AI Agents and Agentic Frameworks: An Overview

Frank Brockners, Reinaldo Penno - Distinguished Engineers

AIHUB-2170

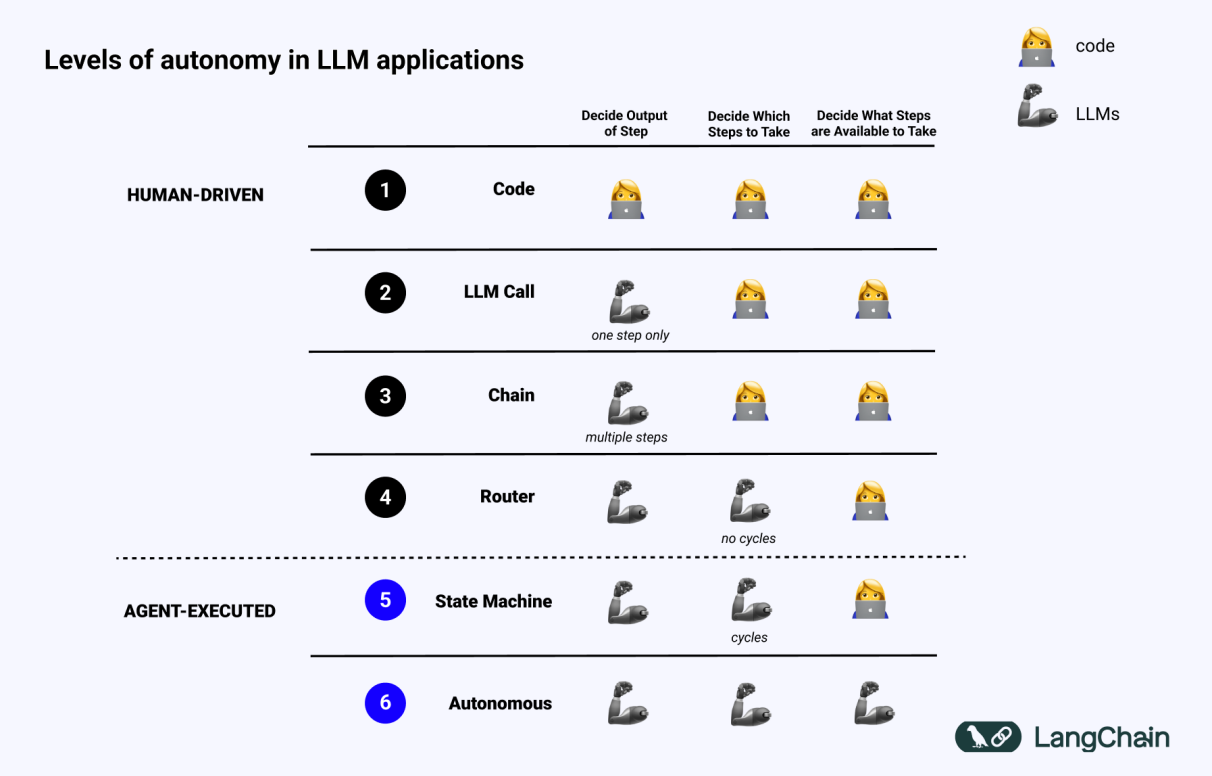
What is “Agentic AI”?

First, what even is an agent?

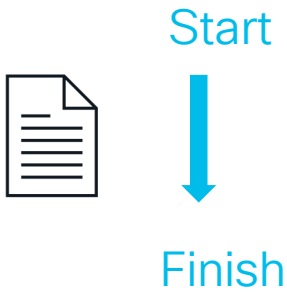
At LangChain, we define an agent as a system that uses an LLM to decide the control flow of an application. Just like the levels of autonomy for autonomous vehicles, there is also a spectrum of agentic capabilities.

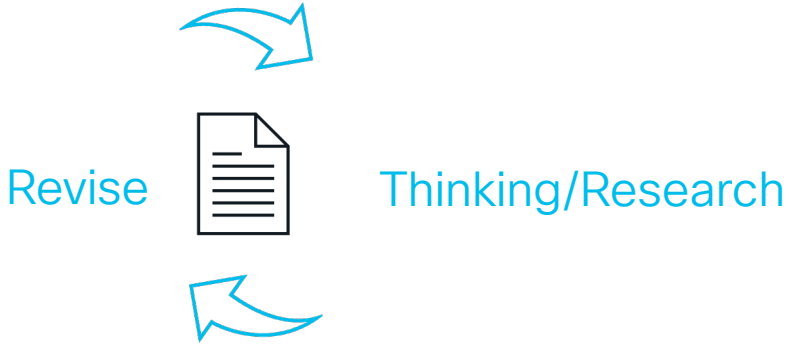
https://www.langchain.com/stateofaiagents
https://blog.langchain.dev/what-is-an-agent/

Levels of autonomy in LLM applications



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





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”

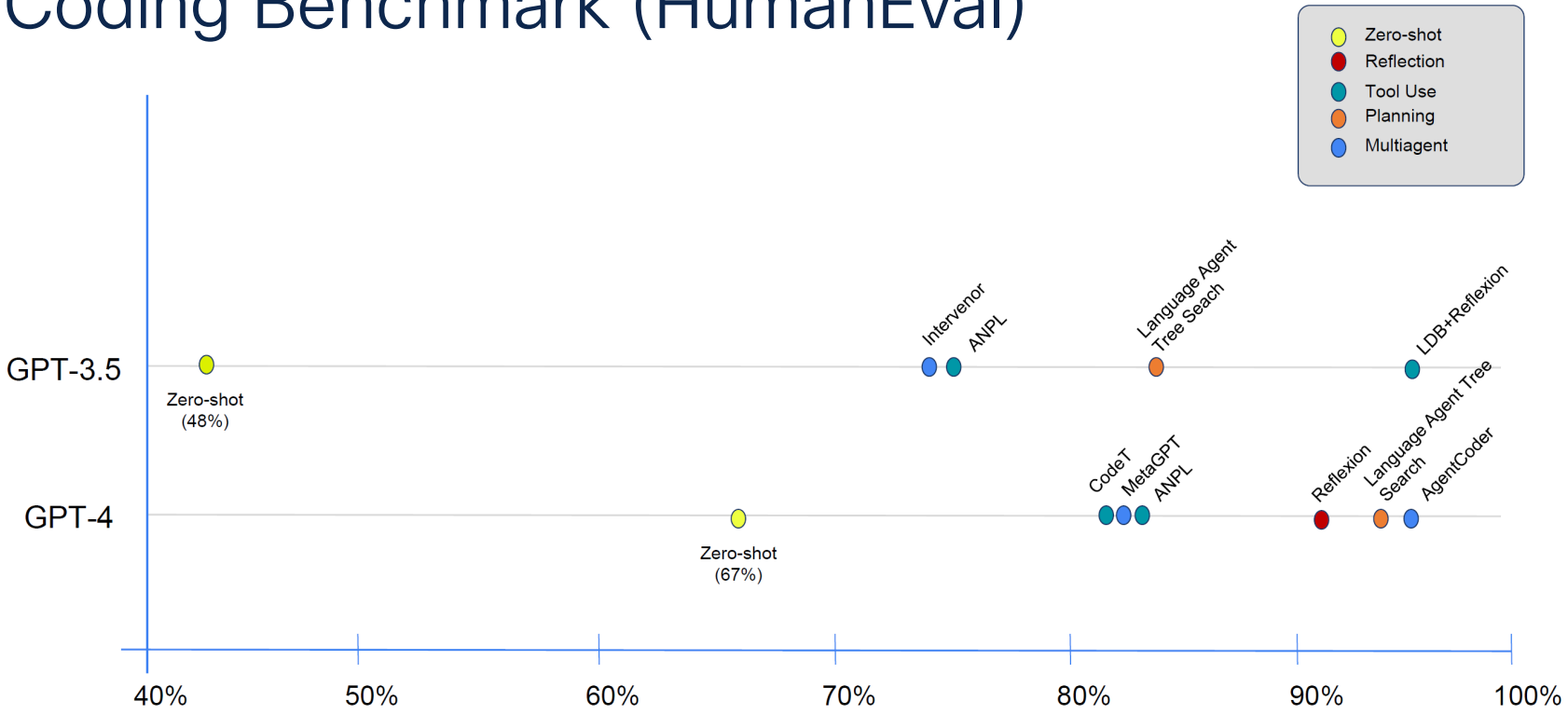
“Consider what parts need revision and more research”

“Revise your draft”

….

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

Coding Benchmark (HumanEval)



[Thanks to Joaquin Dominguez and John Santerre (DeepLearning.AI) for help with analysis.]

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

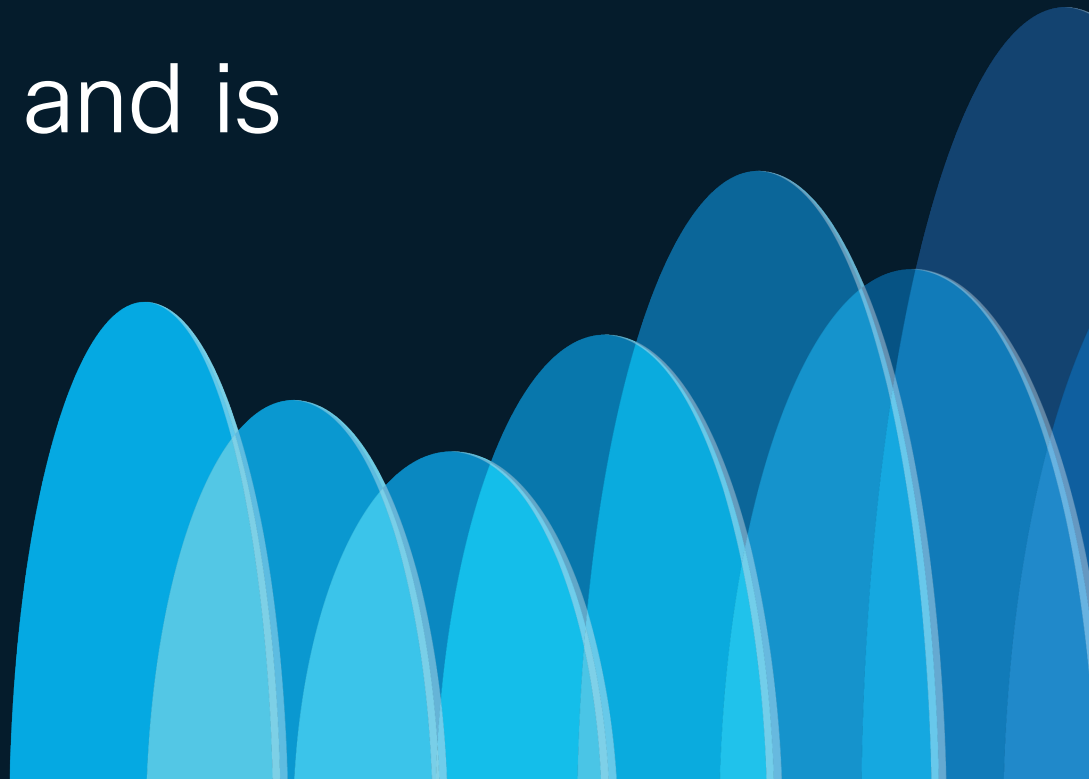
Collaboration

Multiple agents collaborate and communicate

Memory

Track progress/ results and learn individually/ collectively

Agentic AI is hot and is getting hotter



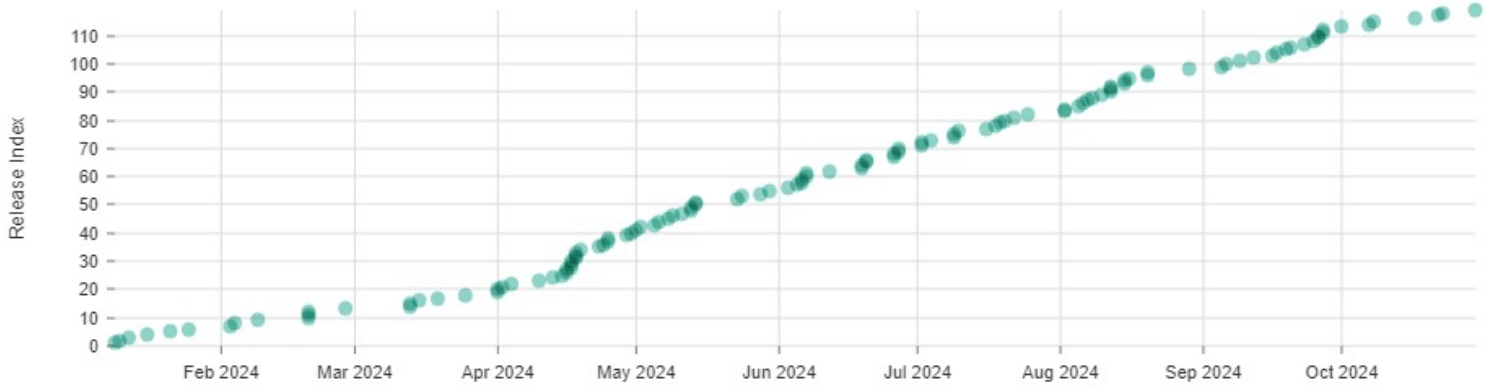
“In the first three quarters of this year, GenAI startups secured over $20 billion, according to S&P Global Market Intelligence data. That puts 2024 on track to exceed the 2023 total of $22.7 billion.”

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

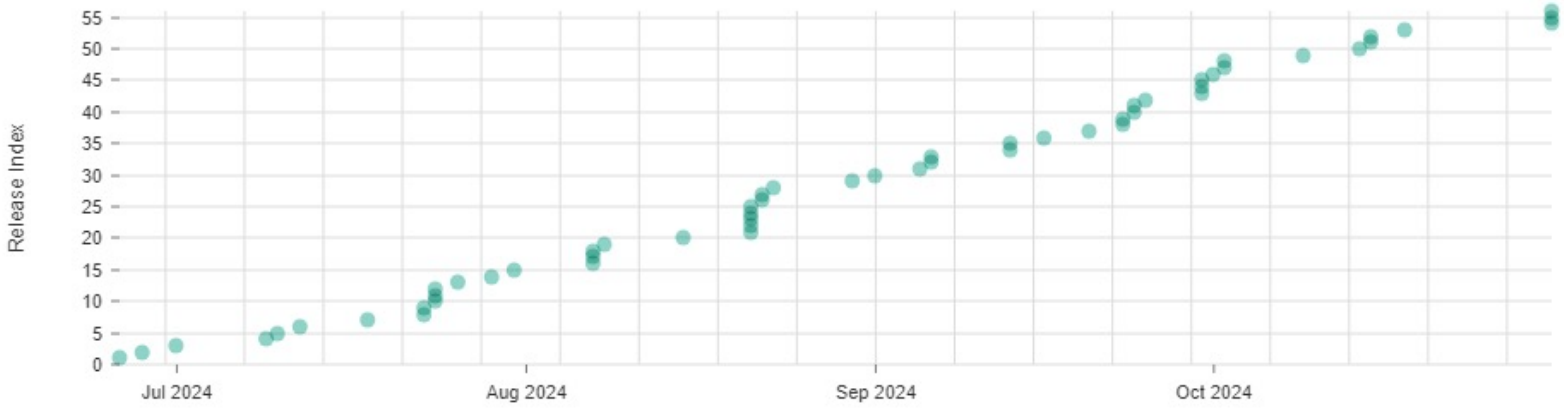
https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/genai-funding-on-track-to-set-new-record-in-2024-85779779

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

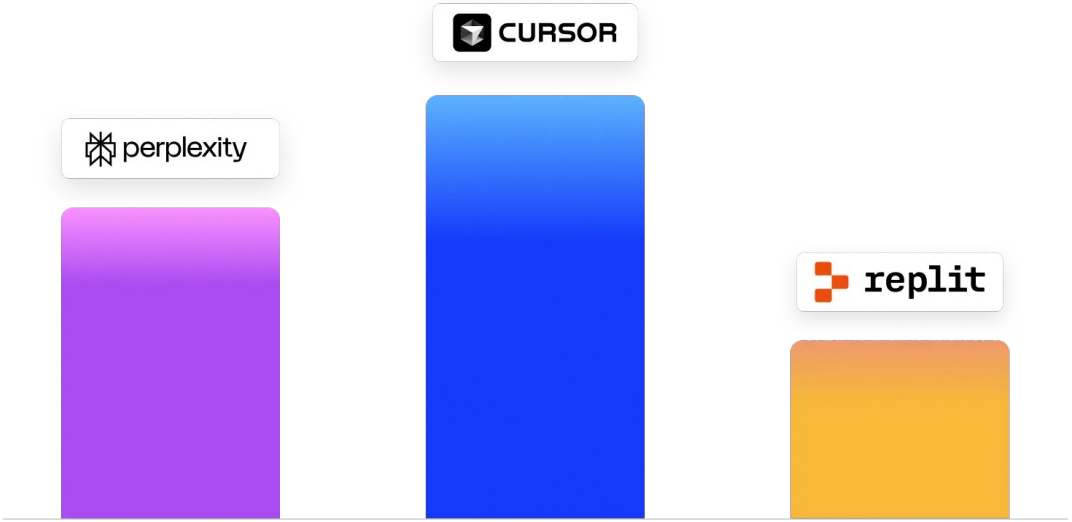
Open AI SDK Releases



LangGraph Releases



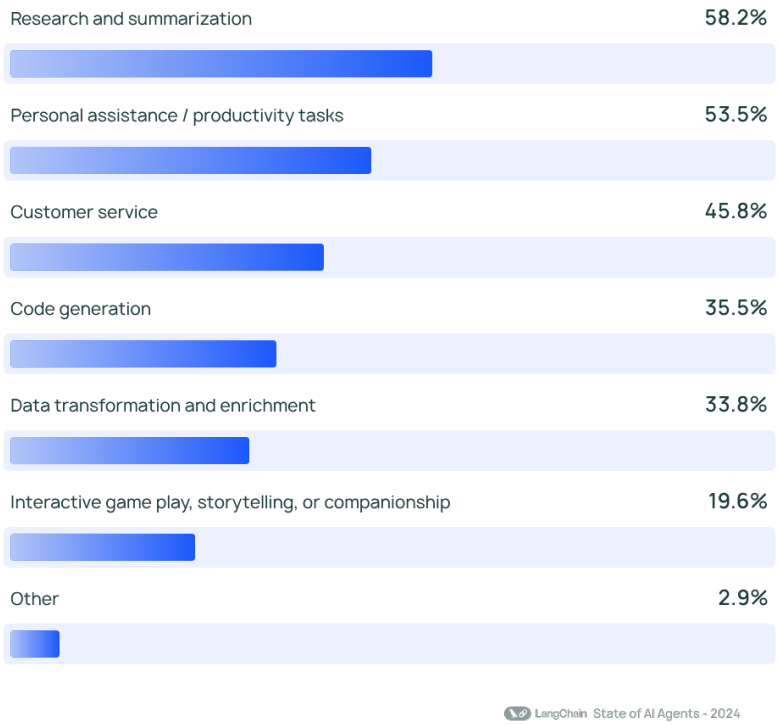
Agent Success Stories:
The buzziest AI agent Applications



LangChain State of AI Agents - 2024

https://www.langchain.com/stateofaiagents

Survey: Which tasks are agents best suited to perform?

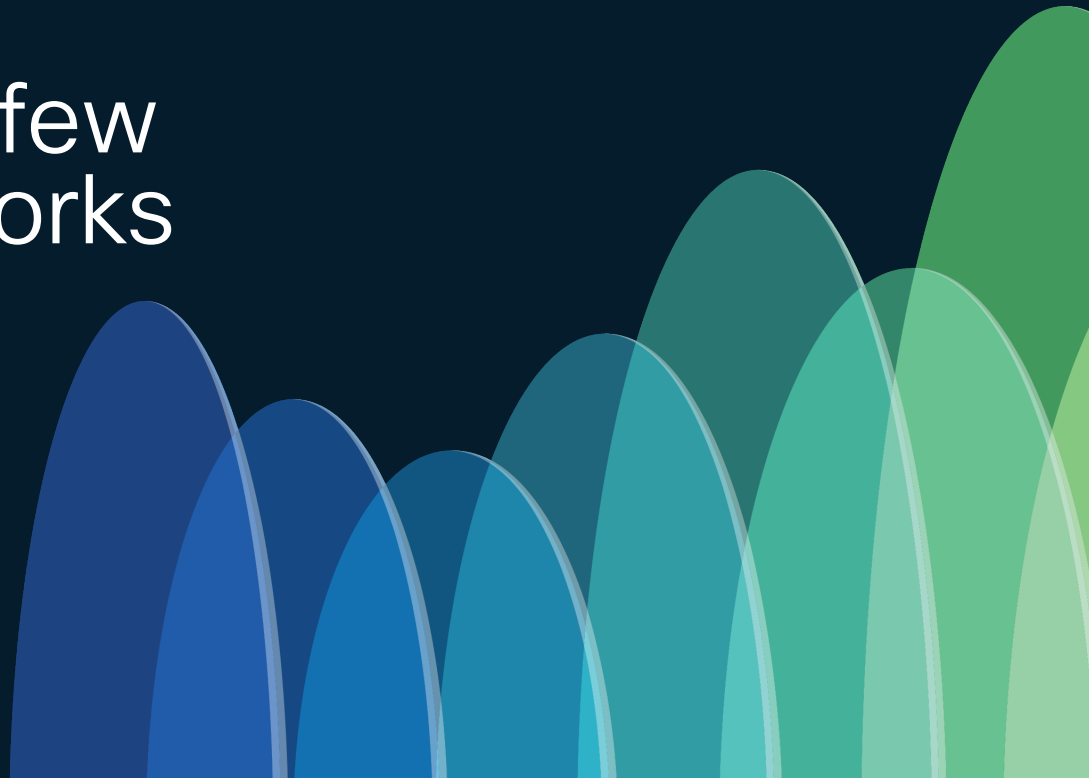


https://www.langchain.com/stateofaiagents

Advancing GenAI and Agentic Development Requires Full Commitment

- Deep Immersion Essential
  - Specialists must be fully dedicated to AI agents and systems.
  - Part-time efforts or sporadic tinkering are insufficient.
- 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.

A brief look at a few Agentic Frameworks



Autogen
(Microsoft)

Semantic Kernel
(Microsoft)

Swarm
(OpenAI)

LangGraph
(LangChain)

... 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
- Collaboration Control / Workflows, Collaboration Patterns and Termination
- State management / Memory

Letta phidata Intercom Promptflow

Autogen
(Microsoft)

Semantic Kernel
(Microsoft)

Swarm
(OpenAI)

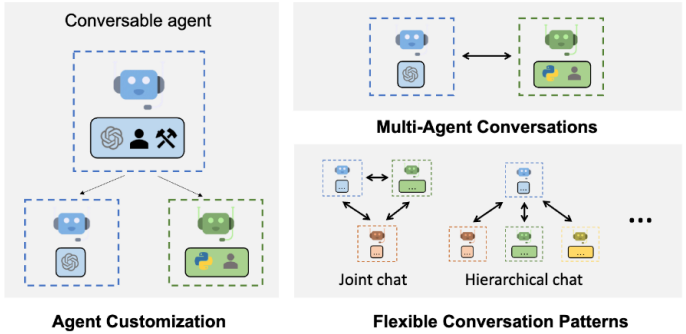
LangGraph
(LangChain)

LLAMA STACK RASA crewai LlamaIndex DSPy botpress lyzr ... and many more

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 - Usability

- **Learning curve**: Designed to be user-friendly, making it accessible to developers with varying levels of experience.
- **Extensible & customizable**; optimized for performance with efficient memory management and prompt execution.
- **Docs**: Extensive documentation provided by Microsoft; various tutorials/examples provided by the greater Autogen community.

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

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.
- **Memory Integration**: Memory simplifies context management for AI applications. Maintain context across interactions, ensuring coherent and context-aware responses. Enables complex workflows.
- **Planner, Orchestration**: SK includes a planner component to dynamically generate action sequences. Orchestrates tasks based on user-inputs and context.
- **Agent Framework (evolving):** SK extension to incorporate agentic patterns into any application. Chat/Group-chat collaboration pattern.
- **Modular and Extensible**: Extensible via Plugins

Semantic Kernel – Usability

- **Learning curve**: Designed with user-friendliness in mind. While knowledge of AI concepts is advantageous, the SDK is accessible even to those new to LLMs.
- **Extensible & customizable**: Plugins and adaptors to tailor the SDK to specific needs.
- **Docs**: Comprehensive documentation: Documentation and tutorials guide developers through the setup process and feature utilization.

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

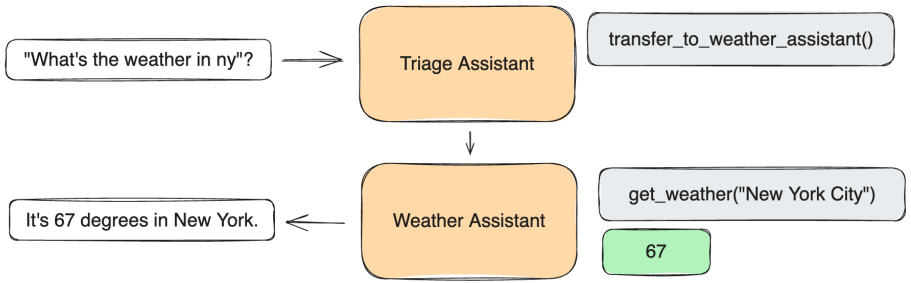
Swarm (by OpenAI)

Swarm

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.
- **Tool Integration** – for complex operations and interaction with external systems.
- **Stateless Design** – simplifies system architecture; relies entirely on Chat Completions API.

Swarm – Usability

- **Learning curve**: Emphasizes simplicity, making it accessible to developers familiar with AI concepts. The abstractions of Agents and Handoffs are intuitive, allowing for a relatively smooth learning experience.
- **Extensible & customizable:** Modular architecture facilitates seamless integration into existing systems. Developers can customize and extend functionalities
- **Docs**: Comprehensive documentation and tutorials (by OpenAI)

Swarm - References

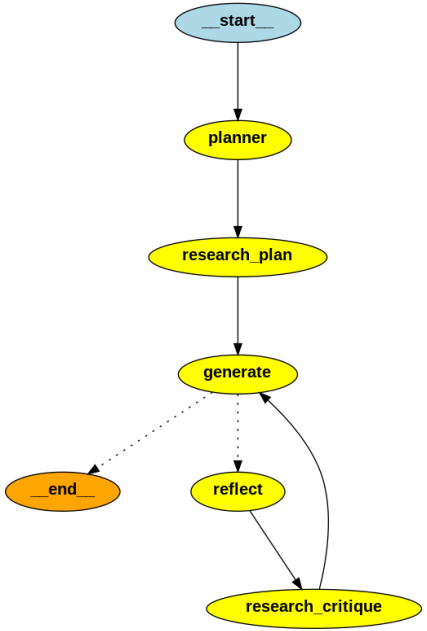
- OpenAI Cookbook: Orchestrating Agents: Routines and Handoffs
- OpenAI Swarm GitHub Repository: Swarm GitHub
- OpenAI Developer Documentation: Swarm Overview
- OpenAI Developer Forum: Swarm Discussions
- FutureSmart Blog: OpenAI Swarm Hands-On Introduction

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 – Usability

- **Learning curve**: 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.
- **Extensible & customizable:** integrates seamlessly with LangChain; focused on flexibility and control to tailor agent behavior to application specific needs.
- **Docs**: Comprehensive documentation (by Langchain)

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

Agent Frameworks: No one size fits all

- Autogen: Conversational interactions, techability, inference optimization – highly dynamic and user-centric.
- Semantic Kernel: SDK to create conversational agents, with modular task orchestration and memory management. Evolved to adopt a lot of LangGraph concepts.
- Swarm: Exploration of lightweight, stateless multi-agent systems that include “hand-offs”. Best suited for simpler workflows, requiring modular, scalable interactions.
- LangGraph: Graph-based workflows for stateful, intricate multi-agent applications – well suited for real-world enterprise class solutions.

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

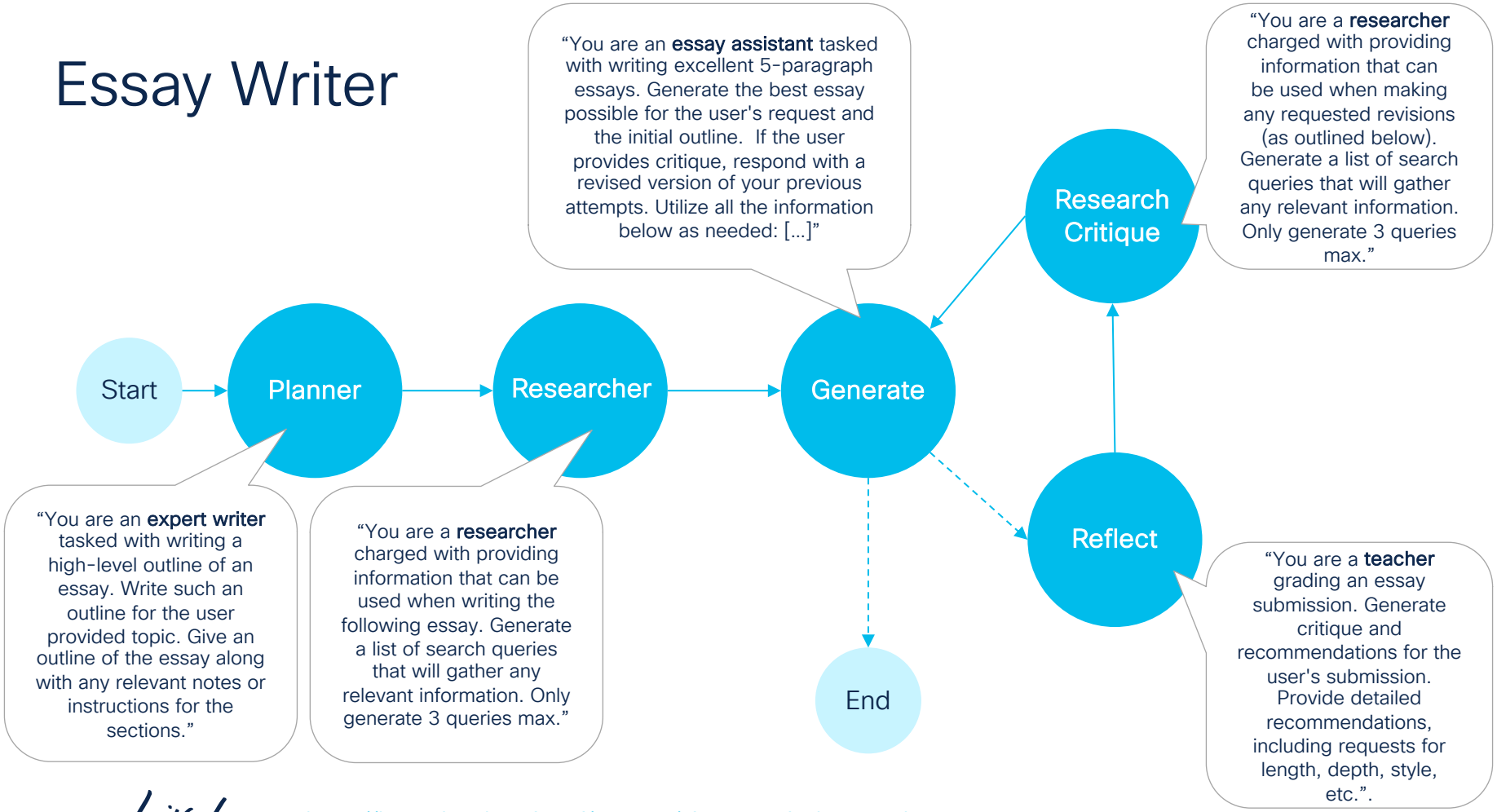
- Semantic Memory, Episodic Memory, Procedural Memory

- 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

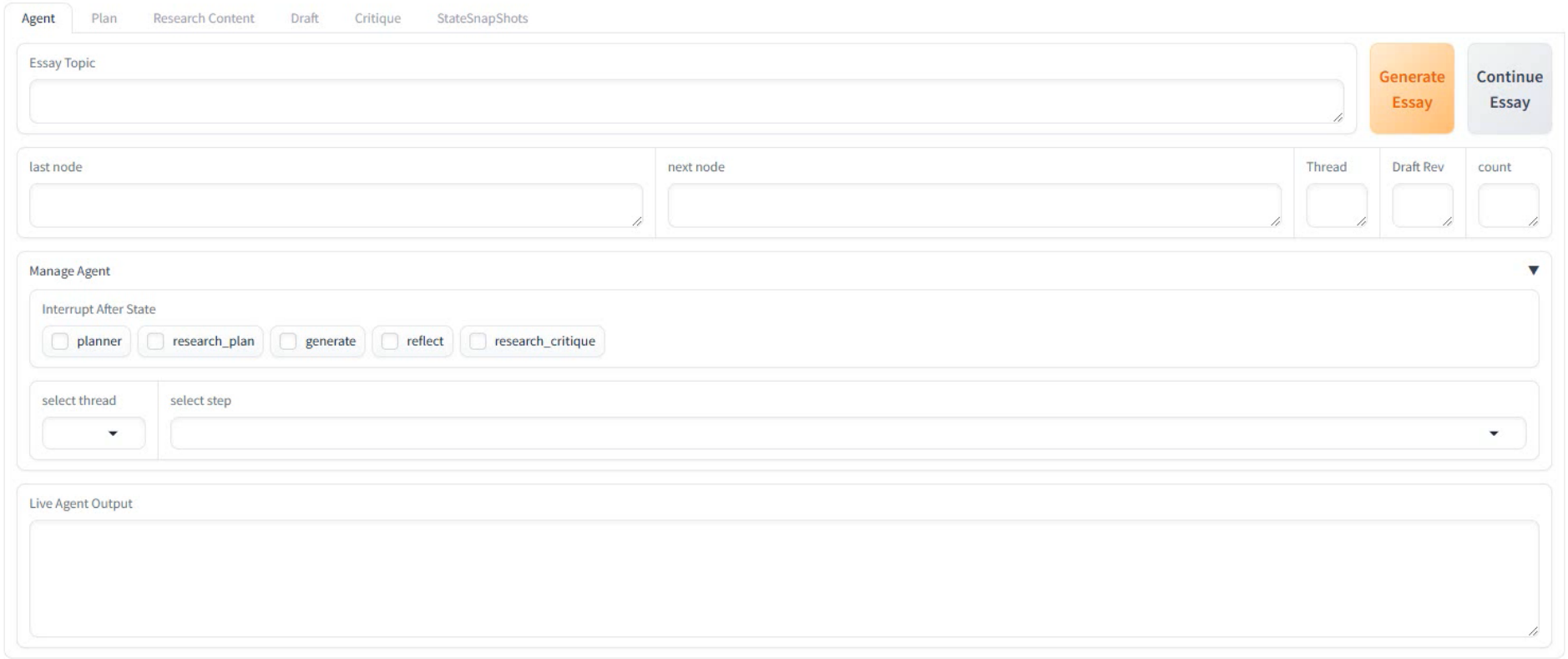
See also: Cognitive Architectures for Language Agents (CoALA)
https://langchain-ai.github.io/langgraph/concepts/memory/

Example: Essay Writer

Essay Writer



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



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

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.

Some trends to watch

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

- Decentralized Agent Networks
- Inter-Agent Communication Protocols
- Fault tolerance, Redundancy
- Security, Reputation, Operations

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

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

Webex App

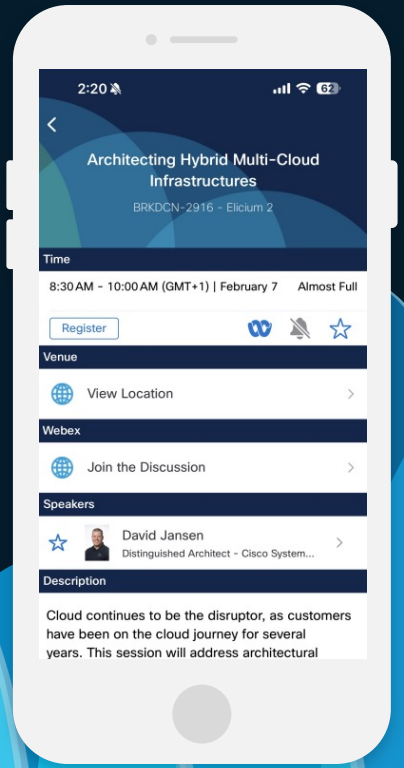
Questions?

Use the Webex app to chat with the speaker after the session

How

1. Find this session in the Cisco Events 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 February 28, 2025.



Fill Out Your Session Surveys

Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)

All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the ‘Participant Dashboard’

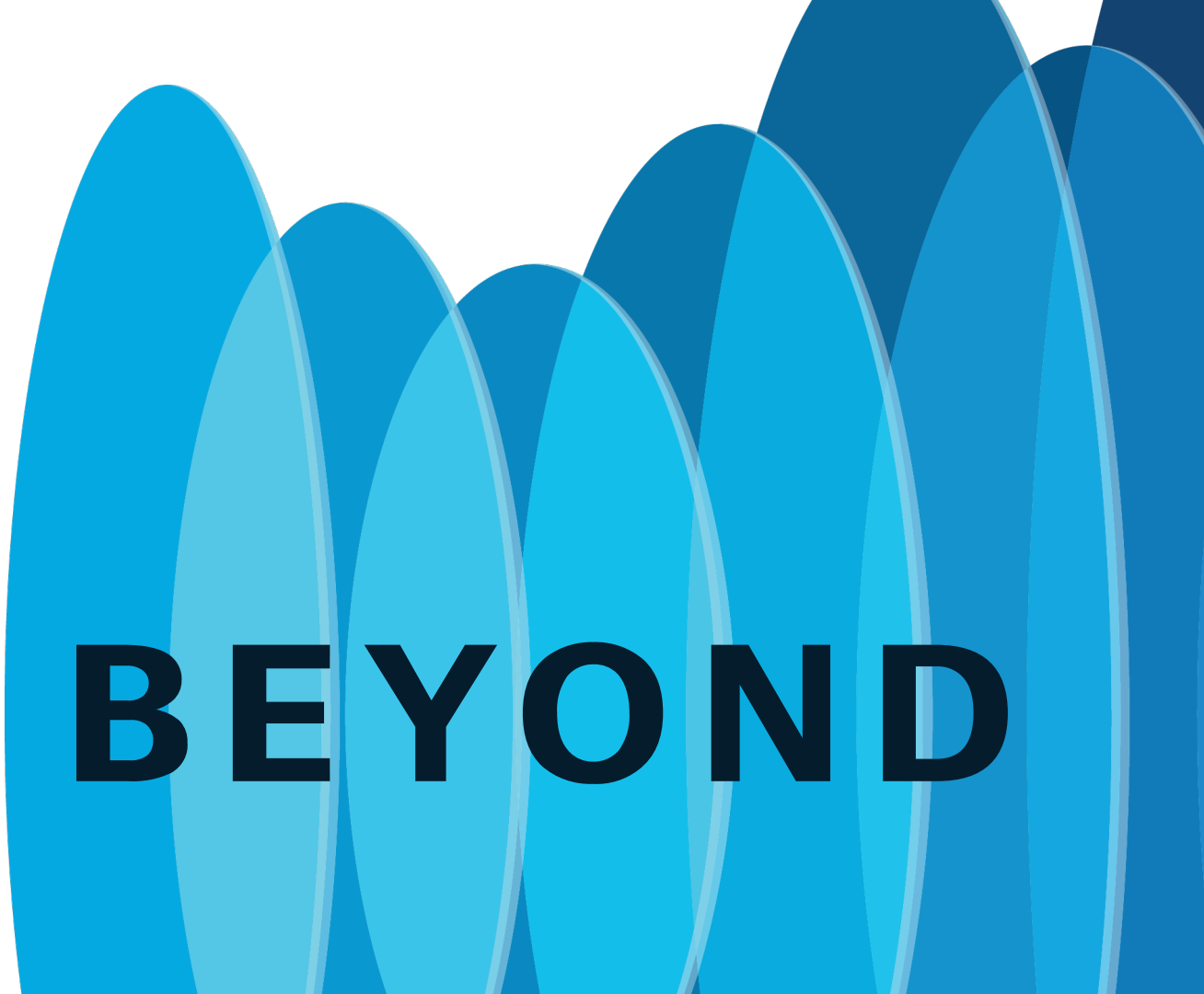
Content Catalog

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 ciscolive.com/on-demand. Sessions from this event will be available from March 3.


cisco
Thank you
cisco Live!

cisco Live!



GO BEYOND

Additional References

- Semantic Kernel
  https://www.deeplearning.ai/short-courses/microsoft-semantic-kernel/
  - Sam Schillace (inventor of google docs, and MSFT semantic kernel):
    https://devblogs.microsoft.com/semantic-kernel/early-lessons-from-gpt-4-the-schillace-laws/
- Autogen
  https://www.deeplearning.ai/short-courses/ai-agentic-design-patterns-with-autogen/
- Langgraph
  https://www.deeplearning.ai/short-courses/ai-agents-in-langgraph/

- Langgraph vs. Autogen vs. Crew vs. Swarm https://dev.to/exemplar/ai-agents-langgraph-vs-autogen-vs-crew-ai-key-differences-1di7
- Langgraph, Semantic Kernel, Autogen https://medium.com/data-science-at-microsoft/harnessing-the-power-of-large-language-models-a-comparative-overview-of-langchain-semantic-c21f5c19f93e
- Autogen and Semantic Kernel https://devblogs.microsoft.com/autogen/microsofts-agentic-frameworks-autogen-and-semantic-kernel/