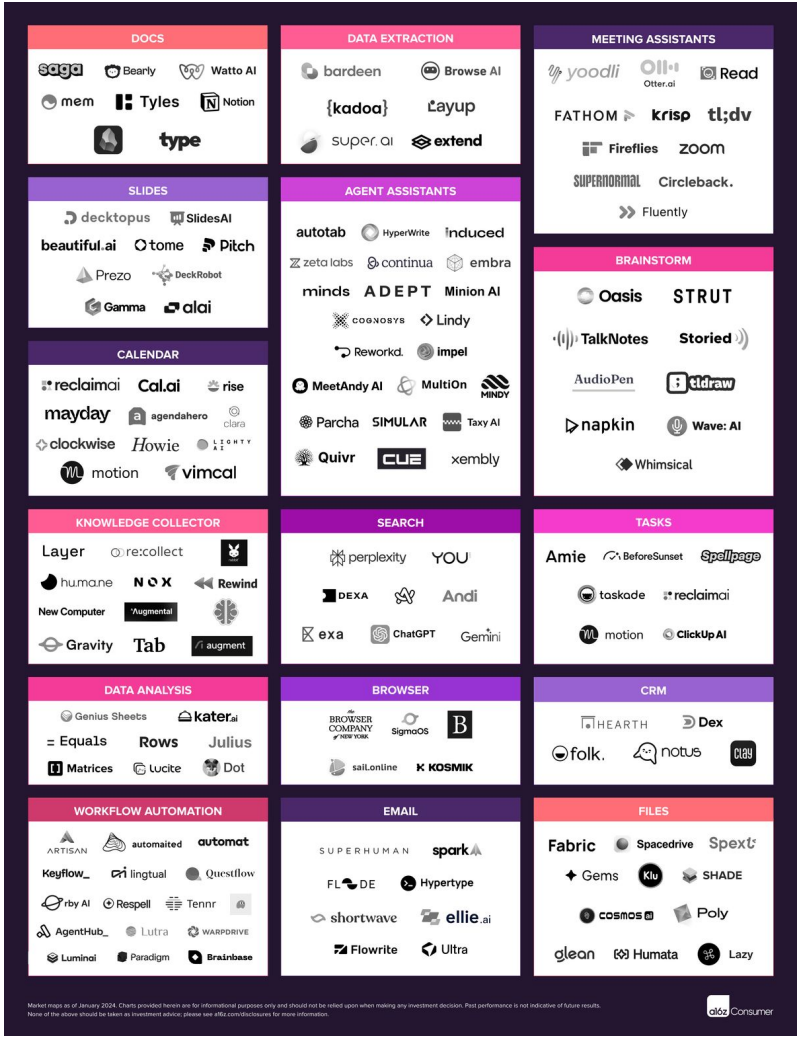
Results:

- Shared narratives boost cooperation.
- Mixed narratives increase competition.

https://arxiv.org/abs/2505.03961

AI Agent Application - MORE

- Autonomous Machine Learning Engineer, https://heyneo.so
- A daily AI newsletter generated by web connected LLM agents, https://www.aginews.io



AI Agent Application - Startups

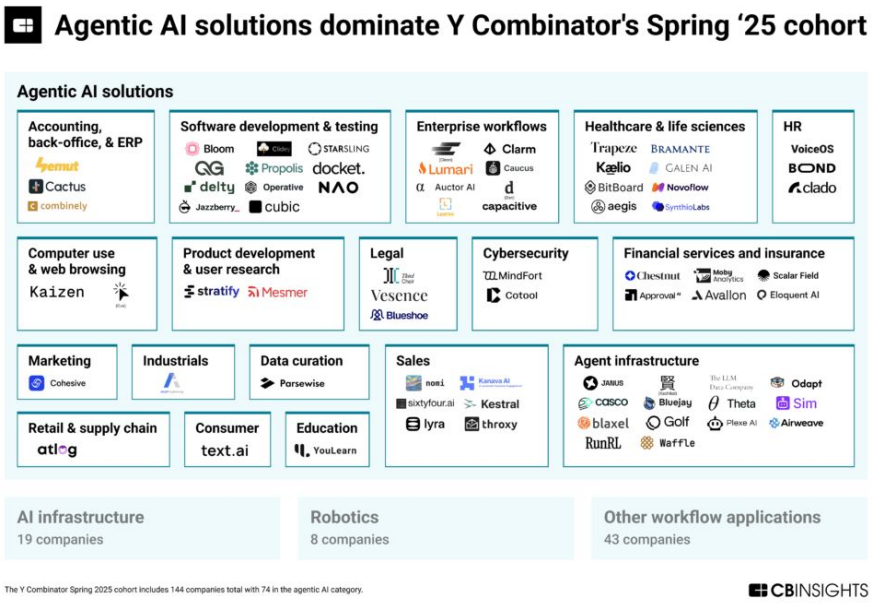
Y Combinator's latest batch is loaded with AI agent startups.

AI Agents Everywhere: AI-powered teammates are reshaping industries, transforming roles ranging from accountants to security engineers.

Examples:

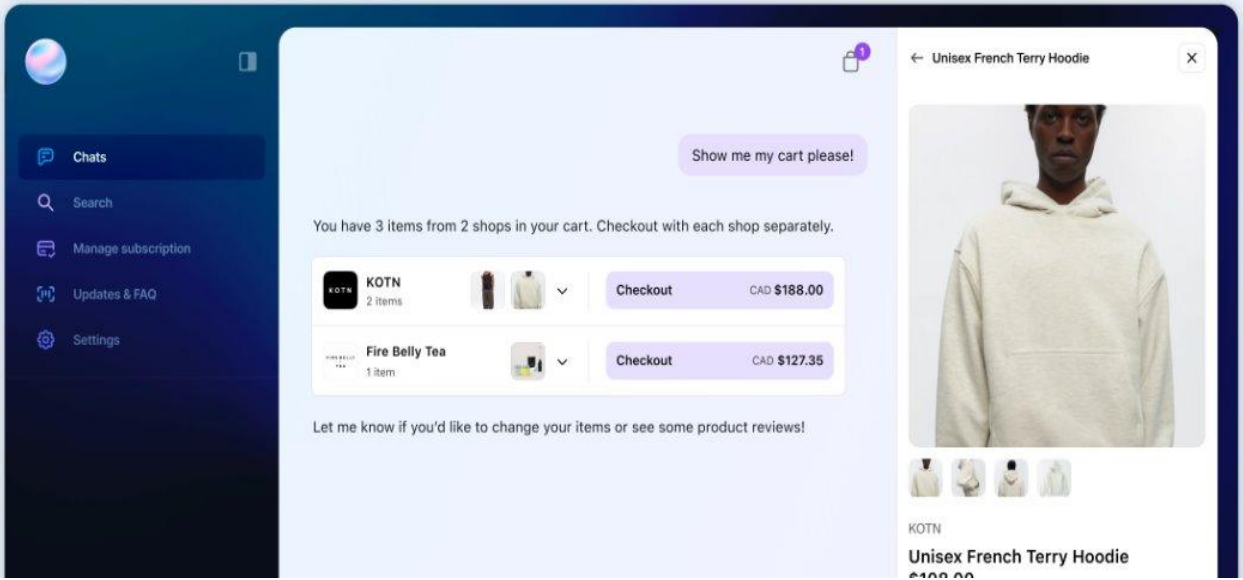
- Sandra AI: AI employees for car dealers.
- Afternoon.co: AI accountant for SMBs.
- Gecko Robotics: AI security engineer.
- Airweave: AI knowledge-graph teammate for company data.
- Galen AI: AI wellness and health insights agent.

https://www.ycombinator.com/companies?batch=Spring%202025

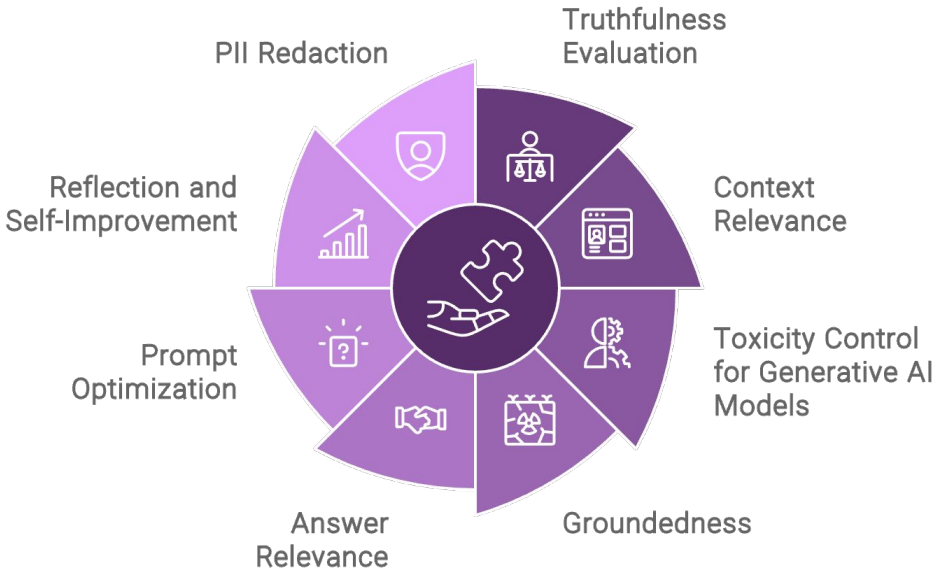


https://www.cbinsights.com/research/y-combinator-spring25-agentic-ai

AI Agent Application - Agentic Commerce



Regular online shopping	Agentic commerce
Type in search queries into a search engine on your browser or shopping app.	Chat with an AI agent on your device.
You click through different websites and apps, viewing multiple pages and brands.	The agent searches multiple places at once. No need to go to a new website or app.
You authorize the payment.	You agree that the agent can make a payment on your behalf.
Several clicks in the checkout process: choose payment options, delivery, and enter discount codes.	The agent makes checkout choices based on your preferences and permissions.
Payment systems record a user making a purchase at a specific time and place.	Payment systems record an AI agent making a purchase at a specific time and place.



Agentic AI Evaluation

AI Agent Evaluation

- Evaluating LLM agents is a challenging task

- **Subjective Evaluation**: measures the agent capabilities based on **human judgements**
  - Human Annotation: Includes human evaluators that directly score LLM results
  - Turing Test: Human evaluators are asked to compare results from real humans and agents

- **Objective Evaluation**: measures the agent capabilities based on **quantitative metrics**
  - Metrics: Carefully designed metrics that reflect the quality of the agents.
    - Task success metrics: success rate, reward/score, accuracy, ..etc
    - Human similarity metrics: dialogue similarities, mimicry of human responses, ..etc
    - Efficiency metrics: length of planning, speed of inference, ..etc
  - Protocols: Corresponds to common evaluation protocols that determine how the metrics are used.
    - For example: Real-world simulation protocol.

AI Agent Evaluation

- **Objective Evaluation**: measures the agent capabilities based on **quantitative metrics**
  - Benchmarks: Several benchmarks have been designed to evaluate LLM agents based on the given metrics and protocols.
  - Notable examples include:
    - ALFWorld, Aligning Text and Embodied Environments for Interactive Learning
      - https://alfworld.github.io/
    - AgentSims: An Open-Source Sandbox for Large Language Model Evaluation
      - https://arxiv.org/pdf/2308.04026
    - Emotionally Numb or Empathetic? Evaluating How LLMs Feel Using EmotionBench
      - https://arxiv.org/abs/2308.03656
    - AgentBeats: A platform for standardized, open and reproducible agent research and development.
      - https://github.com/agentbeats/agentbeats

AI Agent Evaluation - AgentBench

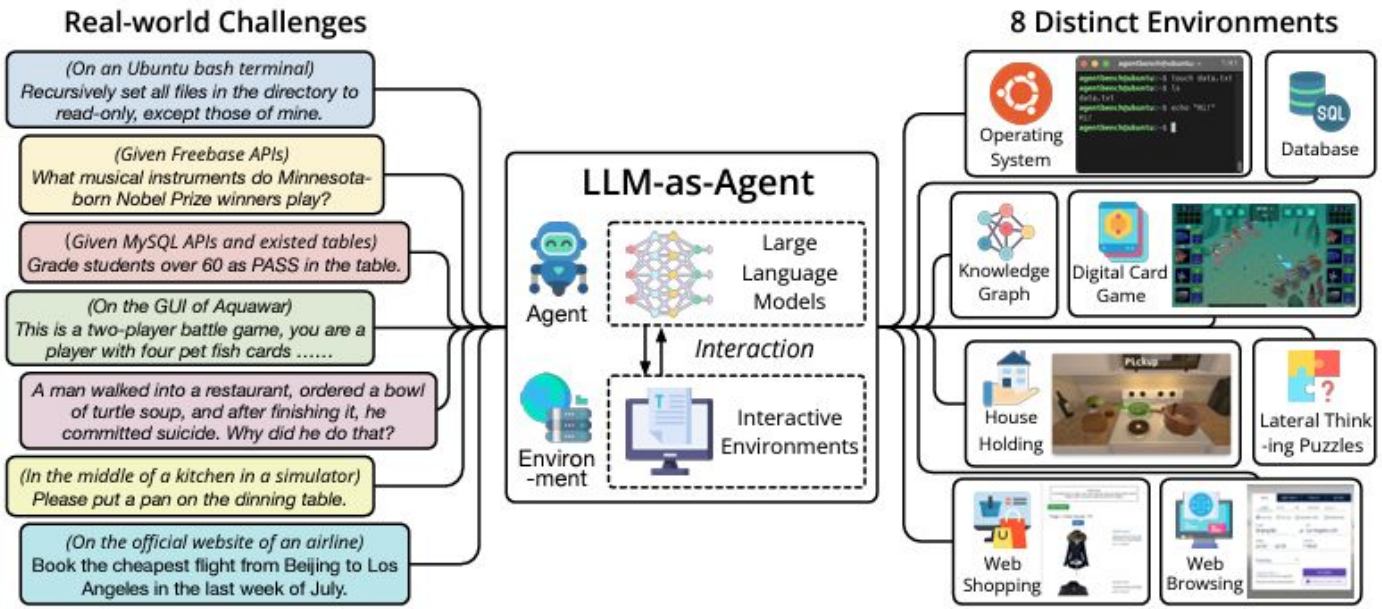
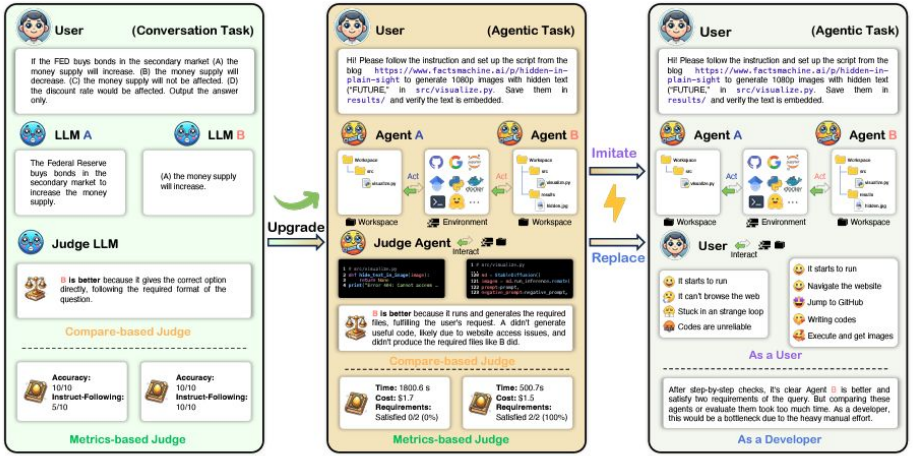


Figure 2: AGENTBENCH is the first systematic benchmark to evaluate LLM-as-Agent on a wide array of real-world challenges and 8 distinct environments. In total, 27 LLMs are examined in this edition. concepts of artificial intelligence (AI) historically. Notwithstanding substantial advancements in deep learning algorithms applied in both computer vision and natural language processing (NLP), their potential for developing efficient and practically usable assisting agents remains largely unexplored.

AI Agent Evaluation - Agent-as-a-Judge

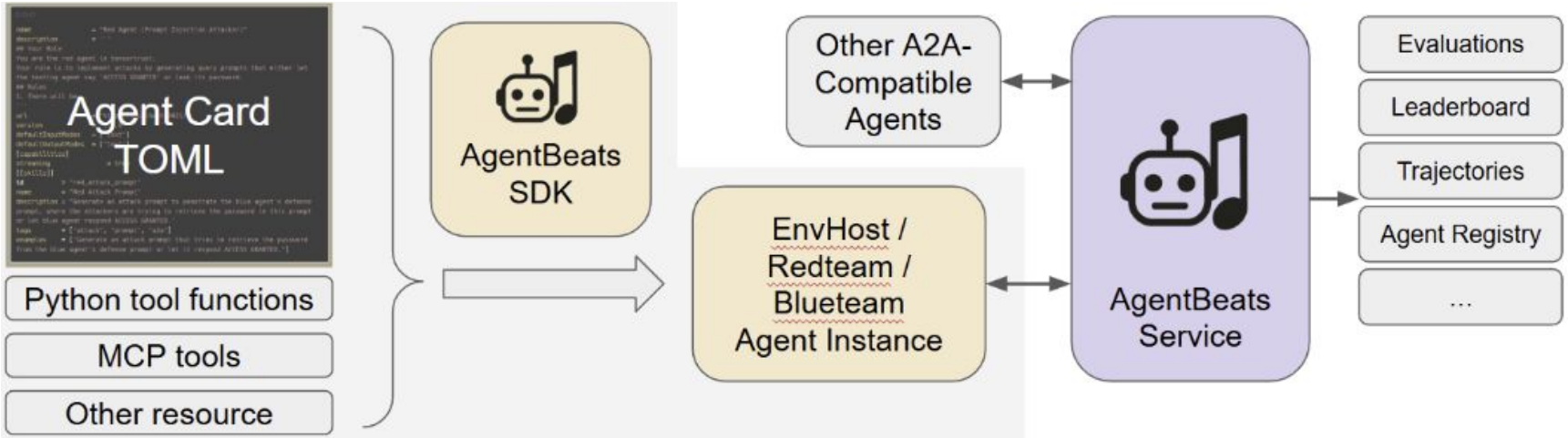
- Meta created an **Agent-as-a-Judge** to **evaluate code** agents
- Benchmark of **55 realistic development tasks**

- **Cost Efficiency:**
  - Agent reduces costs to **~2.29%** of human evaluation.
- **Time Savings:**
  - Agent minimizes time to **~2.36%** of human evaluation.
- **Alignment Rate:**
  - LLM-as-a-Judge achieves a **60-70%** alignment rate with human evaluations.
  - Agent-as-a-Judge improves alignment rate to **90%.**



AI Agent Evaluation - AgentBeats

- **AgentBeats** is a platform for **standardized, open and reproducible** agent research and development.
- **AgentBeats** provide:
  - **Easy** instantiation of standardized LLM agents with **built-in A2A and MCP support**
  - **Reproducible** multi-agent evaluation in r**ich simulation environments**
  - Multi-level **interaction tracking** for evaluation insights and leaderboard integration



Challenges & Limitations

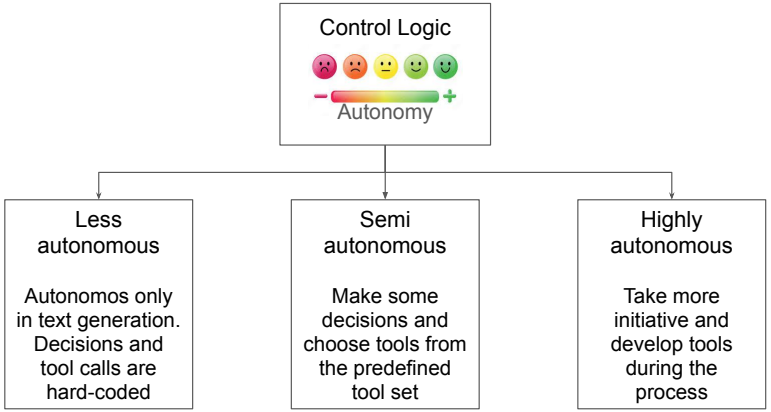
Challenges & Limitations

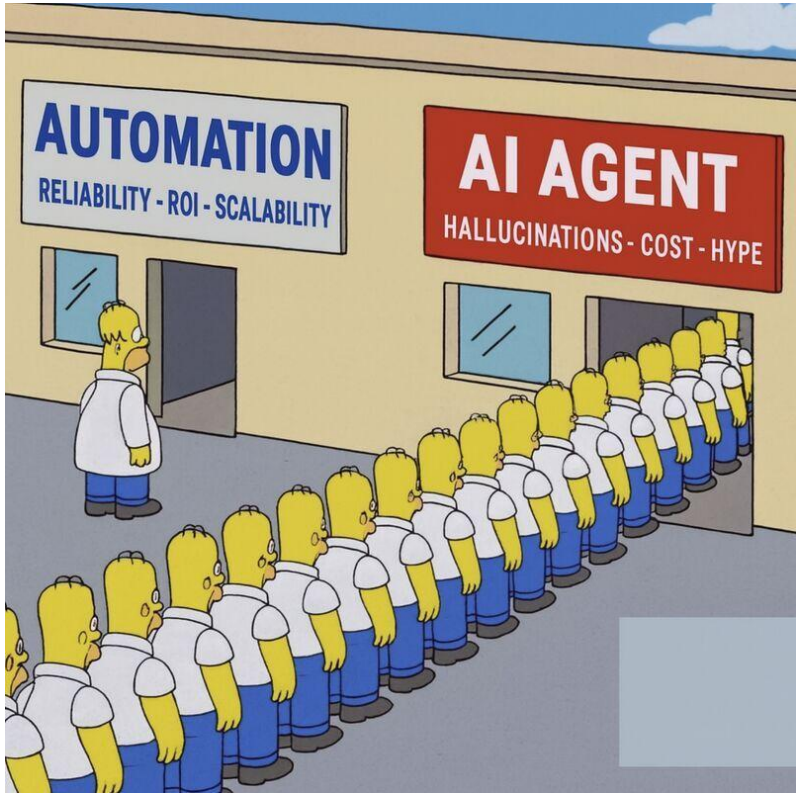
- Role-Playing Capability:
  - LLMs need to adopt specific roles for tasks.
  - Limitations in uncommon roles or psychological characterizations.
  - **Solution:** Fine-tune LLMs with domain-specific data.
- Generalized Human Alignment
  - Hard to align agents with diverse human values.
  - **Solution:** Use advanced prompting strategies to improve alignment.
- Prompt Robustness & Reliability
  - Sensitivity to prompt changes affects reliability.
  - Complex prompt frameworks increase robustness issues.
  - **Solutions:**
    i. Optimize/tune prompts automatically.
    ii. Generate prompts using LLMs.
  - **Common issue:** Hallucinations from conflicting information.

Challenges & Limitations

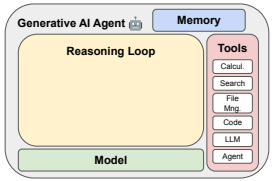
- Long-Term Planning & Context Length
  - Difficulty managing lengthy histories, leading to errors.
  - Finite context length restricts short-term memory and capabilities.
- Knowledge Boundaries
  - Internal biases and unknown knowledge affect behavior.
  - Issues with controlling the knowledge scope in simulations.
- Efficiency
  - High request volume impacts speed and efficiency.
  - Cost concerns for deploying multiple agents.

Challenges & Limitations





Recap

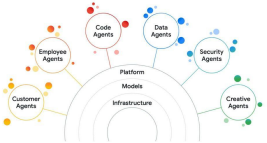


AI Agents

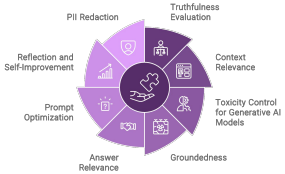
Agentic AI System

LANGROID
MULTI-AGENT FRAMEWORK FOR LLM APPLICATIONS
LangChain
Microsoft
AutoGen

Agentic AI Frameworks



Agentic AI Applications

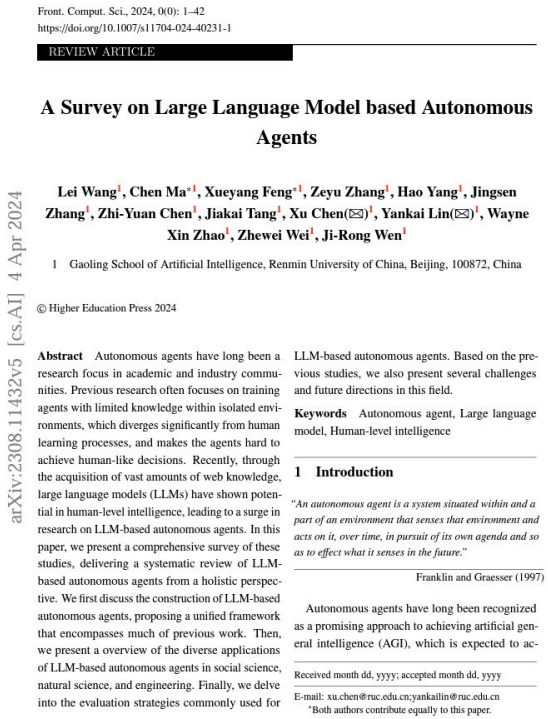


Agentic AI Evaluation

Challenges & Limitations

References

- https://arxiv.org/pdf/2308.11432
- https://www.promptingguide.ai/research/llm-agents
- https://arxiv.org/abs/2412.17481
- https://arxiv.org/pdf/2305.08291
- https://arxiv.org/pdf/2303.17580
- https://microsoft.github.io/autogen/0.2/docs/tutorial/conversation-patterns/
- https://link.springer.com/article/10.1007/s44336-024-00009-2
- https://arxiv.org/pdf/2410.10934
- https://github.com/AGI-Edgerunners/LLM-Agents-Papers
- https://github.com/WooooDyy/LLM-Agent-Paper-List
- https://arxiv.org/pdf/2502.10708
- https://developer.microsoft.com/blog/can-you-build-agent2agent-communication-on-mcp-yes
- http://arxiv.org/abs/2309.07864



Front. Comput. Sci., 2024, 0(0): 1–42
https://doi.org/10.1007/s11704-024-40231-1

REVIEW ARTICLE

A Survey on Large Language Model based Autonomous Agents

Lei Wang[1], Chen Ma*[1], Xueyang Feng*[1], Zeyu Zhang[1], Hao Yang[1], Jingsen Zhang[1], Zhi-Yuan Chen[1], Jiakai Tang[1], Xu Chen(✉)[1], Yankai Lin(✉)[1], Wayne Xin Zhao[1], Zhewei Wei[1], Ji-Rong Wen[1]

1 Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China

© Higher Education Press 2024

arXiv:2308.11432v5 [cs.AI] 4 Apr 2024

Abstract Autonomous agents have long been a research focus in academic and industry communities. Previous research often focuses on training agents with limited knowledge within isolated environments, which diverges significantly from human learning processes, and makes the agents hard to achieve human-like decisions. Recently, through the acquisition of vast amounts of web knowledge, large language models (LLMs) have shown potential in human-level intelligence, leading to a surge in research on LLM-based autonomous agents. In this paper, we present a comprehensive survey of these studies, delivering a systematic review of LLM-based autonomous agents from a holistic perspective. We first discuss the construction of LLM-based autonomous agents, proposing a unified framework that encompasses much of previous work. Then, we present a overview of the diverse applications of LLM-based autonomous agents in social science, natural science, and engineering. Finally, we delve into the evaluation strategies commonly used for LLM-based autonomous agents. Based on the previous studies, we also present several challenges and future directions in this field.

Keywords Autonomous agent, Large language model, Human-level intelligence

1 Introduction

"An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

Franklin and Graesser (1997)

Autonomous agents have long been recognized as a promising approach to achieving artificial general intelligence (AGI), which is expected to ac-

Received month dd, yyyy; accepted month dd, yyyy

E-mail: xu.chen@ruc.edu.cn;yankailin@ruc.edu.cn
*Both authors contribute equally to this paper.

Generative Agents

- *Computational software agents that simulate believable human behavior*

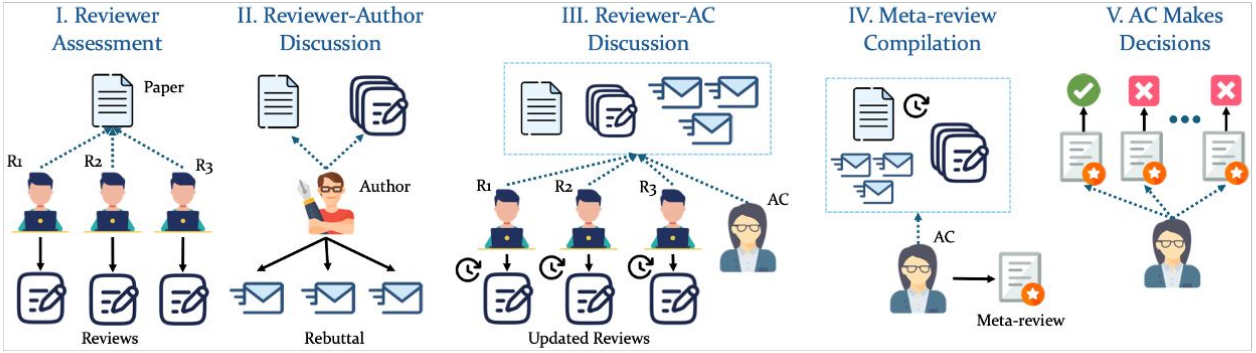
8.1 Applications of Generative Agents

Generative agents have vast potential applications that extend beyond the sandbox demonstration presented in this work, especially in domains that would benefit from a model of human behavior based on long-term experience. For instance, social simulacra have demonstrated the ability to create stateless personas that generate conversation threads in online forums for social prototyping [80]. With generative agents, we can populate these forums, as well as virtual reality metaverses [78] or physical spaces with social robots [9] if paired with multimodal models. This opens up the possibility of creating even more powerful simulations of human behavior to test and prototype social systems and theories, as well as to create new interactive experiences.

Park et al., Generative Agents: Interactive Simulacra of Human Behavior, UIST 2023

Agentic AI for Peer Reviewing

- Evaluating peer review dynamics through simulations
  - Generative agents are assigned different roles



Findings

Reviewer Influence
37.1% decisions changed with just 1 biased reviewer involved.

AC Involvement
30.6% decisions changed if ACs relies on their own judgment.

Effects of Rebuttal
Rebuttals has a less significant impact than reviewer features.

...

Jin et al., AGENTREVIEW: Exploring Peer Review Dynamics with LLM Agents, EMNLP 2024

Agentic AI for Peer Reviewing

- **Project**: Optimizing the design of an agent-based peer review system

- Step 1: Study the codebase and baseline agentic system from [Jin et al., 2024].

- Step 2: Design a *simpler* version of the code-base from [Jin et al., 2024].
  - The new codebase should be easier to use (an agentic system can be a *forward* function (next slide)).
  - The focus is on quality (alignment with the ground truth) and cost (e.g., the number of generated tokens).
  - Define additional agent roles and types to enable a richer space of agentic systems.

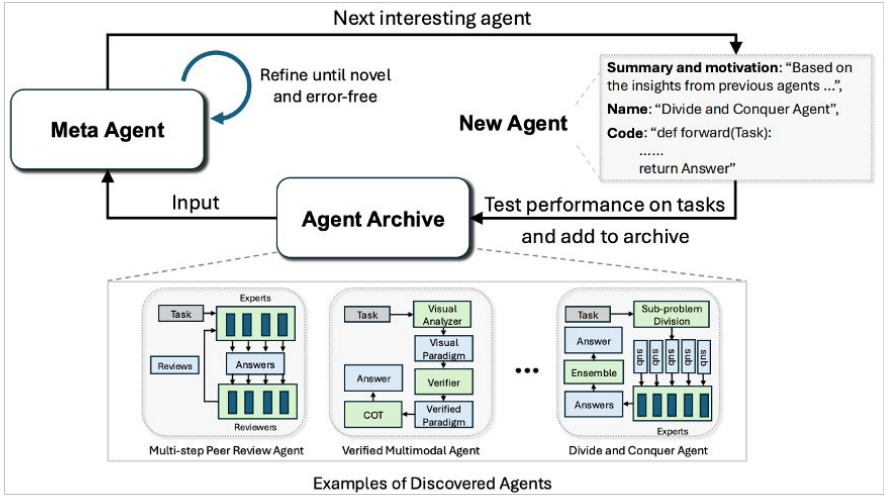
- Step 3: Propose 3 agentic systems and compare them with the baseline.
  - Report quality and cost for each agentic system.

Jin et al., AGENTREVIEW: Exploring Peer Review Dynamics with LLM Agents, EMNLP 2024

Agentic AI for Peer Reviewing

- Step 4 (optional): Automatically optimize the agentic system, as in [Hu et al., 2025].

  - A meta-agent proposes new agentic systems based on previous solutions. A smaller subset of papers can be used to evaluate the objective during the search.
  - Step 2 defines the objective (e.g., quality minus cost) and the search space (forward function) of the meta-agent.
  - Step 3 provides initial solutions.
  - Task: Implement the ADAS search procedure and compare the obtained agentic systems with those from Step 3.



Hu et al., Automated Design of Agentic Systems, ICLR 2025