AI Agents and Agentic Frameworks: An Overview

Frank Brockners with Reinaldo Penno
Distinguished Engineers, Outshift by Cisco

AIHUB-2170

Cisco Webex App

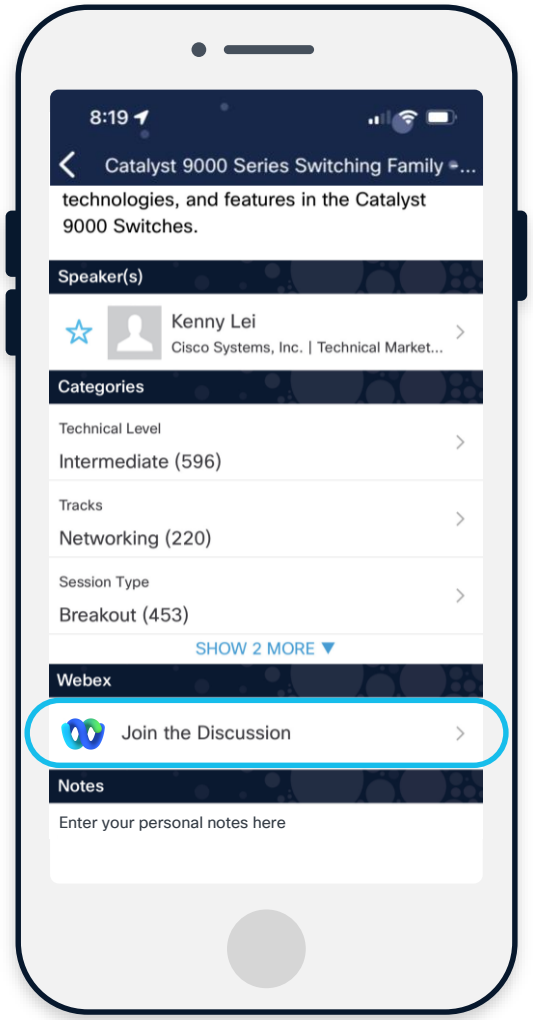
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 13, 2025.



https://ciscolive.ciscoevents.com/
ciscolivebot/**#AIHUB-2170**

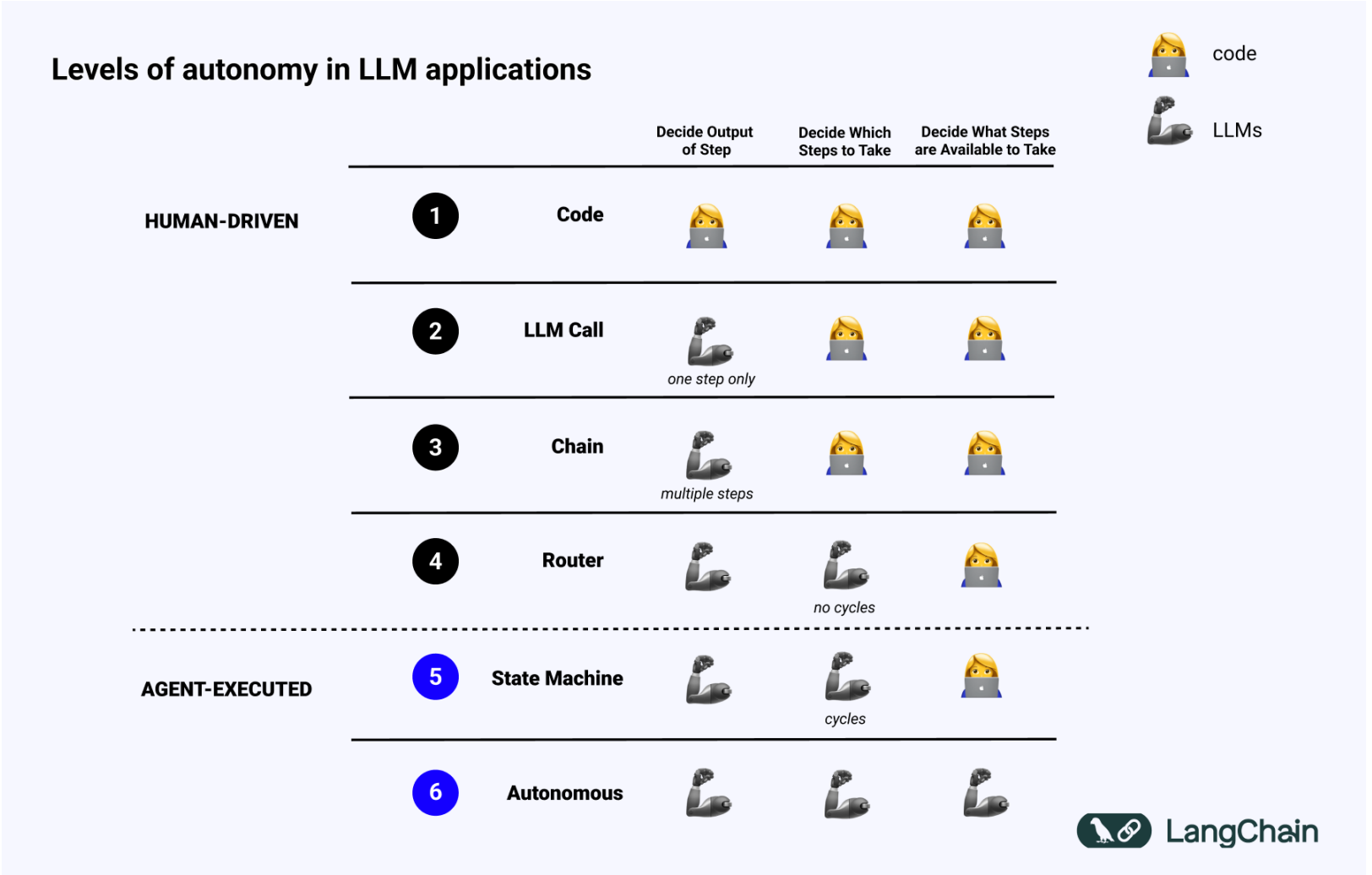
What is “Agentic AI”?

First, what even is an agent?

At LangChain, we define an agent as a system that uses an LLM to decide the control flow of an application. Just like the levels of autonomy for autonomous vehicles, there is also a spectrum of agentic capabilities.

https://www.langchain.com/stateofaiagents
https://blog.langchain.dev/what-is-an-agent/

Levels of autonomy in LLM applications

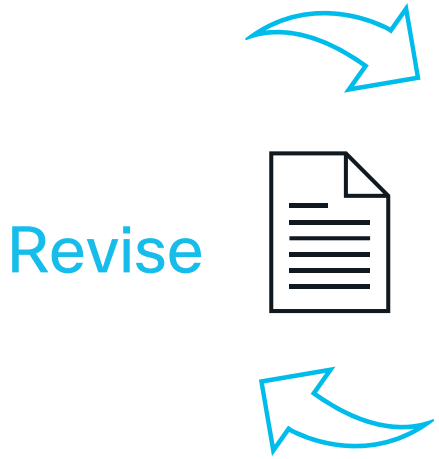


https://blog.langchain.dev/what-is-an-agent/

What is Agentic?

> More and more people are building systems that prompt a large language model multiple times using agent-like design patterns. But there's a gray zone between what clearly is not an agent (prompting a model once) and what clearly is (say, an autonomous agent that, given high-level instructions, plans, uses tools, and carries out multiple, iterative steps of processing).
>
> Rather than arguing over which work to include or exclude as being a true agent, we can acknowledge that there are different degrees to which systems can be agentic. Then we can more easily include everyone who wants to work on agentic systems. We can also encourage newcomers to start by building simple agentic workflows and iteratively make their systems more sophisticated.

Andrew Ng,
https://x.com/AndrewYNg/status/1801295202788983136



Zero Shot – Non-Agentic Workflow

“Please write an essay on topic X from start to finish in one go, without using backspace”

Agentic Workflow

“Write an essay outline on topic X”

“Do web research on the items of the outline”

“Write a first draft”

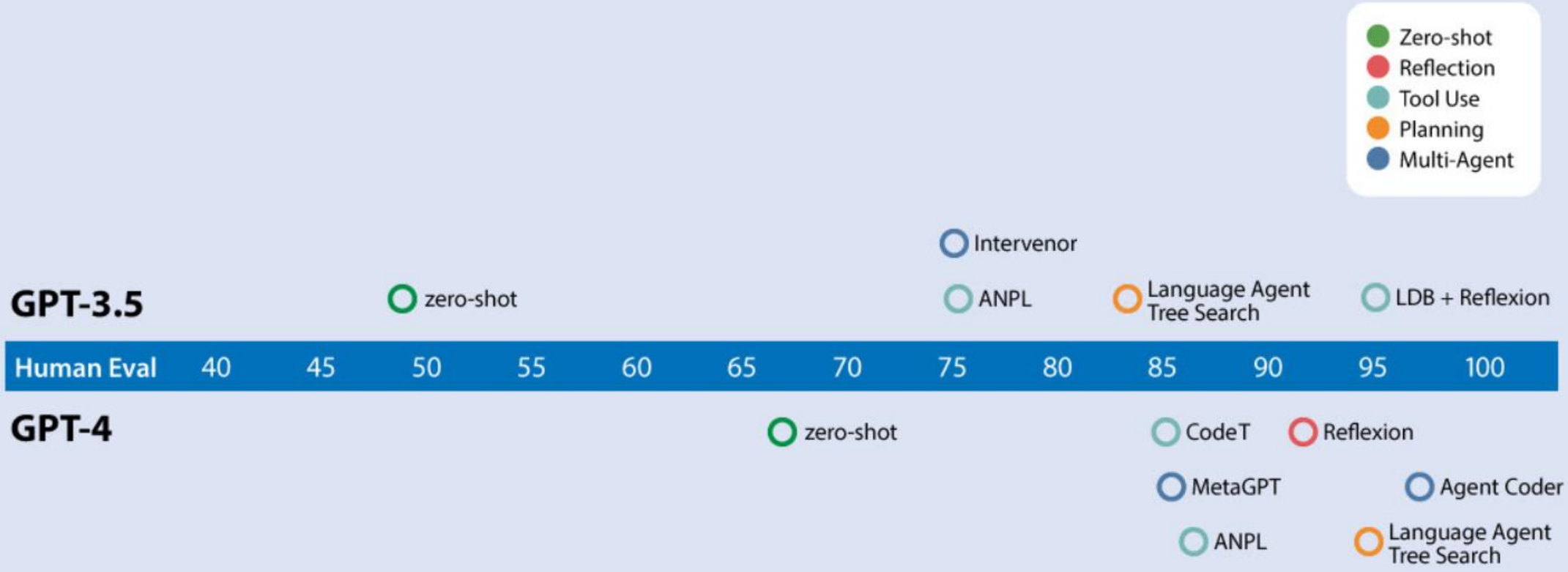
“Consider what parts need revision and more research”

“Revise your draft”

….

Adapted from Andrew Ng: https://youtu.be/q1XFm21I-VQ?si=zdfDMZ3mFobXpTpi

GPT-3.5 and GPT-4 performance using zero-shot and agent workflows



Performance of GPT-3.5 and GPT-4 (zero-shot) on HumanEval, along with algorithms that use agent workflows on top of GPT-3.5 or GPT-4. Thanks to Joaquin Dominguez and John Santerre for help with this analysis.

Source: https://www.deeplearning.ai/the-batch/how-agents-can-improve-llm-performance/

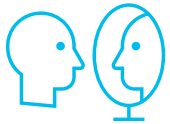
Design Patterns of Agentic Systems

Planning

Think through the steps that need to be taken upfront

Tool Calling

Know which tools are available and how to use them



Reflection

Iteratively improve results through critique, suggestions, and reasoning

Collaboration

Multiple agents collaborate and communicate

Memory

Track progress/ results and learn individually/ collectively

Agentic AI is hot and is getting hotter

“In the first three quarters of this year, GenAI startups secured over $20 billion, according to S&P Global Market Intelligence data. That puts 2024 on track to exceed the 2023 total of $22.7 billion.”

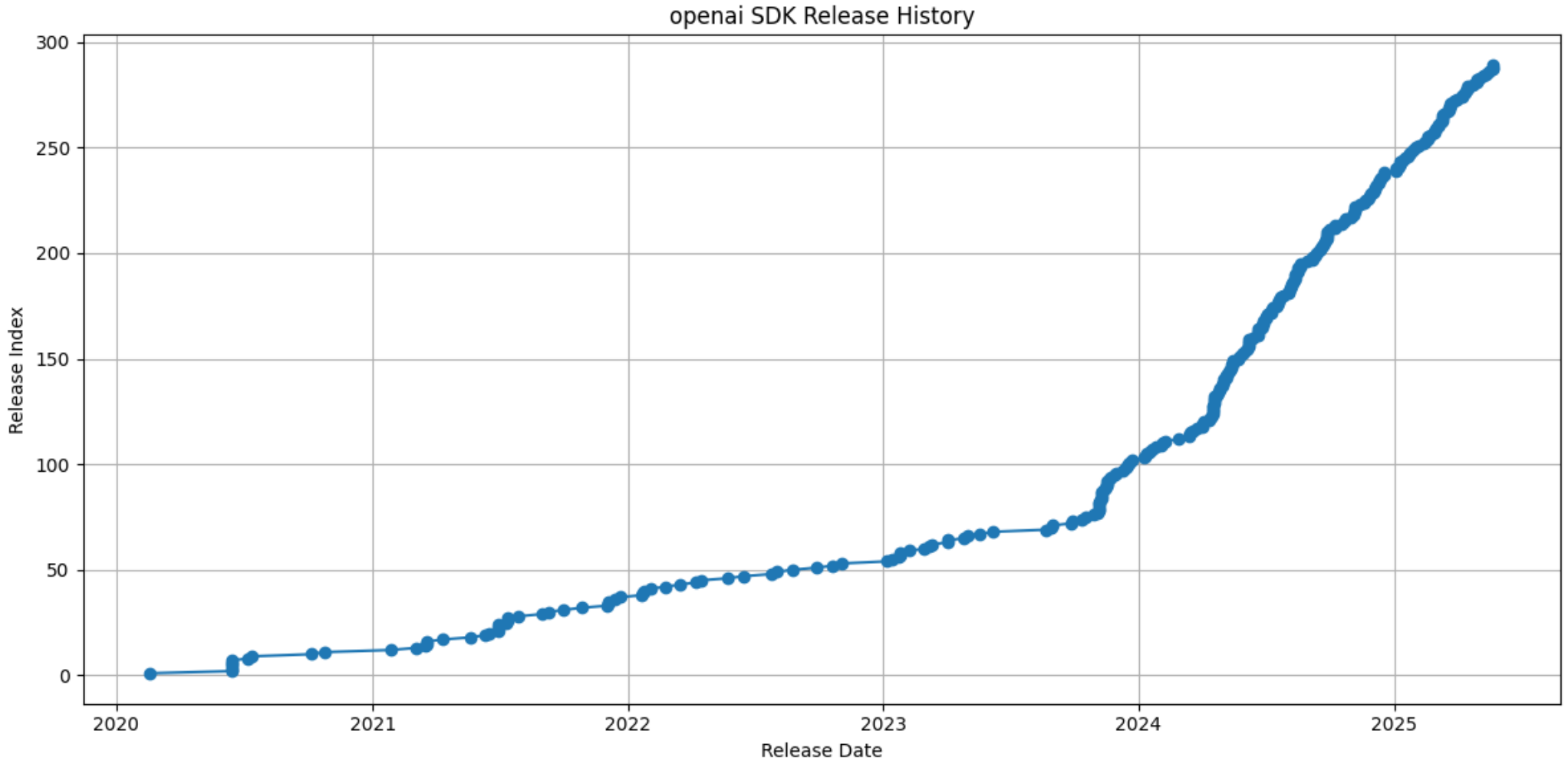
“In the next four years, Gartner predicts that at least 15% of people will make daily work decisions autonomously through agentic AI.”

https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/genai-funding-on-track-to-set-new-record-in-2024-85779779

https://www.linkedin.com/feed/update/urn:li:activity:7267653215356149760/

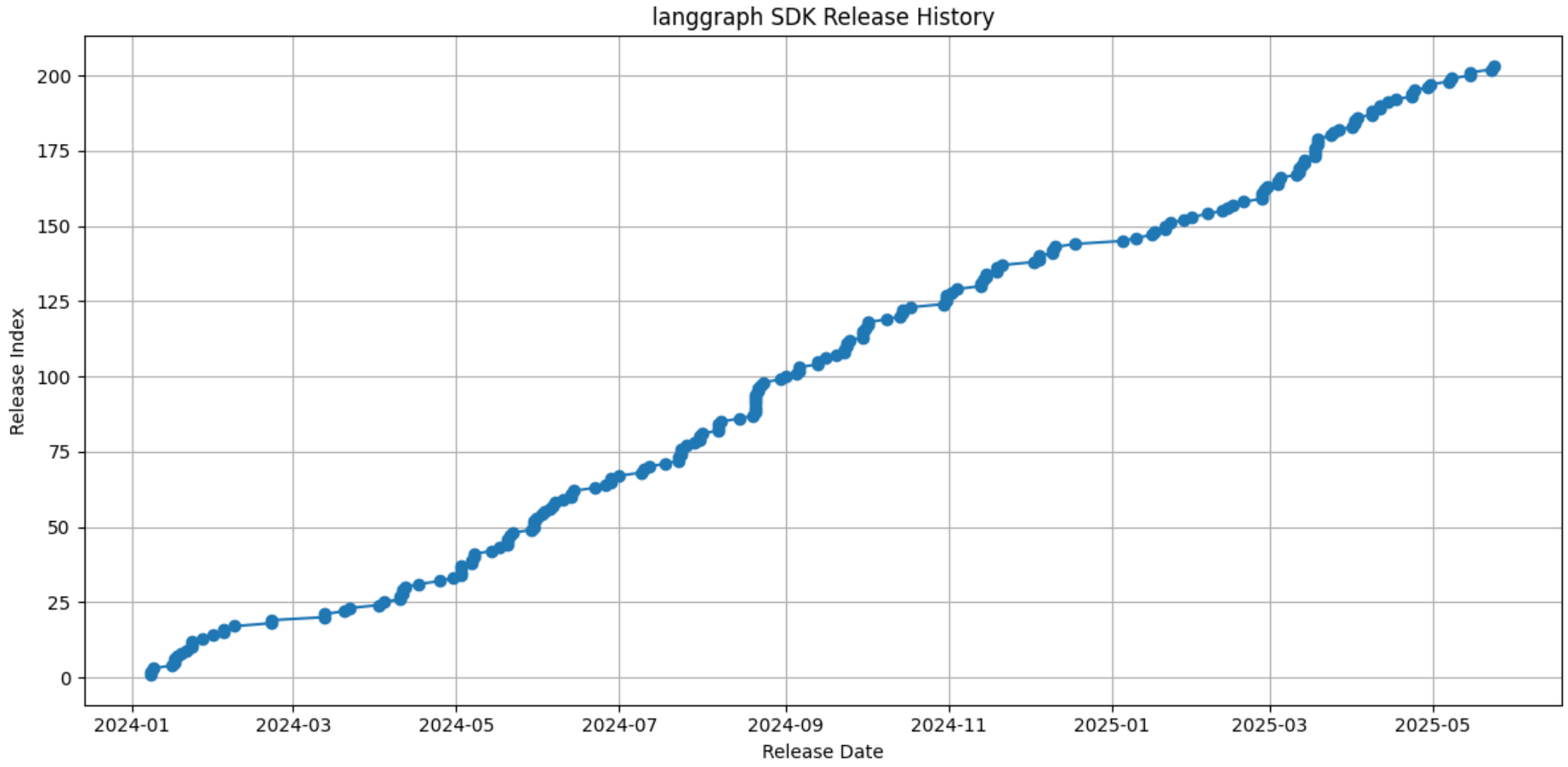
AIHUB-2170

Open AI SDK Releases



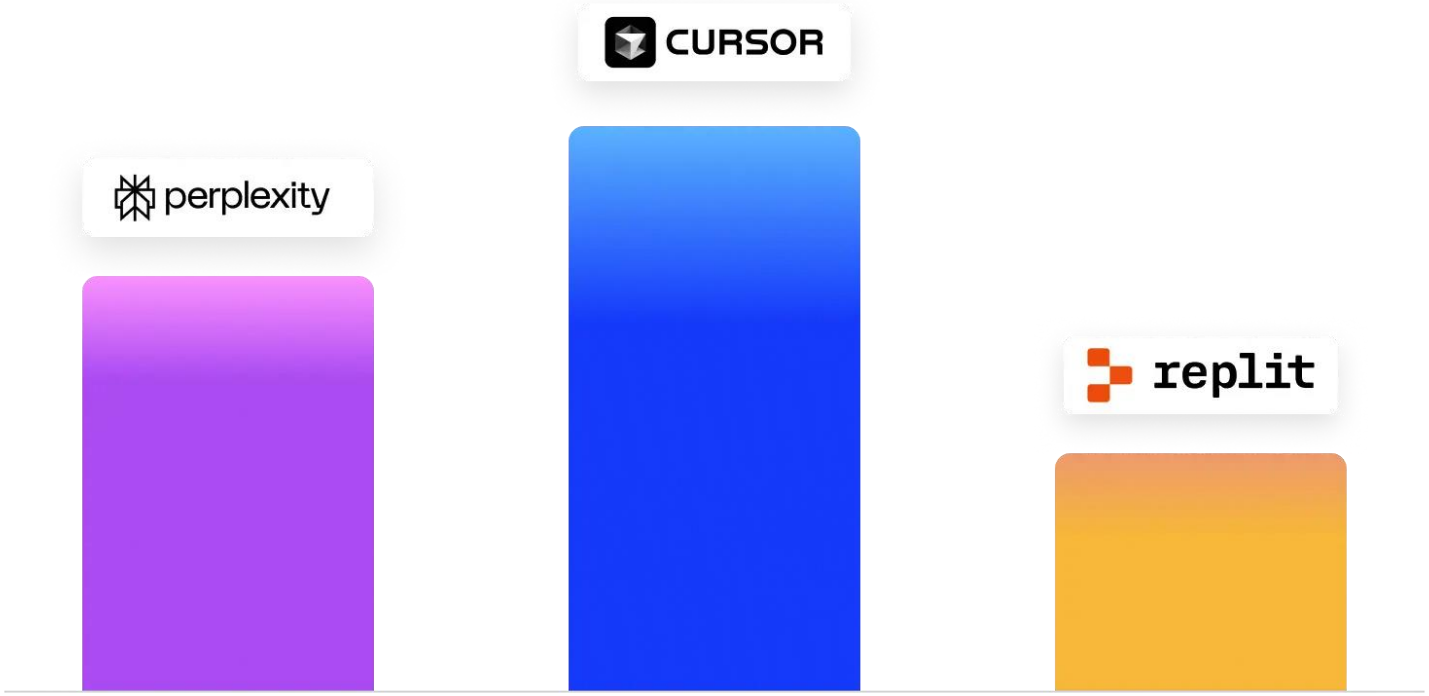
Release history as of May 29, 2025

LangGraph Releases



Release history as of May 29, 2025

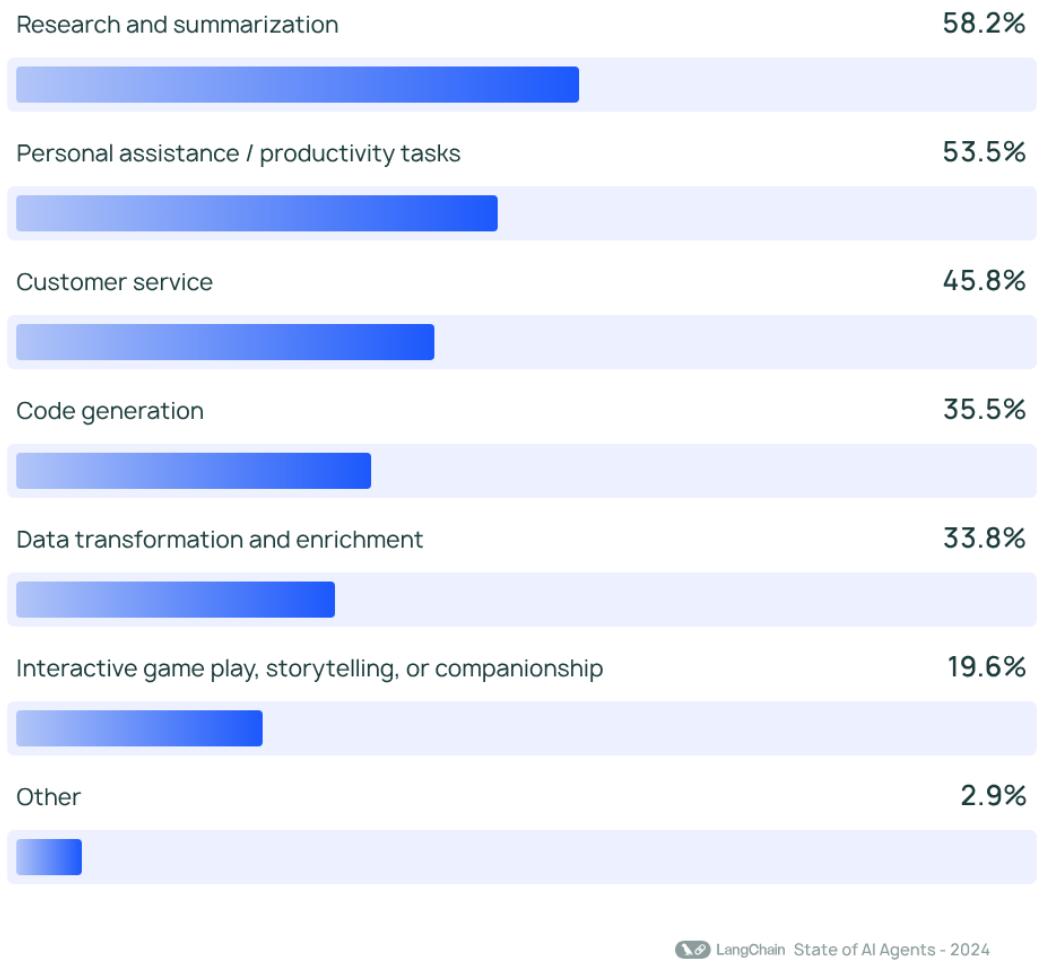
Agent Success Stories:
The buzziest AI agent Applications



LangChain State of AI Agents - 2024

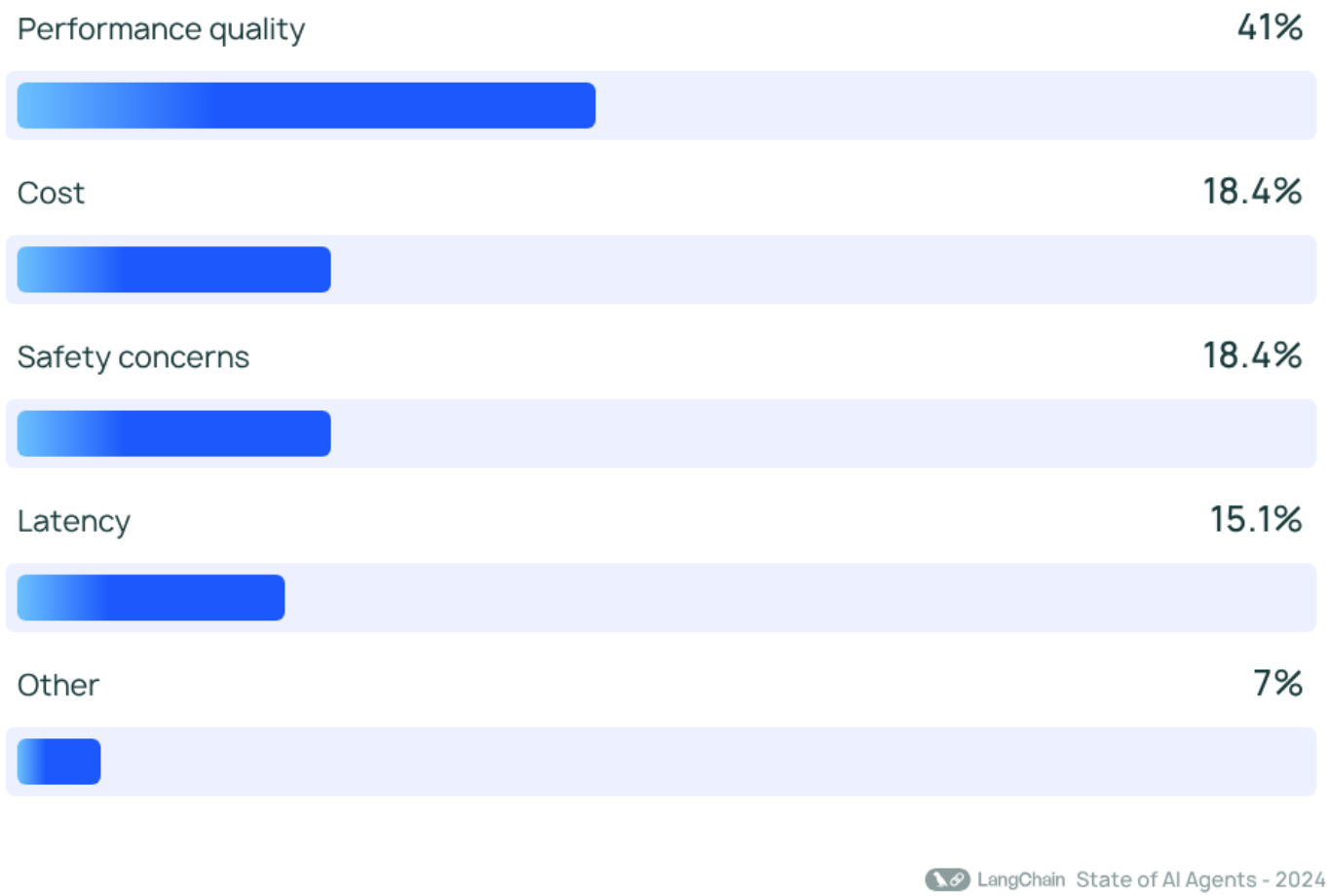
https://www.langchain.com/stateofaiagents

Survey: Which tasks are agents best suited to perform?



https://www.langchain.com/stateofaiagents

“What is your biggest limitation of putting more agents in production?”



https://www.langchain.com/stateofaiagents

We’re in the middle of a transition

- Historical Parallels of Technological Shifts
  - *1980/90s Software Boom*: Success demanded embracing new programming paradigms with full engagement.
  - *Internet Revolution*: Companies needed experts in web technologies; lack of investment led to struggles.
- Current Paradigm Shift in GenAI
  - Developing intelligent agents and reasoning models is central, not peripheral.
  - Mastery of tools and orchestration is crucial.
  - Dedicated teams are necessary to stay competitive in AI-driven applications.

Agentic Frameworks: A brief look

What can an Agentic AI Framework offer?

“Prompt Engineering”

- Agent Abstractions (standardized definitions, execution models, control flow patters incl. reflection/self-improvement, template management)
- Tool Integration and Management
- Planning and Decomposition

Agent Ensemble Operations

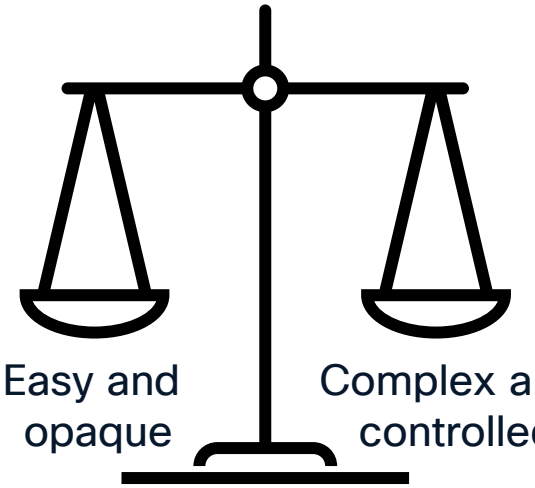
- Memory and Context Management (incl. RAG)
- Orchestration and Multi-Agent Coordination, incl. routing data to LLM
- Logging, Tracing, Debugging Tools
- Scalability and Deployment Support

[..] a lot of people feel that using a full framework is overkill.

The issue is: if the LLM doesn't use the tool correctly or something breaks, the abstraction becomes a pain because you can't debug it easily. This can also be a problem if you switch models – the system prompt might have been tailored for one and not transfer well to others.

Some frameworks are lighter, some heavier and offer additional features, but there's community around them to help you get started. And once you learn one (including how it works under the hood), it becomes easier to pick up others.

Ida Silfverskiöld
https://www.ilsilfverskiold.com/articles/agentic-al-comparing-new-open-source-frameworks

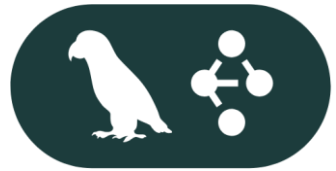


Intercom

Promptflow

Autogen

Semantic Kernel



LangGraph

Mastra

... and many more

Qualities of Frameworks to Consider

Communication / Messages

Use of different **Models, Tools**

Types of Agents / pre-defined and custom Agents

Human in the Loop

Teams of Agents

Orchestration: Collaboration Control / Workflows, Collaboration Patterns and Termination

State management / Memory (persistence, fault tolerance, knowledge gathering)

Ease of use / Learning curve / Professional Focus (Exploration, Production,..)

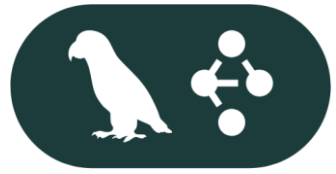
… Focus of Frameworks differs

Swarm
(OpenAI)

Autogen
(Microsoft)

CrewAI
(CrewAI)

Semantic Kernel
(Microsoft)



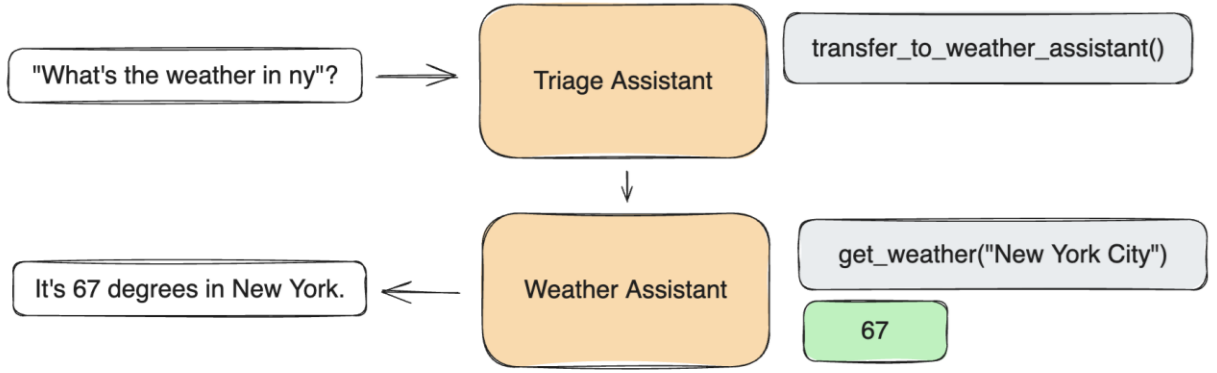
LangGraph
(LangChain)

Swarm (by OpenAI)

Oct/2024

“OpenAI's Swarm experimental framework addresses the complexities inherent in coordinating multiple AI agents. By introducing abstractions like Agents and Handoffs, Swarm provides a structured approach to agent interactions, promoting modularity and reusability in AI system development”.

https://github.com/openai/swarm



A few features to highlight

Agent Abstractions – Agents encapsulate instructions and tools functioning as an autonomous unit.

Handoff Mechanisms – Handoff enables an agent to transfer a conversation or task to another agent seamlessly. Enables dynamic workflows and task delegation among agents.

Stateless Design – simplifies system architecture; relies entirely on Chat Completions API.

Swarm - References

OpenAI Cookbook: Orchestrating Agents: Routines and Handoffs

OpenAI Swarm GitHub Repository: Swarm GitHub

OpenAI Developer Documentation: Swarm Overview

OpenAI Developer Forum: Swarm Discussions

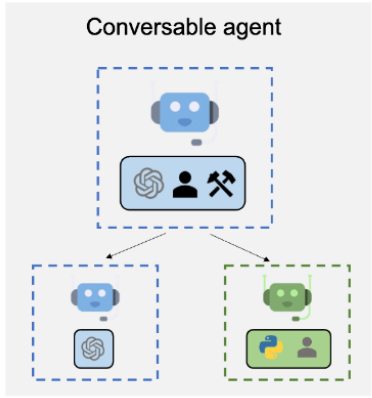
FutureSmart Blog: OpenAI Swarm Hands-On Introduction

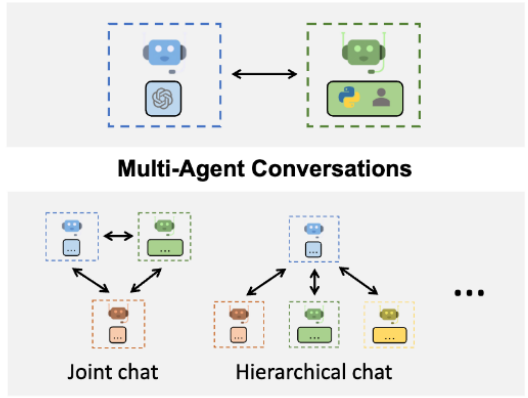
Autogen (by Microsoft)

Oct/2023

"AutoGen is a framework for simplifying the orchestration, optimization, and automation of LLM workflows. It offers customizable and conversable agents that leverage the strongest capabilities of the most advanced LLMs, like GPT-4, while addressing their limitations by integrating with humans and tools and having conversations between multiple agents via automated chat."

https://github.com/microsoft/autogen





https://www.microsoft.com/en-us/research/blog/autogen-enabling-next-generation-large-language-model-applications/

A few features to highlight

Multi-Agent Conversation Framework – Agents collaborate human-like on tasks, complex workflows are simplified

Enhanced LLM Inference and Optimization – ensure that agentic applications are efficient and cost effective.

Techability and Personalization – enable teachable, personalized agents, resulting in intuitive, user-friendly applications

Modular and Extensible Design – extend the framework to your needs, if you need to and are able to do so.

Autogen - References

Microsoft AutoGen GitHub Repository: AutoGen GitHub

Microsoft Documentation: AutoGen Overview

Microsoft Research Blog: AutoGen and Next-Gen AI Applications

AutoGen Community Tutorials: AutoGen Use Cases

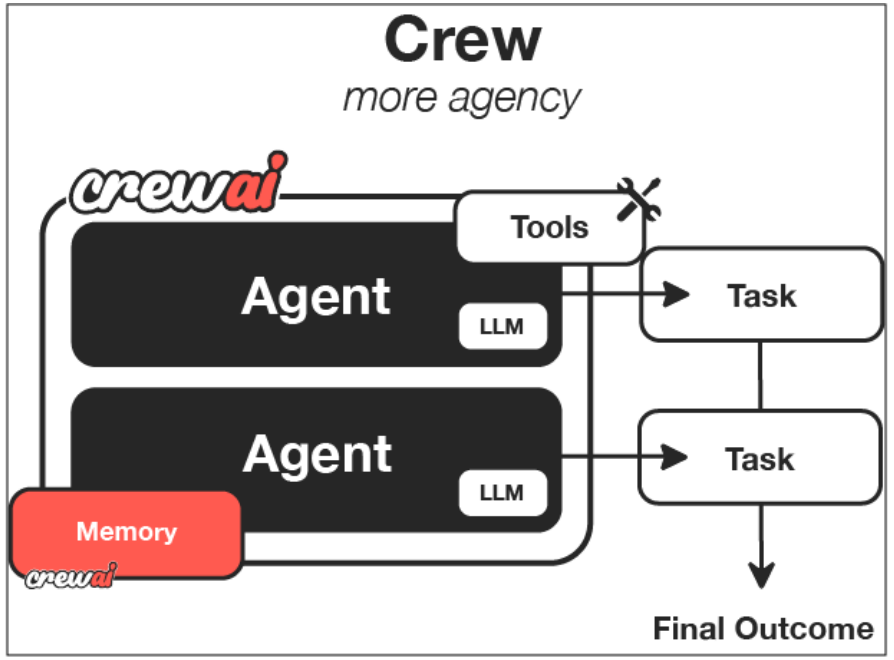
CrewAI

"CrewAI is a lean, fast Python framework built entirely from scratch—completely independent of LangChain or other agent frameworks.

CrewAI empowers developers with both high-level simplicity and precise low-level control, ideal for creating autonomous AI agents tailored to any scenario"

https://docs.crewai.com/introduction

https://github.com/crewAIInc/crewAI



CrewAI – A few things to highlight

Nov/2023

- **Role-Based Agents**: Create specialized agents with defined roles, expertise, and goals–from researchers to analysts to writers.
  Agents are defined by role, goal, backstory, tools and LLM
- **“Autonomous Crews” and “Flows”**: Crews for “open-ended research or code generation”, Flows for “decision workflows or API orchestration”.
  - Crews: Optimize for autonomy and collaborative intelligence, enabling you to create AI teams where each agent has specific roles, tools, and goals.
  - Flows: Enable granular, event-driven control, single LLM calls for precise task orchestration and supports Crews natively. Define sequential or hierarchical workflows, with agents automatically handling task dependencies.
- **Extensible Design**: Easy to add new tools, roles, and capabilities. Leverages LiteLLM for LLM provider integration.

CrewAI - References

CrewAI GitHub Repository: CrewAI GitHub

CrewAI Documentation: CrewAI Introduction

CrewAI Community: CrewAI Community

Semantic Kernel (by Microsoft)

Mar/2023

"At its simplest, the kernel is a dependency injection container that manages all of the services and plugins necessary to run your AI application."

"Semantic Kernel is Microsoft's initiative to streamline the integration of LLMs into traditional software development workflows. By providing a set of abstractions and tools, SK allows developers to harness the power of AI models without delving into the complexities of prompt engineering or model management. This SDK is particularly beneficial for enterprises aiming to incorporate AI capabilities into their existing systems efficiently."

https://github.com/microsoft/semantic-kernel

A few features to highlight

Prompt Management: Prompt management features to define, store and reuse prompts effectively. This reduces the unpredictability of AI responses and ensures consistent output across scenarios.

Planner, Orchestration, Memory Integration: SK includes a planner component to dynamically generate action sequences. Orchestrates tasks based on user-inputs and context. Memory to maintain context across interactions, ensuring coherent and context-aware responses. Enables complex workflows.

Agent Framework (evolving): SK extension to incorporate agentic patterns into any application. Chat/Group-chat collaboration pattern. Evolution to shared runtimes for SK and Autogen*.

*Autogen and Semantic Kernel:
https://devblogs.microsoft.com/semantic-kernel/semantic-kernel-and-autogen-part-2/

Semantic Kernel - References

Semantic Kernel GitHub Repository: Semantic Kernel GitHub

Microsoft Documentation: Semantic Kernel Overview

Semantic Kernel Tutorial: Quick Start Guide

Galileo AI Blog: Semantic Kernel Analysis

Semantic Kernel GitHub Discussions: Community Engagement

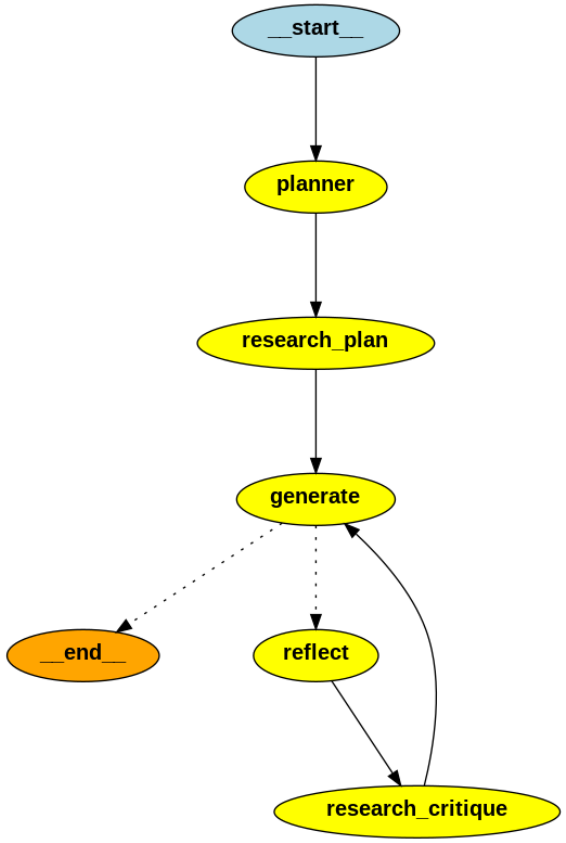
LangGraph (by LangChain)

LangGraph

Jan/2023

"LangGraph is an open-source library developed by LangChain to facilitate the construction of stateful, multi-agent applications utilizing Large Language Models (LLMs). It extends LangChain's capabilities by introducing graph-based workflows, enabling the creation of complex, cyclic processes essential for advanced AI agent architectures."

https://github.com/langchain-ai/langgraph



A few features to highlight

- **Graph-based Workflows**
  - Agent interactions are workflows (inspired by Pregel and Apache Beam) built as directed graphs, where each node represents a specific task or function. Allows explicit control over the sequence of agent interactions, accommodating deterministic and dynamic control flows.
- **State Management**
  - Built-in statefulness – allowing agents to maintain context across interactions: Error recovery, human in the loop, time-travel, workflow re-execution
- **Multi-Agent Collaboration**
  - Support for multiple agent interaction patterns, incl. hierarchical and sequential setups.
  - Distributed Agentic Applications – with LangGraph Agent Protocol and Remote Graphs

LangGraph References

LangGraph Official Documentation: LangGraph Documentation

LangChain AI GitHub Repository: LangGraph GitHub

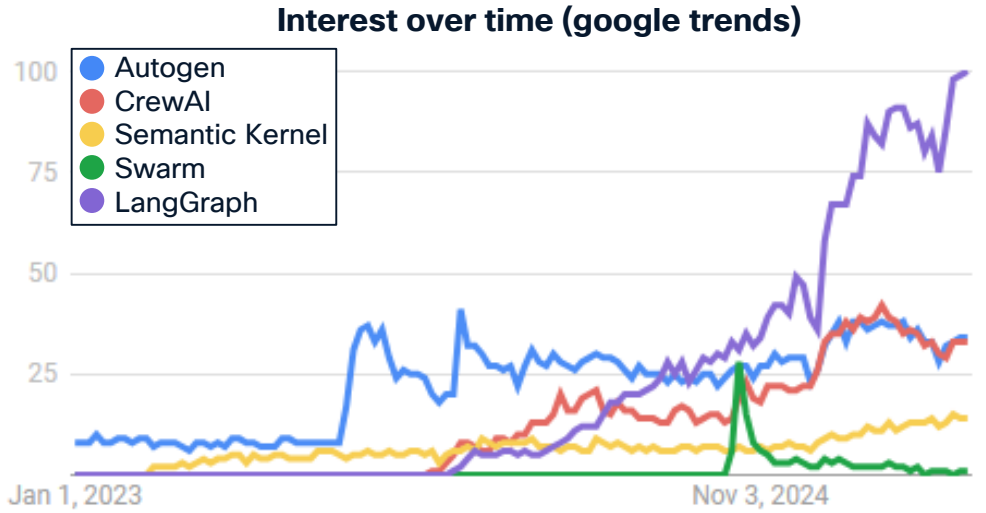
Galileo AI Blog: AutoGen vs. LangGraph

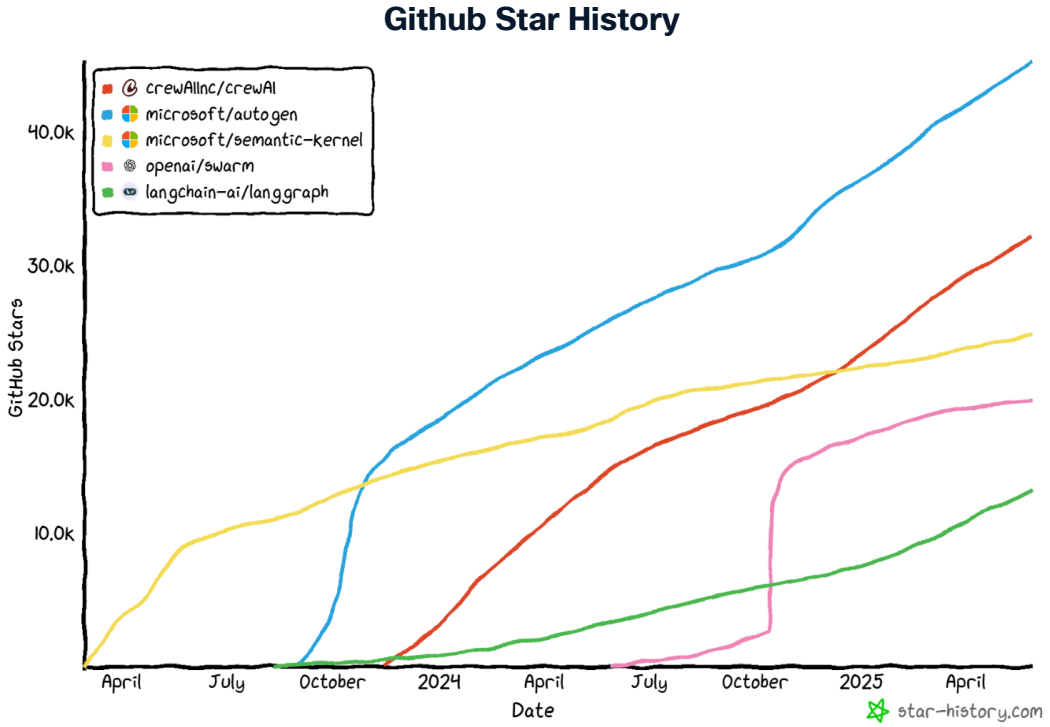
Analytics Vidhya Tutorial: LangGraph Applications

GitHub Discussions: Community Engagement

Usability

OpenAI Swarm	Emphasizes simplicity, making it accessible to developers familiar with AI concepts. Intuitive abstractions of Agents and Handoffs are intuitive
AutoGen	Designed to be user-friendly, making it accessible to developers with varying levels of experience.
CrewAI	Intuitive concepts (crews/agents/process/task), structured approach, templates – well suited for beginners.
Semantic Kernel	Designed with user-friendliness in mind. While knowledge of AI concepts is advantageous, the SDK is accessible even to those new to LLMs.
LangGraph	Developers have noted that while LangGraph offers powerful capabilities for constructing complex workflows, it may present a steeper learning curve compared to more other frameworks.





Agent Frameworks: No one size fits all

- **Swarm**: Exploration of lightweight, stateless multi-agent systems that include “hand-offs”. Best suited for simpler workflows, requiring modular, scalable interactions.
- **Autogen**: Conversational interactions, techability, inference optimization – highly dynamic and user-centric. Templates and pre-defined Agents.
- **CrewAI**: Beginner-friendly framework – with intuitive abstractions and support for “autonomous” crews and more controlled “flows”.
- **Semantic Kernel**: SDK to create conversational agents, with modular task orchestration and memory management. Integrate AI into traditional code. Evolved to adopt a lot of LangChain concepts.
- **LangGraph**: Graph-based workflows for stateful, intricate multi-agent applications – well suited for real-world enterprise class solutions.

[..] the main value that should be provided by a framework [..]:

a reliable orchestration layer that gives developers explicit control over what context reaches their LLMs while seamlessly handling production concerns like persistence, fault tolerance, and human-in-the-loop interactions.

Harrison Chase,
https://blog.langchain.dev/how-to-think-about-agent-frameworks/

Agent Memory is a key consideration

Memory is a cognitive function that allows people (and agents) to store, retrieve, and use information to understand their present and future.
(Consider a human who either forgets everything – or who remembers everything…)

Versatile agent memory management is required to adapt to the specifics of a deployment and for efficiency

- Semantic Memory (topic knowledge), Episodic Memory (experience), Procedural Memory (rules)

Short term memory (thread / single conversation scoped memory) – typically managed as part of the agent state ("checkpointer")

- Edit past threads (incl. selective forgetting), summarize past threads, replay threads (time travel)

Long term memory (shared across threads / conversations);
includes **collective memory** (shared across agents and conversations) – typically managed in stores (databases, etc.)

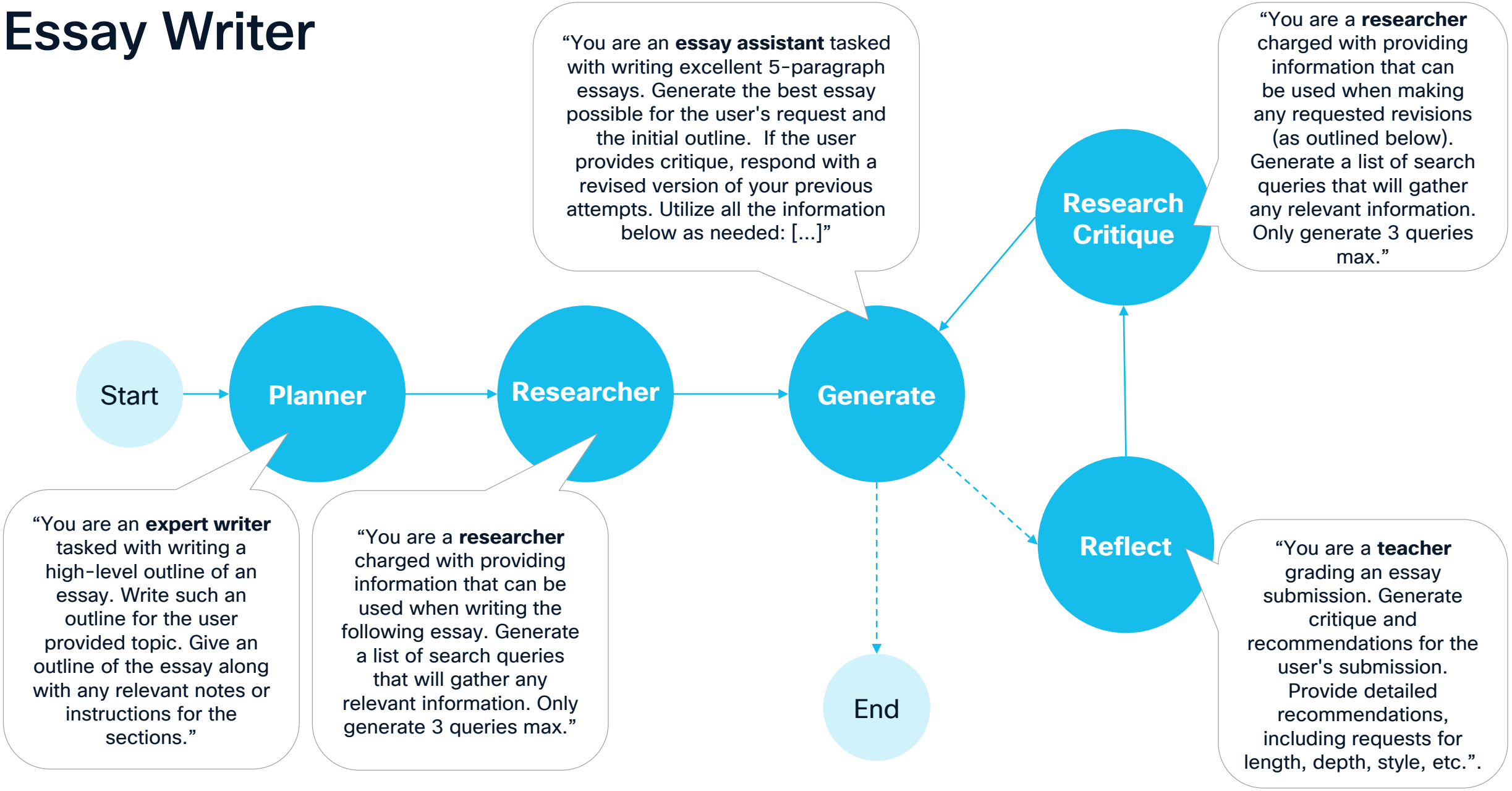
- Edit past results (incl. selective forgetting), summarize past results

See also: Cognitive Architectures for Language Agents (CoALA)
https://langchain-ai.github.io/langgraph/concepts/memory/

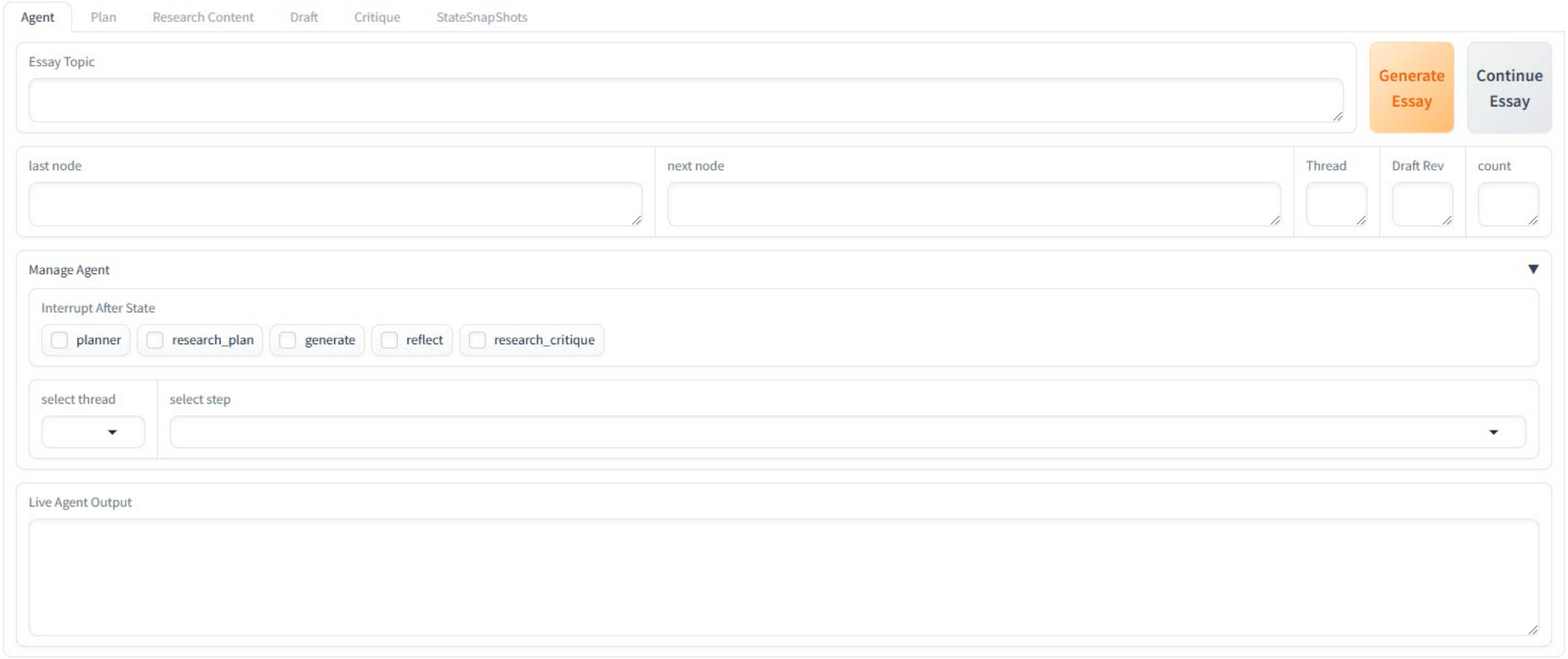
Apollo Research: Scheming reasoning evaluations

Example:
Essay Writer

Essay Writer



https://learn.deeplearning.ai/courses/ai-agents-in-langgraph



https://learn.deeplearning.ai/courses/ai-agents-in-langgraph

A few closing thoughts...

Building on Existing Frameworks: Accelerating Innovation in Agentic Applications

Leverage Existing Frameworks: Foundational tools like LangGraph and LangChain provide essential assets but require significant customization for operational agentic applications. Focus on building on top of these frameworks.

Efficiency Through Templates:

- Prebuilt agentic patterns (e.g., Reflexion, Chain-of-Thought) reduce development complexity.
- Reusable prompt libraries simplify crafting and iterating on prompts for common use cases.

Integrated Evaluation Tools: Embed performance metrics and monitoring directly into frameworks to enable real-time feedback and optimization.

Lessons from Industry:

- Web development: Platforms like WordPress enabled faster innovation by offering foundational structures.
- Mobile apps: Native SDKs (e.g., Android/iOS) encouraged differentiation through custom libraries and UX.

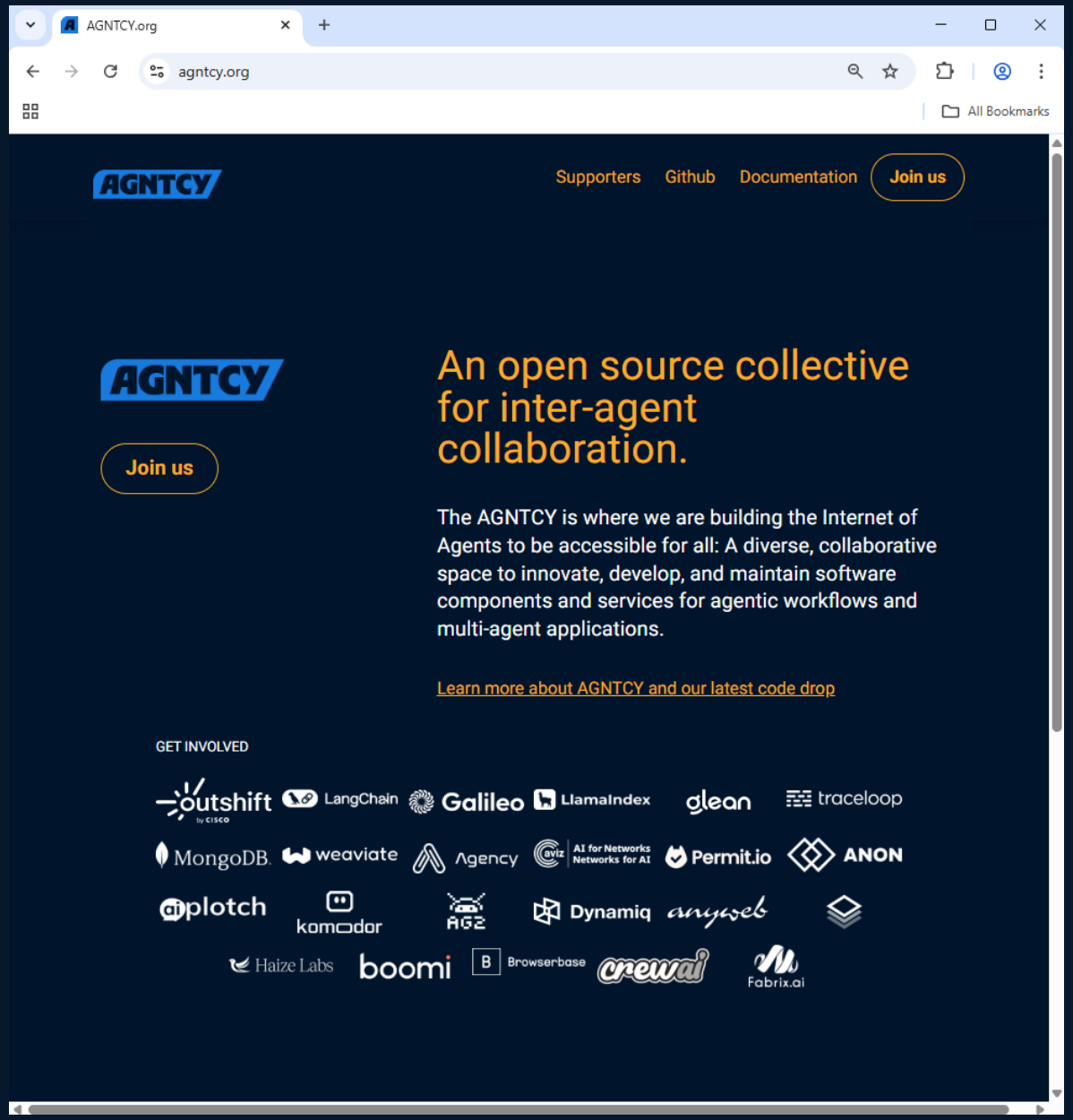
Key Takeaway: Avoid reinventing the wheel. Use existing frameworks to build modular, domain-specific tools and achieve faster time-to-market with reduced effort. Agentic frameworks provide sets of helpful features to shorten the route to success.

Some trends to watch

- Unified Agentic Architectures
- Self-optimizing Agents
- Higher-level Abstractions
- Multi-modal Agents
- Cross-domain Reasoning
- Personalized Agents

- Fault tolerance, Redundancy
- Security, Reputation, Operations
- Decentralized Agent Networks: “Internet of Agents”
- Inter-Agent Communication Protocols

See also https://outshift.cisco.com/blog/the-next-wave-beyond-LangChain-and-LangGraph-in-the-agentic-ecosystem



- Agent Identity, Announcement, Discovery and Verification
- Agentic ensemble workflows, composition, deployment and supervision
- Interoperability between agents from multiple agentic frameworks and contexts
- Agent to agent connectivity and interworking
- Agentic ensemble trust and security
- Agentic ensemble observability, evaluation, metering

... see also https://agntcy.org/ AGNTCY

Building Agentic Frameworks: Challenges and Requirements for Success

Beyond Matching Features

Rapid Development and Maintenance

Multidisciplinary Expertise

Community Engagement

Accessibility and Education

Technical Excellence and Long-Term Vision

Complete your session evaluations

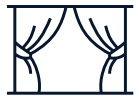
Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn exclusive prizes!

Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos

Book your one-on-one Meet the Engineer meeting

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Thank you

cisco Live !


CISCO