

Microsoft Partner Project Ready

Agentic AI Accelerator

■ Day 3 of 4

<Presenter Name>



Course Plan and Learning Objectives

Day 1

Showcasing AI Potential with Agentic AI

Module 1: Innovate with Microsoft 365 Copilot and agents

- M365 Copilot and agents
- How Copilot works
- Semantic index for Copilot
- M365 Copilot Chat
- Agents Use cases
- New agents in Microsoft 365
- Unlock more value with SharePoint agents

Module 2: Extend Microsoft 365 Copilot with Agents

- Microsoft 365 Extensibility Planning and approach
- Declarative agents and agent tooling
- Explore Copilot Studio Agent Builder
- Build declarative agents with Microsoft 365 Agents Toolkit
- Build custom agents with Copilot Studio
- Autonomous agents overview
- Agent Governance - Overview
- Gen-AI decision guide – when to build, buy or extend

Hands-on Labs

- Explore Copilot Studio Agent Builder
- Build HR Assistant Agent with Copilot Studio
- Incorporate actions in HR Agent
- Enable Autonomous Capabilities in Microsoft Copilot Studio for HR Activities

Day 2

Architecting Success with Multi-Agent AI Systems

Module 3: Customize Agents with Gen AI in Copilot Studio

- Customizing your agents – Orchestrator, UI, Knowledge, Actions, Autonomy
- Copilot Studio implementation guidance for architects
- Generative AI in Copilot Studio
- Copilot Studio + Power Platform
- Building voice-enabled agents
- AI Foundry integration
- Developing agents using Microsoft 365 Agents SDK

Module 4: Innovate with Azure AI Platform

- How language models work
- AI Foundry and SDK introduction
- AI Foundry Model Catalog
- Azure AI Services
- Azure OpenAI Service and model guidance
- Models-as-a-Service
- Azure AI Foundry Agent Service
- Safeguard with Trustworthy AI

Hands-on Labs

- Setup AI Project and perform Chat Completion from VS Code
- Build a simple AI Agent
- Develop a multi-agent system

Day 3

Multi-Agent AI: Advanced Agent Dev in Azure AI Foundry

Module 5: Customize, orchestrate and experiment with Azure AI Foundry

- Retrieval Augmented Generation (RAG)
- Customizing models – Fine tuning, distillation
- Responses API (preview)
- Azure AI Foundry Agent Service - Orchestrate and debug AI workflows

Module 6: Build your own multi agents with Semantic Kernel or AutoGen

- Multi-agent applications
- Understanding Semantic Kernel
- Understanding AutoGen Agents Framework
- Multi-Agent Collaboration & Orchestration with AutoGen / Semantic Kernel

Hands-on Labs

- Set Up Azure AI Foundry SDK and Provision Resources
- Build a Retrieval-Augmented Generation(RAG) Pipeline
- Evaluate and Optimize RAG Performance
- Semantic Kernel Fundamentals
- Semantic Kernel Plugins

Day 4

Enterprise Grade: Optimization and production at scale

Module 7: Enterprise grade production at scale

- Scaling challenges and agent controls
- Manage AI performance in production
- Observability Tools
- Enabling Enterprise governance and management
- Enterprise grade security and data protection
- Monitoring and observability

Module 8: Advanced AI risk evaluation and mitigation

- Identifying risks
- Azure AI Content Safety
- Evaluation and GenAIops
- Identity and access management
- Network Security for AI apps
- Continuous security for AI

Hands-on Labs

- Understanding the Lifecycle of Flow Development
- Building and Customizing Prompt Flows
- Evaluation Flow Setup
- Fine-Tuning Prompts for Optimal Performance
- Implementing Chat Flow and Tool Integration
- Ensuring Responsible AI Practices with Content Safety

Journey A – Deal-ready

Deal-ready assessment 

Journey B – Solution Design

Solution Architecture assessment 

Journey C – Project-ready

Capstone Project 

Building Intelligence: Advanced Agent Development in Azure AI Foundry

⌚ 4 days (5 hours/day)



Module 1: Innovate with Microsoft 365 Copilot and agents

- M365 Copilot and agents
- How Copilot works
- Semantic index for Copilot
- M365 Copilot Chat
- Agents Use cases
- New agents in Microsoft 365
- Unlock more value with SharePoint agents

Module 2: Extend Microsoft 365 Copilot with Agents

- Microsoft 365 Extensibility Planning and approach
- Declarative agents and agent tooling
- Explore Copilot Studio Agent Builder
- Build declarative agents with Microsoft 365 Agents Toolkit
- Build custom agents with Copilot Studio
- Autonomous agents overview
- Agent Governance - Overview
- Gen-AI decision guide – when to build, buy or extend

Hands-on Labs

⌚ 3 hours

- Explore Copilot Studio Agent Builder
- Build HR Assistant Agent with Copilot Studio
- Incorporate actions in HR Agent
- Enable Autonomous Capabilities in Microsoft Copilot Studio for HR Activities

Module 3: Customize Agents with Gen AI in Copilot Studio

- Customizing your agents – Orchestrator, UI, Knowledge, Actions, Autonomy
- Copilot Studio implementation guidance for architects
- Generative AI in Copilot Studio
- Copilot Studio + Power Platform
- Building voice-enabled agents
- AI Foundry integration
- Developing agents using Microsoft 365 Agents SDK

Module 4: Innovate with Azure AI Platform

- How language models work
- AI Foundry and SDK introduction
- AI Foundry Model Catalog
- Azure AI Services
- Azure OpenAI Service and model guidance
- Models-as-a-Service
- Azure AI Foundry Agent Service
- Safeguard with Trustworthy AI

Hands-on Labs

⌚ 3 hours

- Setup AI Project and perform Chat Completion from VS Code
- Build a simple AI Agent
- Develop a multi-agent system

Module 5: Customize, orchestrate and experiment with Azure AI Foundry

- Retrieval Augmented Generation (RAG)
- Customizing models – Fine tuning, distillation
- Responses API (preview)
- Azure AI Foundry Agent Service - Orchestrate and debug AI workflows

Module 6: Build your own multi agents with Semantic Kernel or AutoGen

- Multi-agent applications
- Understanding Semantic Kernel
- Understanding AutoGen Agents Framework
- Multi-Agent Collaboration & Orchestration with AutoGen / Semantic Kernel

Hands-on Labs

⌚ 3 hours

- Set Up Azure AI Foundry SDK and Provision Resources
- Build a Retrieval-Augmented Generation(RAG) Pipeline
- Evaluate and Optimize RAG Performance
- Semantic Kernel Fundamentals
- Semantic Kernel Plugins

Module 7: Enterprise grade production at scale

- Scaling challenges and agent controls
- Manage AI performance in production
- Observability Tools
- Enabling Enterprise governance and management
- Enterprise grade security and data protection
- Monitoring and observability

Module 8: Advanced AI risk evaluation and mitigation

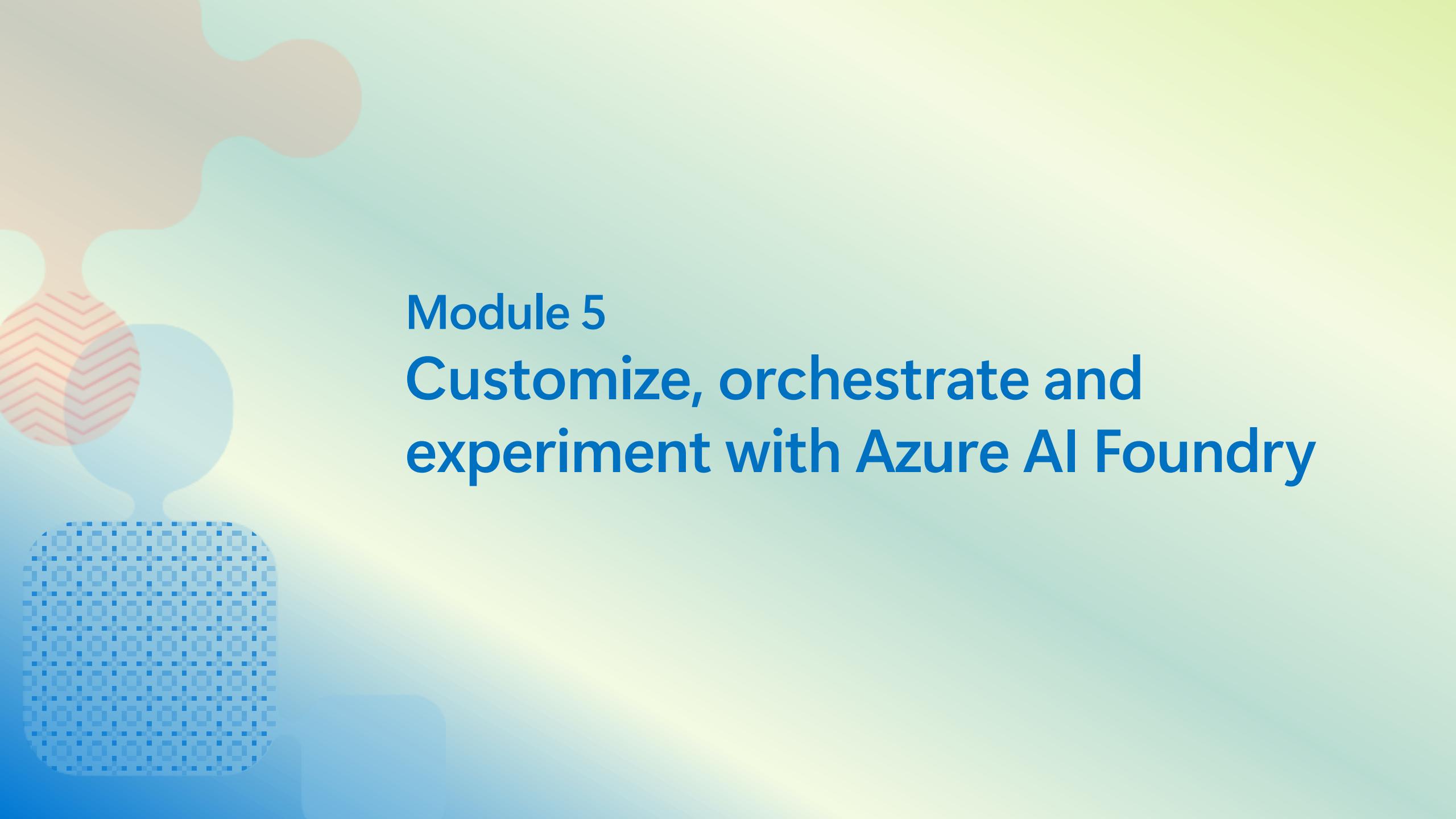
- Identifying risks
- Azure AI Content Safety
- Evaluation and GenAIOps
- Identity and access management
- Network Security for AI apps
- Continuous security for AI



Capstone Project

Objective: Apply your skills to design, develop, and deploy a sophisticated, end-to-end AI solution leveraging Azure AI services, Microsoft Copilot frameworks, and multi-agent orchestration techniques

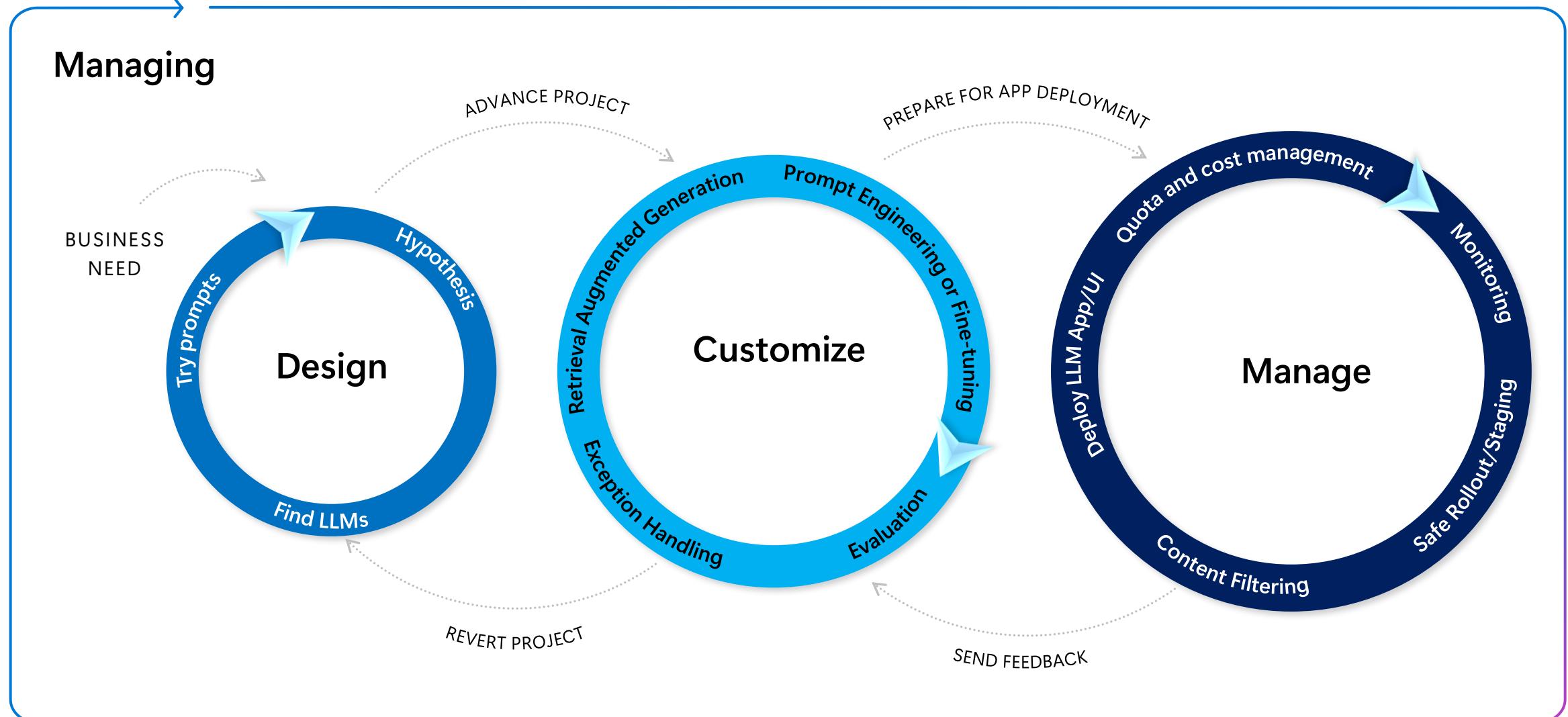
⌚ 3 hours



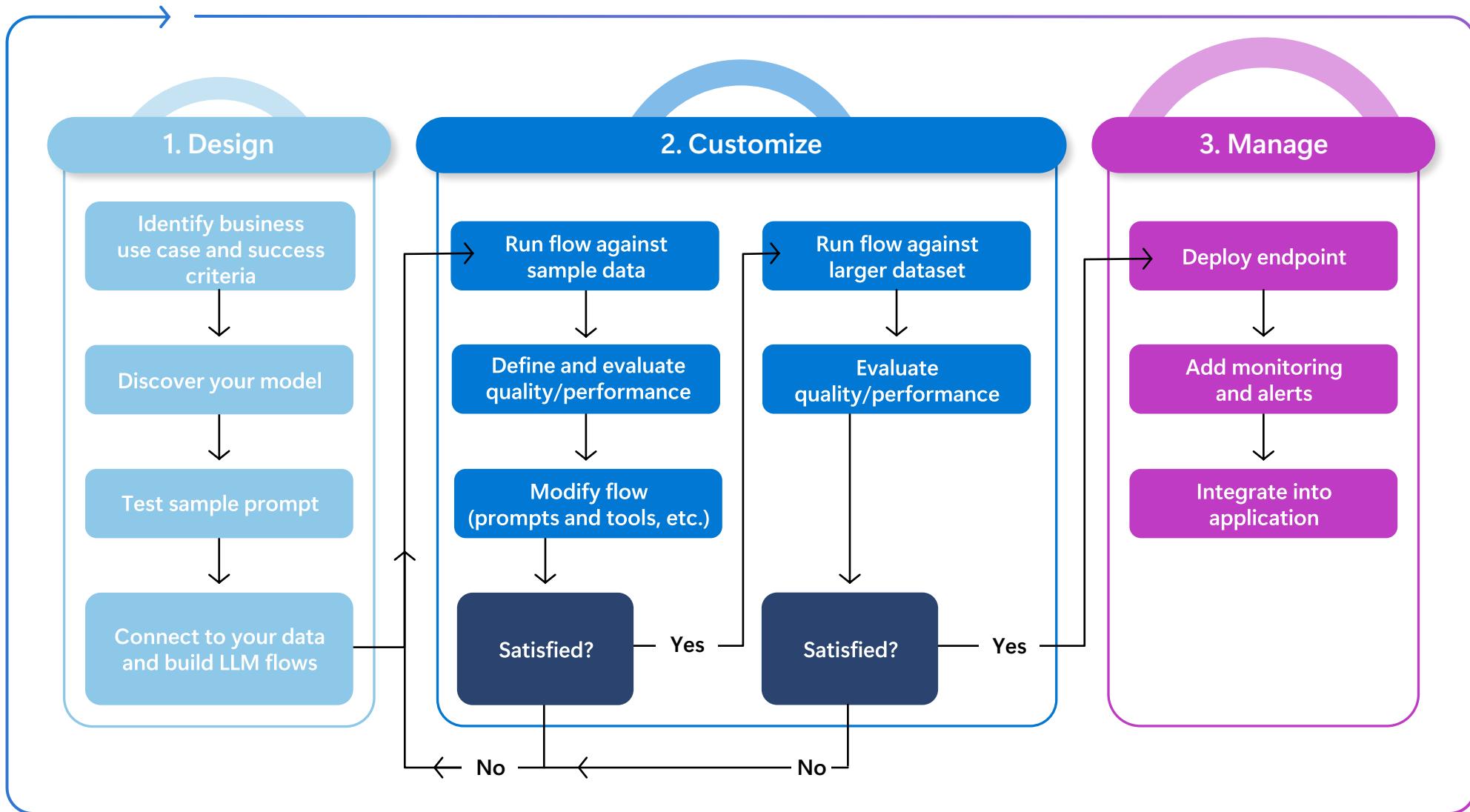
Module 5

Customize, orchestrate and experiment with Azure AI Foundry

AI Development Lifecycle



How do we know our results are any good?



Cutomizing your AI Applications

	RAG	Function/API	Agent
Scenario	One well-defined task (Q&A) Known outcomes (answers)	Multiple tasks Controlled outcomes (calls)	Complex task Open outcomes
Example	Looks for answers in product documentation	Query an API to validate account number	Chain multiple apis to solve a problem
Models	Question answering (understanding/summarization)	Intent detection and planning	Multiple
Orchestration	Systematic workflow	Flexible Resilient (Code is the trigger)	Dynamic Self-healing (Agent is the trigger)

Retrieval Augmented Generation



Common Design Patterns

RAG Agent



Code Generation
Agent



Multi-agent
Systems

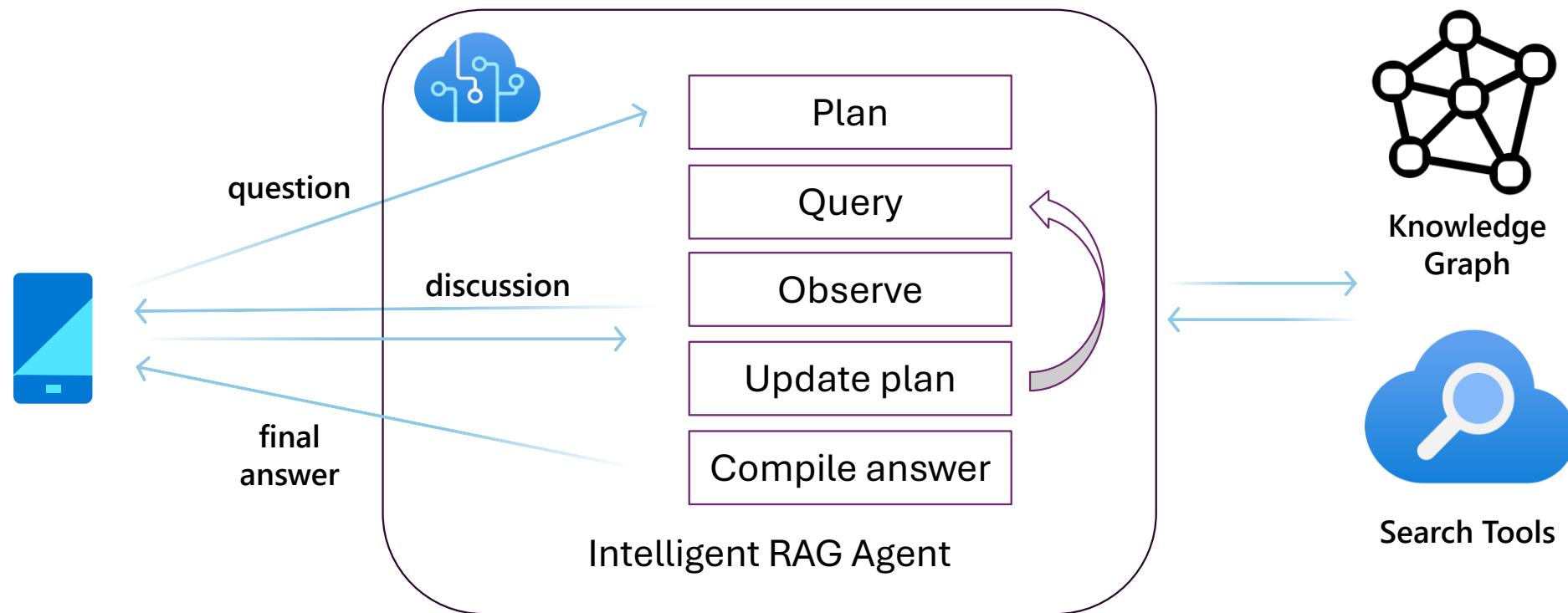


Multi-domain
Agent Systems



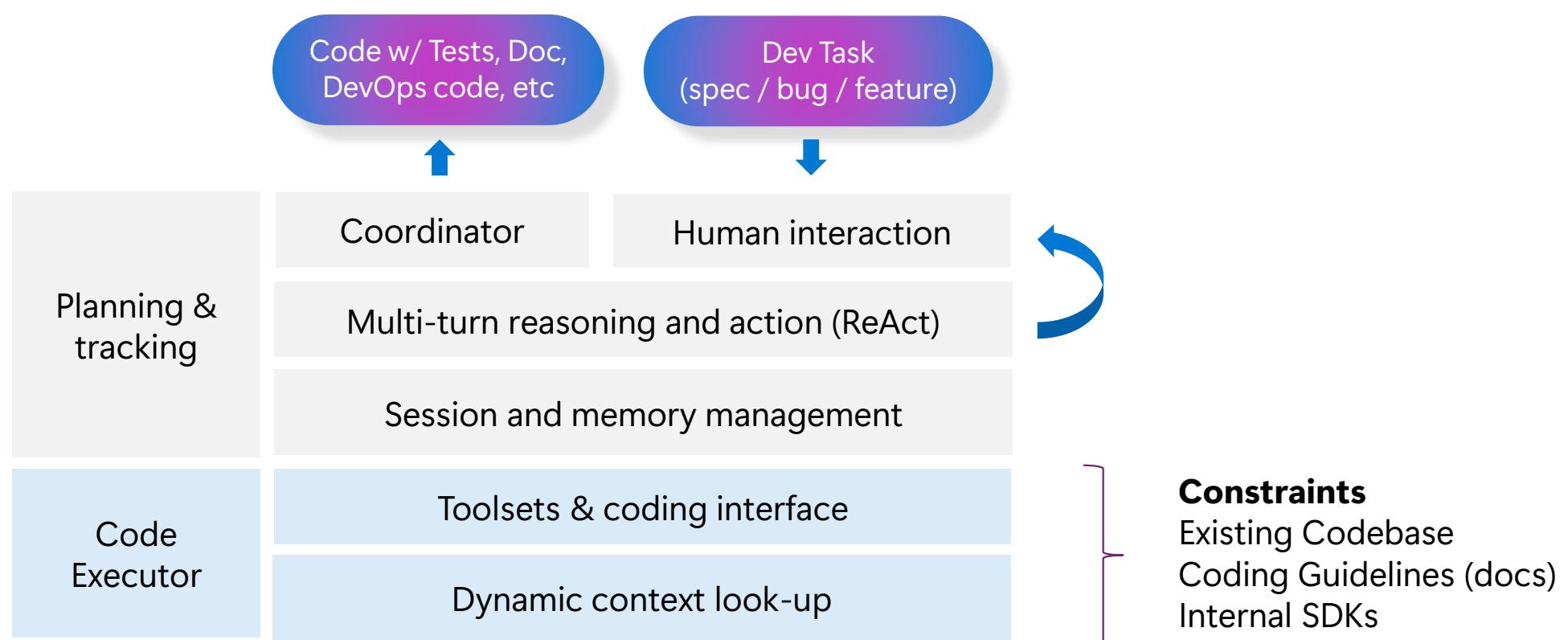
Retrieval Augmented Generation Agent

Translates questions into a research problem with human in the loop to produce high quality answers to complex questions within the scope of its domain



Code Generation Agent

Generates code based on natural language requirements, leveraging existing code base, templates, guidelines, libraries to match policies and best practices while interacting with humans to clarify, validate and deliver functionality as intended.

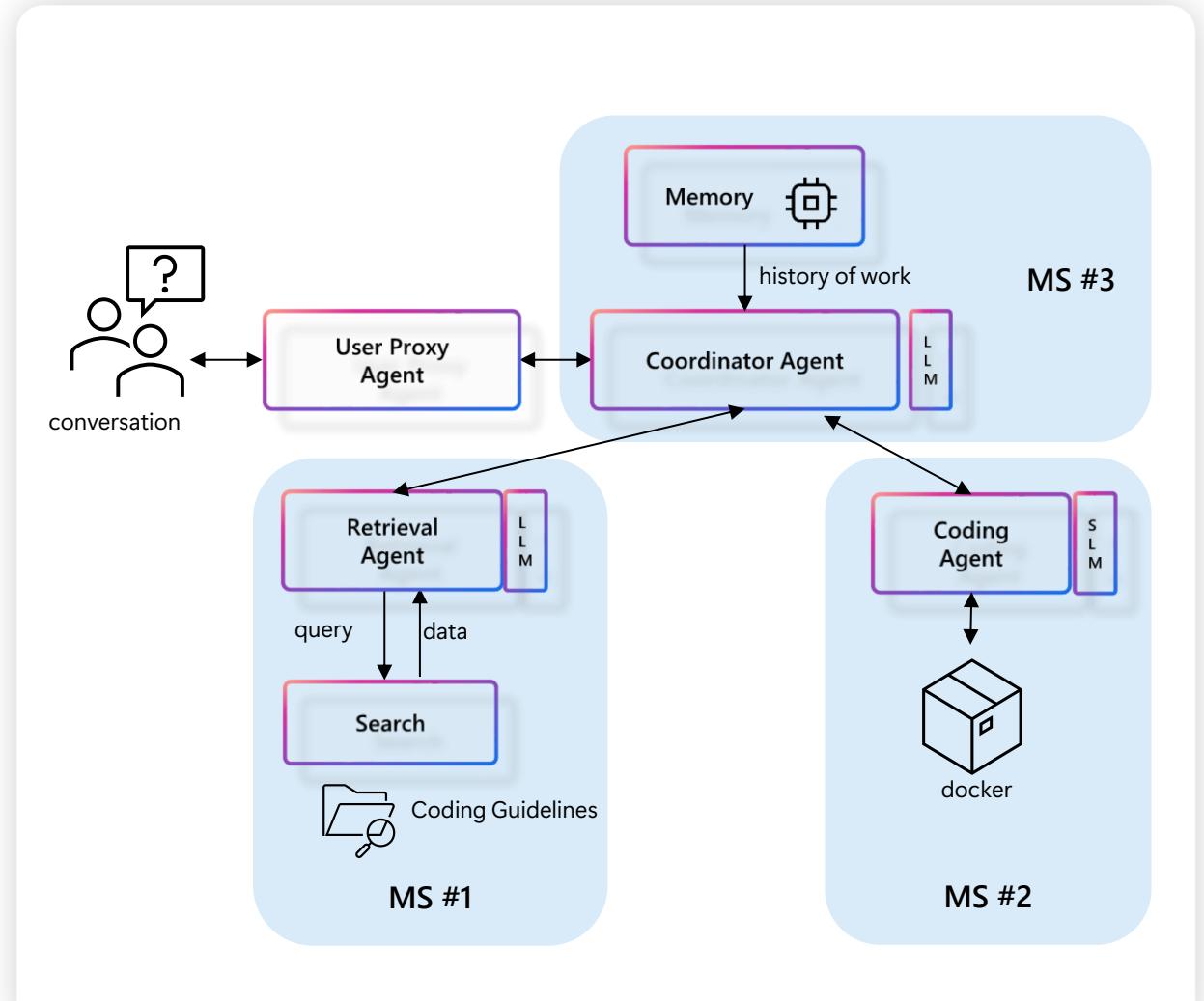


Multi-Agent System

A complex problem is decomposed into smaller, manageable parts, each addressed by specialized agents, effectively a micro-service (MS). These agents work together in a coordinated manner within a workflow to efficiently solve the overall problem.

Critical Design Elements

- ✓ Adaptive planning within scope of existing tightly scoped skills (agents)
- ✓ Handles ambiguity by discussing and refining requirements with human
- ✓ Memory to handle complex long running execution of a plan
- ✓ Effective inter agent communications
- ✓ Test, monitor, release & maintain each agent independently to quickly handle quality & safety issues

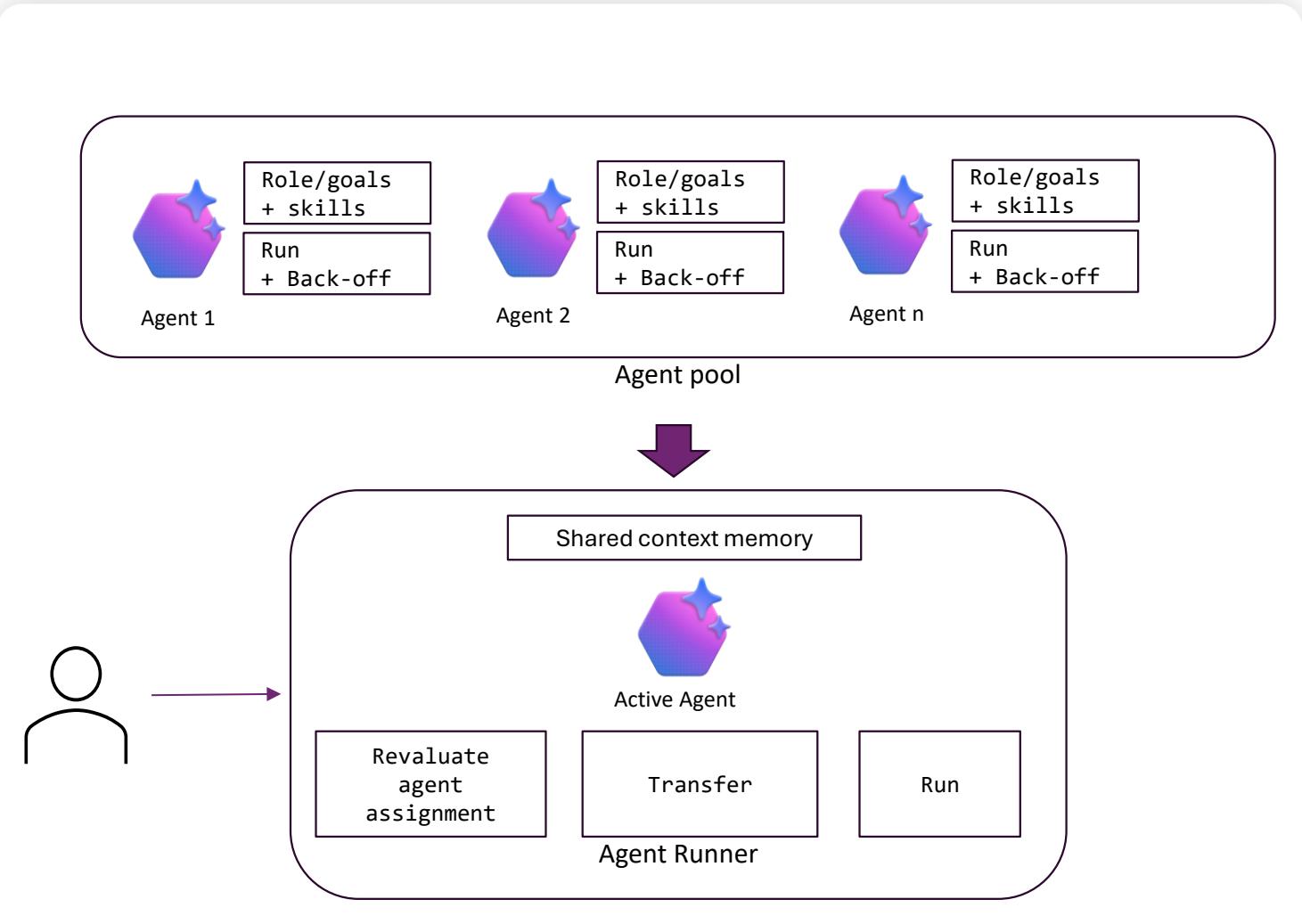


Multi-Domain Agents System

Multiple domain-specific agents are orchestrated by an Agent Runner to scale across multiple domains while appearing as a single agent to users.

Critical Design Elements

- ✓ Agents capability descriptors
- ✓ Scalable Agent Runner able to manage 10s to 100s of agents
- ✓ Ability to manage domain switching with proper memory management
- ✓ Avoid single interceptor problem as individual agents maintain direct communication with user and can hand off when needed



Azure AI Search

Azure AI Search is a battle-tested retrieval system purpose-built to deliver superior retrieval augmented generation (RAG). Use AI to collect, enrich and surface all aspects of your data.

- Ingest and search across all your data, no matter the type, volume or source
- Streamlined RAG pipeline, from import to chunking to embedding.
- Enterprise-ready foundation cost-optimized for performance and scale.
- Enable hybrid search (vector + keyword) results using RRF (Reciprocal Rank Fusion) and multi-vector retrieval for multi-modal search
- Improve retrieved results immediately with reranking and query rewriting.
- Tailor RAG experiences by utilizing customizable features such as granular access control and relevance tuning

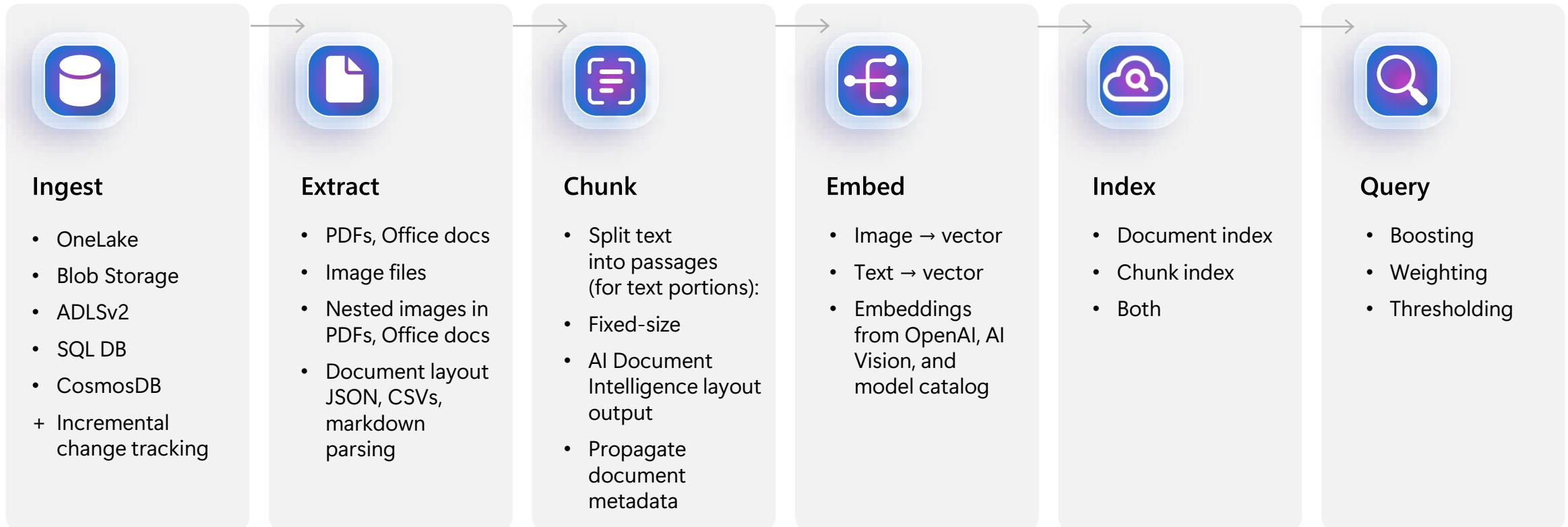
The screenshot shows a configuration interface for Azure AI Search. On the left, a vertical navigation bar lists four steps: 'Data source' (checked), 'Data field mapping' (checked), 'Data management' (selected), and 'Review and finish'. The main area is titled 'Data management' with the sub-instruction 'Set up specific configurations for your data and how the model will respond to requests.' Below this is a link 'Learn more about data privacy and security in Azure AI.' The next section, 'Search type' (marked with a red asterisk), has a dropdown set to 'Hybrid + semantic'. Below it is a dropdown for 'Select an existing semantic search configuration' with 'default' selected. A note states 'Using semantic search will incur usage to your Azure AI Search account. View Pricing.' At the bottom, a purple bar contains the text 'Image may not reflect actual user interface.'



Easily enable search best practices through the UI: like being able to select hybrid search + reranking in one click.

Streamlined RAG data pipeline

End to end, integrated data prep and processing



Retrieval Augmented Generation

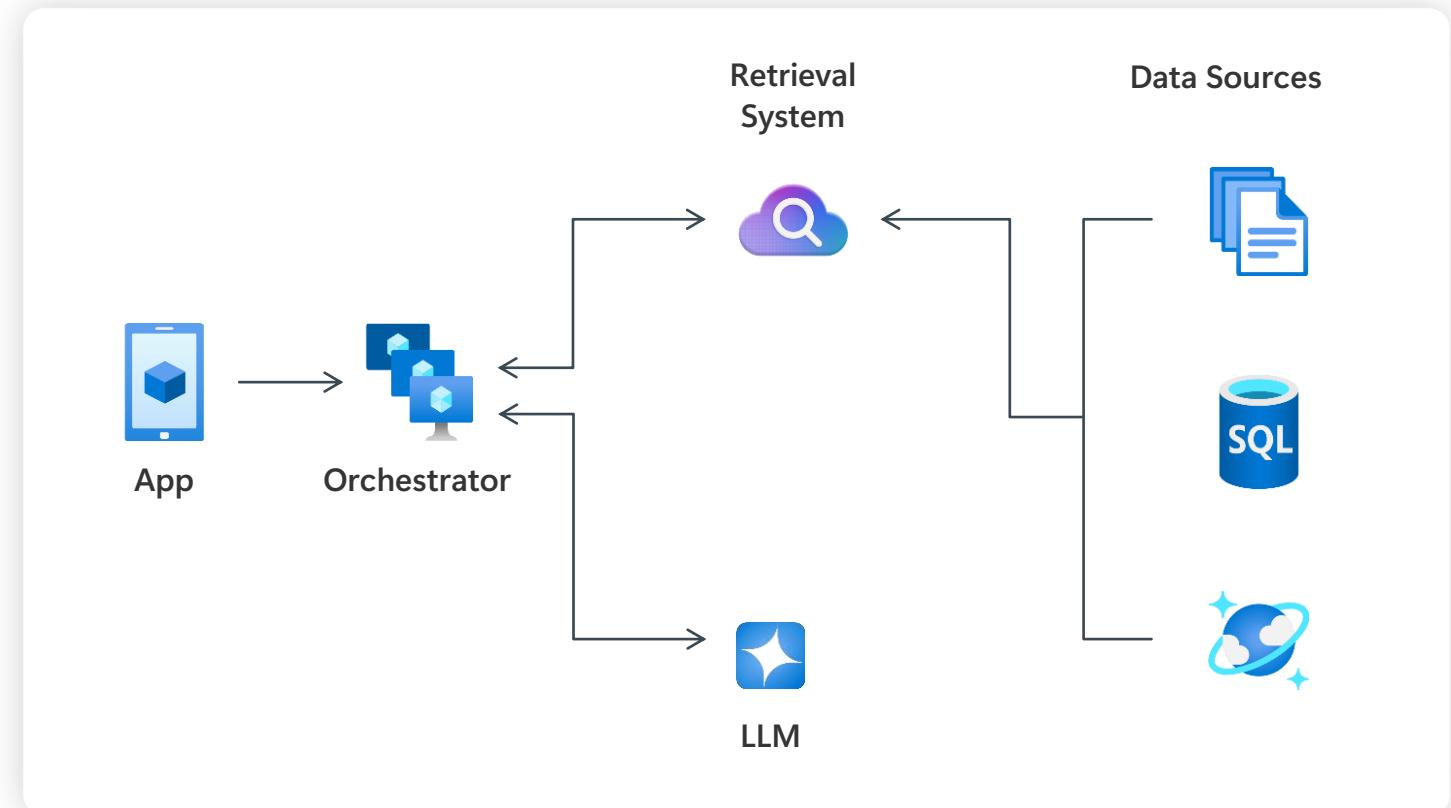
Combine reasoning with knowledge

Elements

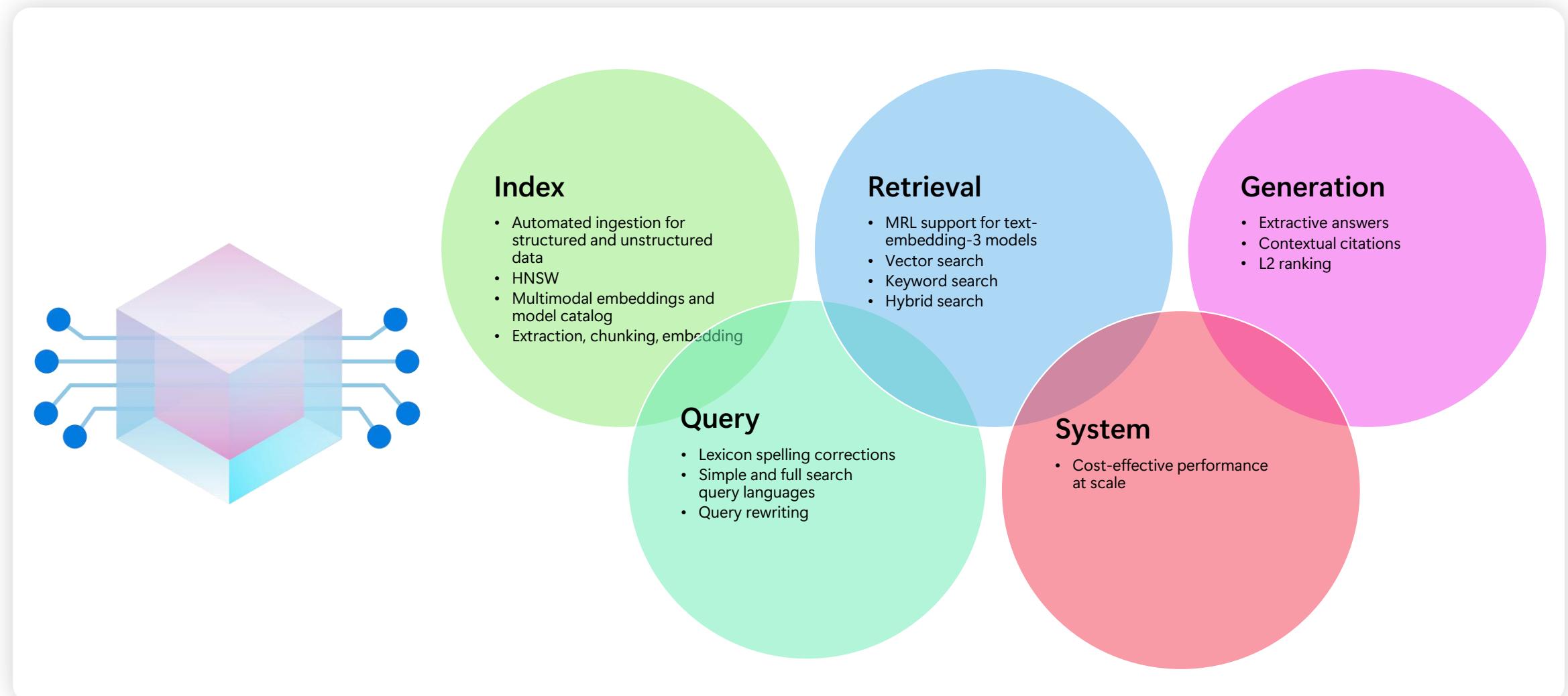
- Retrieval system collects externalized knowledge
- Language model provides reasoning to provided knowledge and prompt

Techniques retrieval system brings

- Querying: spelling corrections, multi-lingual query support
- Retrieval: keyword, vector, hybrid search
- Augmentation: L2 ranking,
- Generation: Extractive answers, citations
- Indexing: chunking, embedding, automated data import



Azure AI Search features across the RAG pipeline



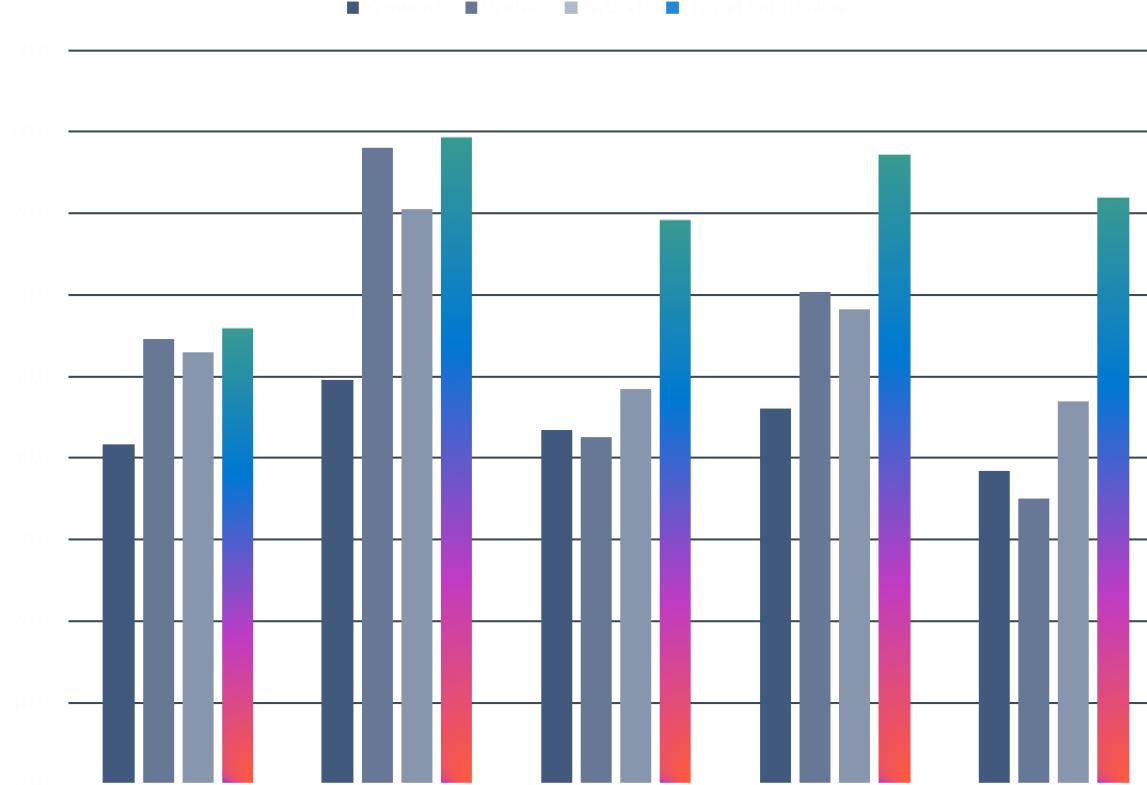
Integrate structured & unstructured data



- Azure AI Search
- Blob storage
- Local files/folders
- Storage URLs including OneLake in Microsoft Fabric
 - Azure Databricks
 - S3 buckets via Amazon S3 Shortcuts

The screenshot shows the 'Add your data' wizard in the Azure AI Foundry interface. The left sidebar lists 'Playgrounds' as the selected category. The main panel displays a four-step process: 1. Source data, 2. Index settings, 3. Search settings, and 4. Review and finish. Step 1 is currently active. A callout box highlights the 'Data source' dropdown, which is set to 'An Azure AI Service resource'. Below it, a list of options includes 'Azure AI Search', 'Data in Azure AI Studio', 'Azure Blob Storage', 'Storage URL', and 'Upload files'. A cursor points at the 'Next' button at the bottom of the wizard.

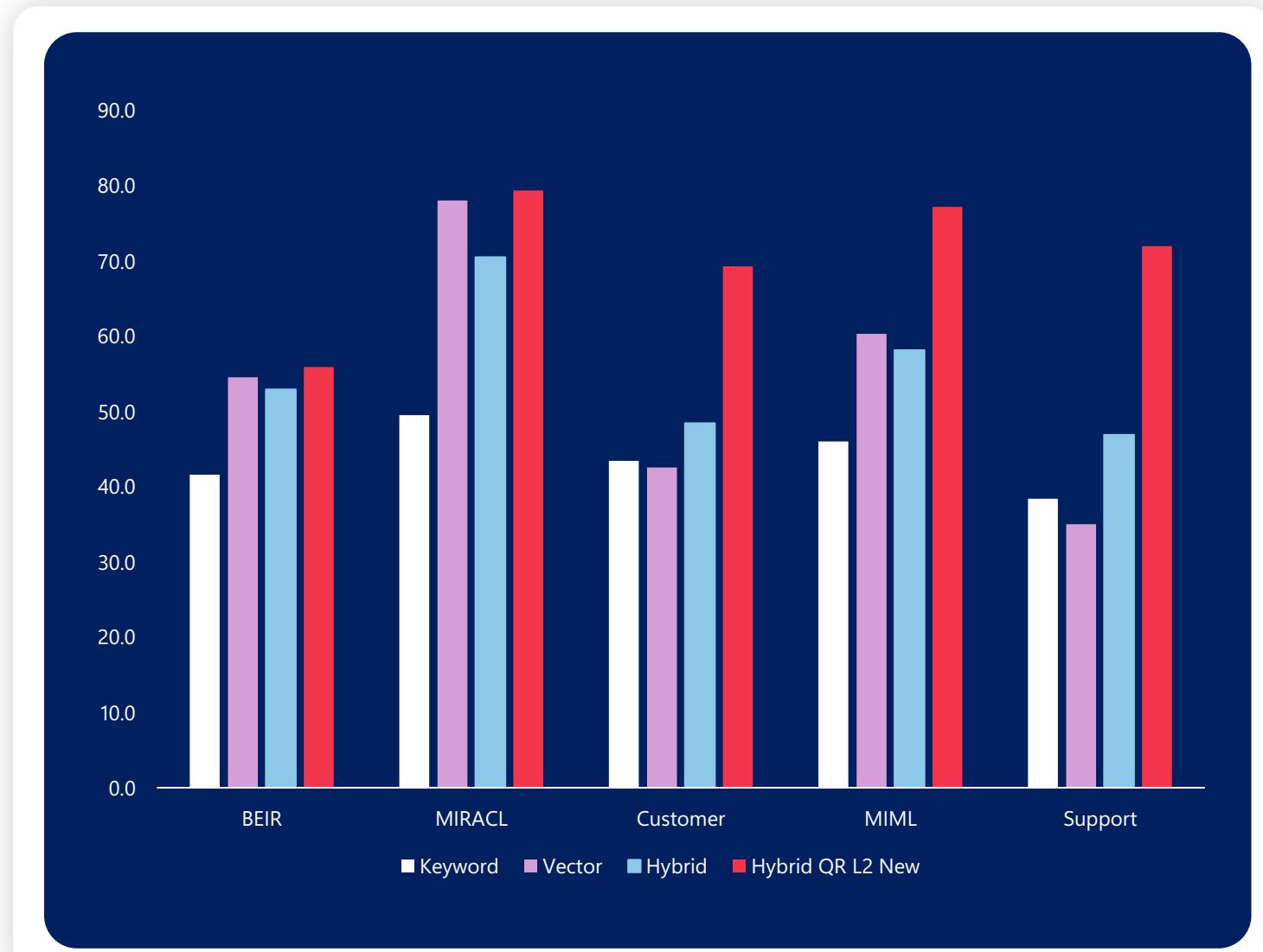
Retrieve using semantic similarity and hybrid search



Find the most relevant information in large datasets using hybrid search with keywords, vectors, and semantic ranker

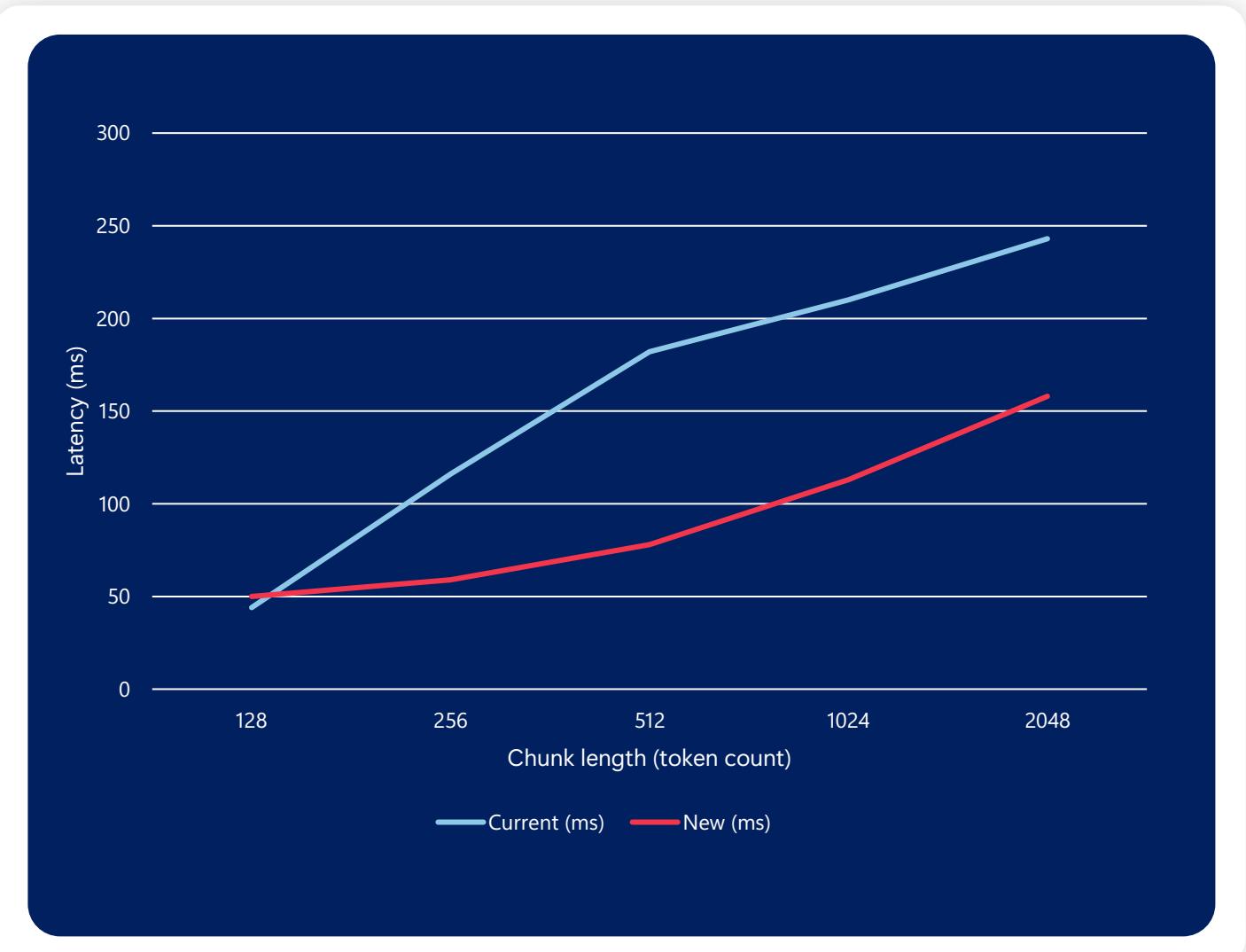
RAG techniques measured for relevance

- New L2 ranker model, query rewriting and hybrid perform best
- Up to *double* the baseline quality for some datasets
- Particularly strong in real-world RAG app datasets

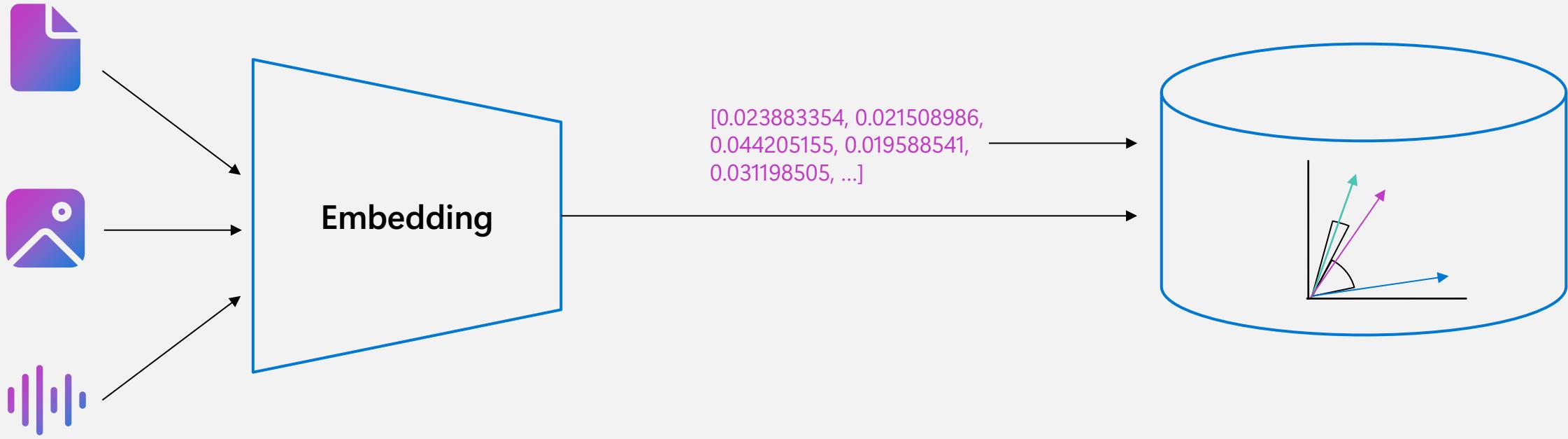


Better relevance, faster responses

- New features had no impact on latency
- No performance compromise
- Similar latency for small chunk sizes, up to twice as fast for larger, more common token counts



Retrieve using semantic similarity and hybrid search



Find the most relevant information in large datasets using hybrid search with keywords, vectors, and semantic ranker

Use embeddings from the Azure AI Foundry model catalog

Catalog includes embedding models

Deploy in Azure AI Foundry, connect to Azure AI Search for index/query-time vectorization

Includes models that use narrow types

Example:

Cohere's RAG-focused model "Cohere-embed-v3-English", supports float, int8, binary

The screenshot shows the Azure AI Foundry Model Catalog interface. At the top, there's a banner with the text "Find the right model to build your custom AI solution". Below it, a section titled "What's new?" features five cards: "tsuzumi is now available!", "Announcing BRIA 3.2 Fast", "News from Cohere!", "New SLM from Mistral", and "Meta Llama 3.2 models are here!". Each card has "Check out models" and "Read blog" buttons. Below this, a section titled "New model benchmarks available now in model catalog" provides information about comparing models across various metrics like Quality, Safety, and Cost. The main content area displays a grid of 1815 models, each with a thumbnail, name, and description. Some models listed include "o1-preview", "gpt-4-realtime-preview", "gpt-4", "gpt-4-mini", "tsuzumi-7b", "Bria-2.3-Fast", "Minstral-3B", "Cohere-embed-v3-english", "Llama-3.2-11B-Vision-Ins...", "Phi-3.5-MoE-instruct", "Phi-3-mini-128k-instruct", "Phi-3-mini-4k-instruct", "Phi-3-small-8k-instruct", "Phi-3-medium-128k-instr...", and "jais-30b-chat". A navigation bar at the bottom indicates "Models 1815". A note at the bottom states "Image may not reflect actual user interface."



RAG for GitHub Models

Free. Frictionless. Just add data.

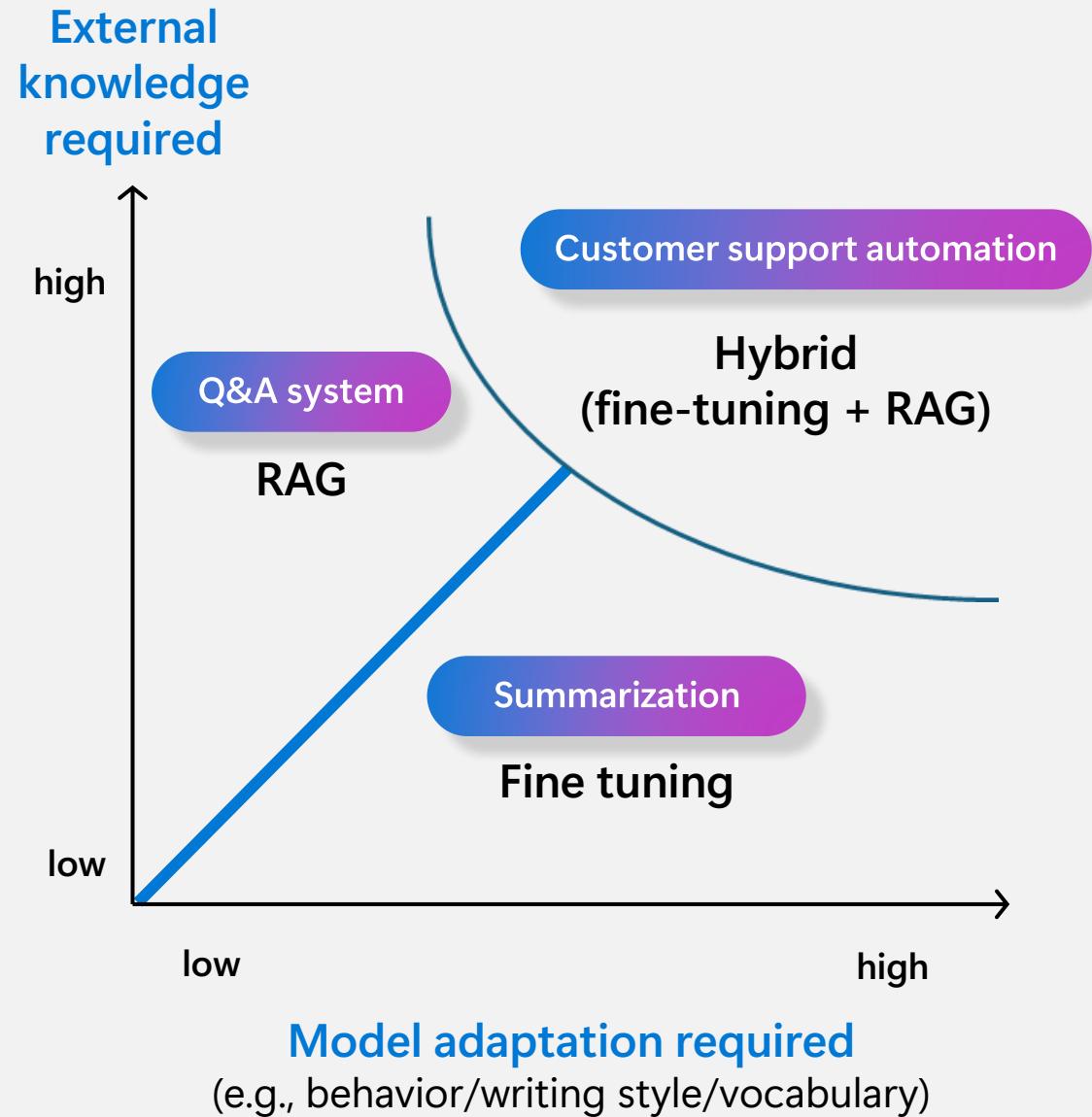
Seamless playground
experience

Auto-provisioned AI
Search index

Full-featured retrieval system

Advanced RAG directly
from code

When to use RAG vs. fine-tuning?



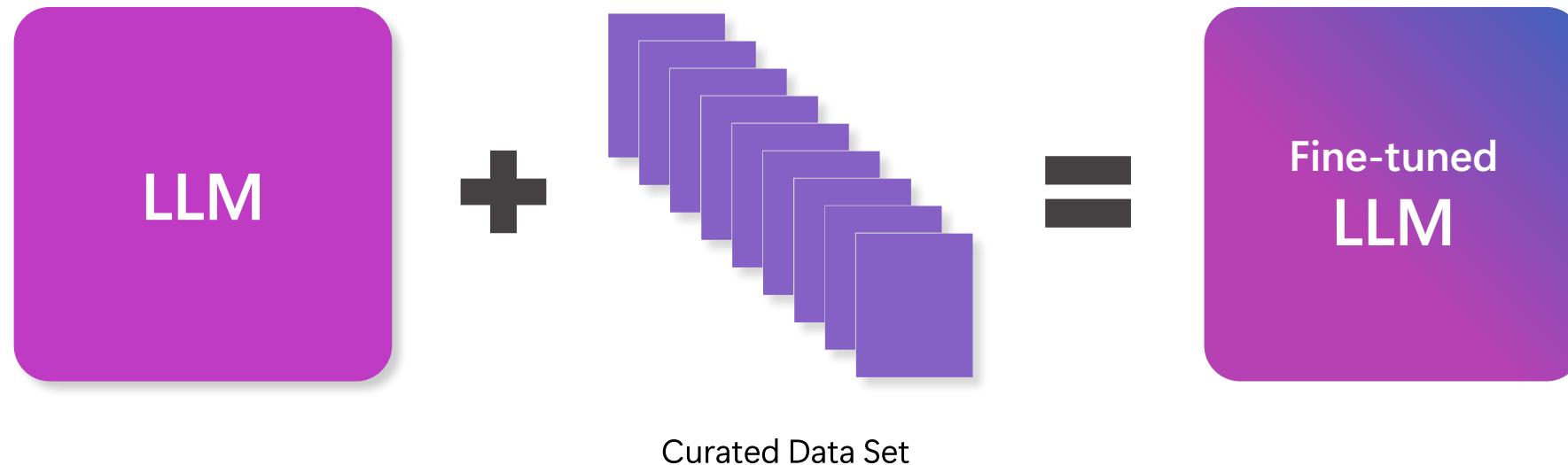
Fine-tuning and Distillation



What is fine-tuning?

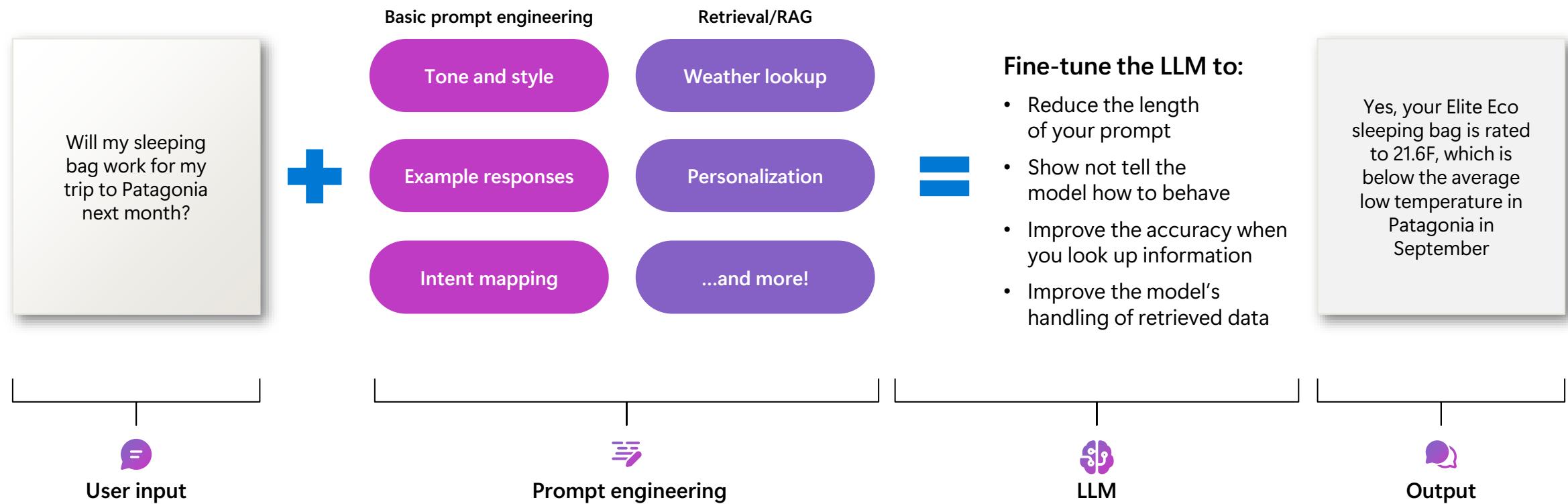
Customize your models

Customizing a pre-trained LLM with additional training on a specific task or new dataset for enhanced performance, new skills, or improved accuracy



Where does RAG and fine-tuning fit in?

LLMs are language calculators



Azure AI Foundry fine-tuning

Fine-tune and customize AOA models, Microsoft Phi models, and 3P models from Meta, Mistral, etc. to meet specific business needs.

Supervised fine-tuning with LoRA to efficiently adapt off the shelf models without any loss of accuracy.

Supported models:

- Babbage-002 (Completion)
- Davinci-002 (Completion)
- GPT-35-Turbo (Chat) – 0613, 1106, 0125
- GPT-4
- GPT-4o-mini (0718)
- GPT-4o (0806)
- Llama 2 and Llama 3
- Llama 405 B
- Mistral Large

The screenshot shows the Azure AI Foundry interface with a sidebar on the left containing links like Overview, Model catalog, Playgrounds, AI Services, Build and customize, Code, Fine-tuning (selected), Prompt flow, Assess and improve, Tracing, Evaluation, Safety + security, My assets, Models + endpoints, Data + indexes, and Web apps. The main area is titled "Fine-tune a model by training it on your own data" and "Select a model to fine-tune". It lists 47 models, with "gpt-4o-mini" selected. The right side provides details about "gpt-4o-mini", noting it's a Chat completion model. It highlights its low cost and latency, ability to handle large volumes of context, and support for text, image, video, and audio inputs. A note mentions it surpasses GPT-3.5 Turbo and other small models on academic benchmarks. At the bottom are "Next" and "Cancel" buttons.

Azure OpenAI Service

Fine-tuning methods

Supervised fine-tuning

- Davinci-002 - Completion
- Babbage-002 - Completion
- GPT-35-Turbo (0613, 1106, 0125) – Up to 16K context – Chat
- GPT-4 (0613) – 8K context – Chat
- GPT-4o (0513) – 128K context – Chat
- GPT-4o-2024-0806
- GPT-4o-mini (0718) – 64k context - Chat

Direct preference optimization

- GPT-4o (0513) – 128K context – Chat
- GPT-4o-2024-0806

Reinforcement learning

- o1-mini

Customize your models

Azure OpenAI Service Distillation

Use the outputs of large, complex models to fine-tune smaller more efficient models

Stored completions

Custom evaluations

Fine-tuning

Customize your models

Azure OpenAI Service

Integration with Weights & Biases

Powerful capabilities to evaluate, monitor, and iterate fine-tuned models

Responses API (Preview)



Responses API

Customize your models

- Combines simplicity of Chat Completions API with tool use capabilities of Assistants API
- Seamless interaction with tools like CUA, code interpreter, function calling, and file search – all in a single API call
- Allows AI systems to retrieve data, process information, and take actions
- Enables AI systems to easily integrate with enterprise workflows

Simplifies AI Agent Development

Orchestrates Multiple AI Tools

Enhances Scalability for Enterprise Use Cases

Offers Flexibility with Azure AI Foundry Agent Service

* As this is a preview version, it is designed for testing and feedback purposes and is not yet optimized for production traffic.

How to compare Responses API and Azure AI Foundry Agent Service?

Responses API

Simple, out-of-the-box solution to **enable multiple tools in a single API**

Azure AI Agents Service

Managed, enterprise-grade agent platform where customers can use built-in connectors with Microsoft ecosystem and enterprise guarantees

Computer Use Agent (CUA)

Customize your models

- Combines Vision and Reasoning
- Easily navigates buttons, menus, and text fields
- Smart task execution with adaptation and error handling
- Top performance across benchmarks – achieves:
 - 38.1% on OSWorld
 - 58.1% on WebArena
 - 87% on WebVoyager

Multimodal
reasoning

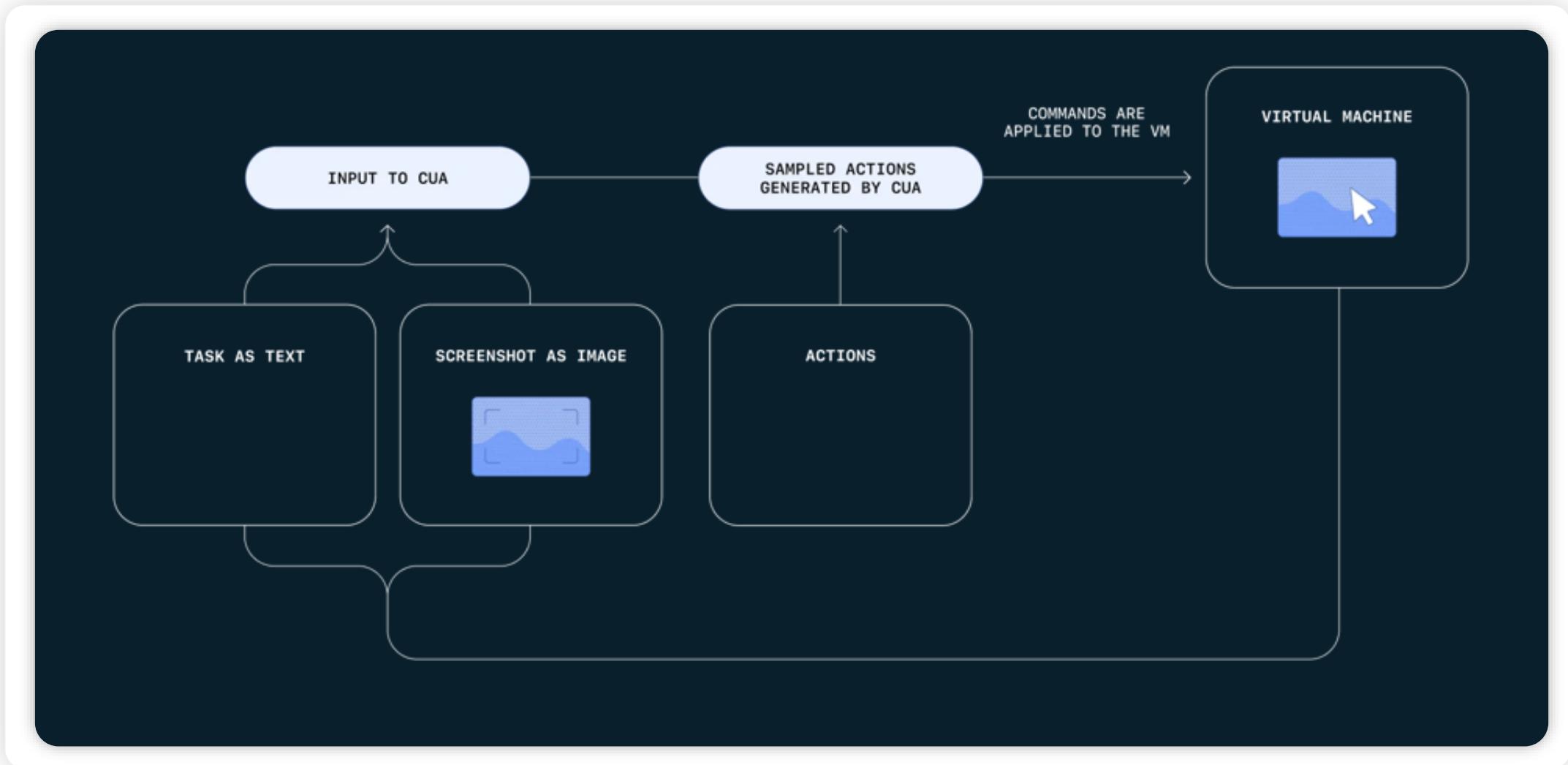
Seamless GUI
Interaction

Smart Task Execution

Top Performance
Across Benchmarks

* As this is a preview version, it is designed for testing and feedback purposes and is not yet optimized for production traffic.

How does CUA work?



Chat vs. Agent frameworks

Orchestrate and debug workflows

Chat

Lightweight and powerful

Inherently stateless

Agents

Stateful (inbuilt conversation state management)

Access persistent threads

Access files in several formats. API handles chunking, embeddings, storage and creation, and implementing vector search

Automatic management of the model's context window

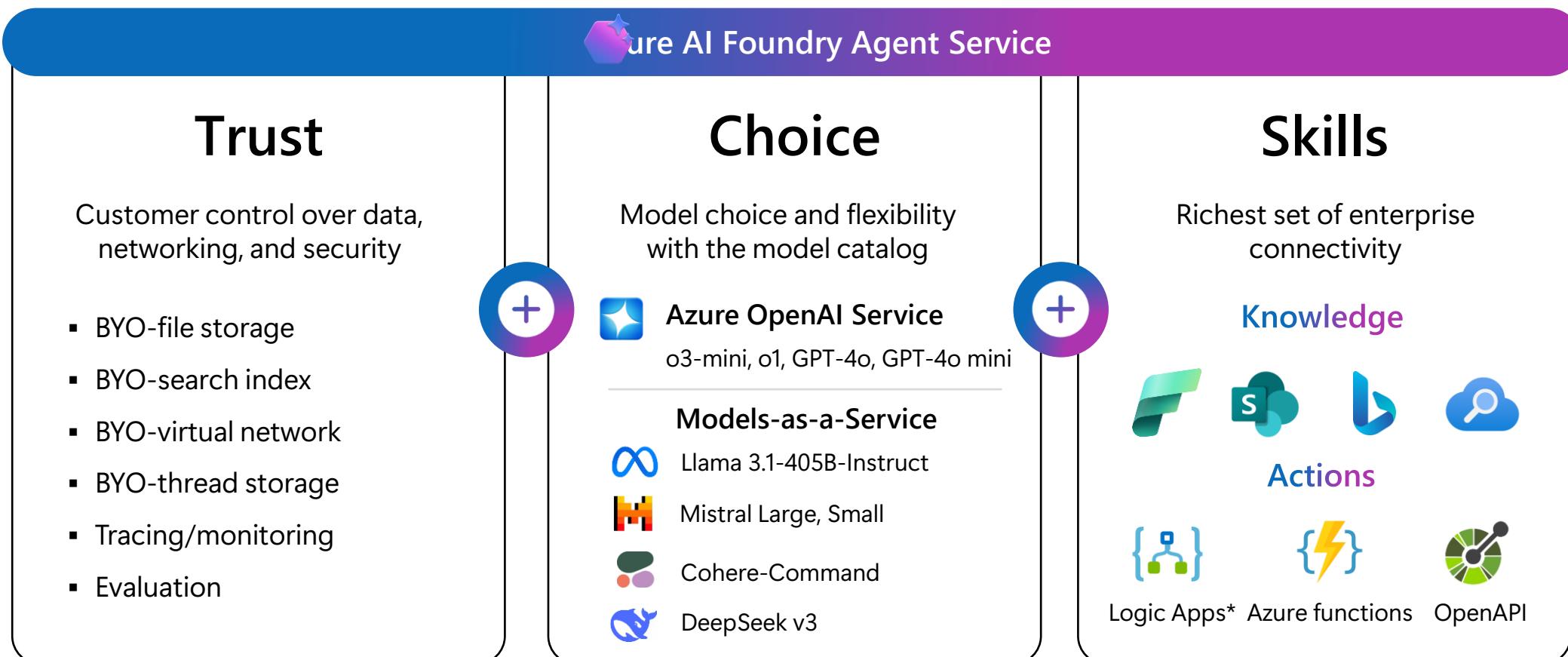
Access multiple tools in parallel (up to 128 tools per assistant) including code interpreter

Build your own function calling

Building multi-agents with Azure AI Foundry Agent Service



The full enterprise package

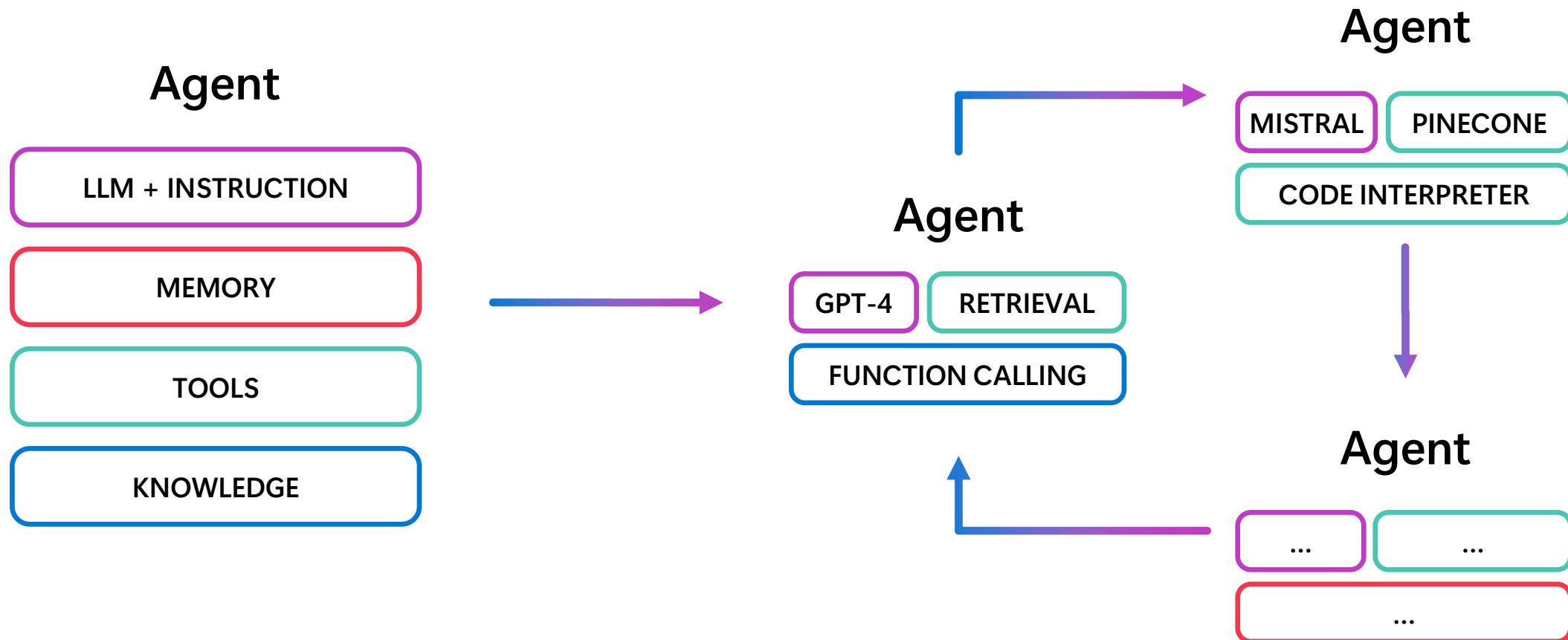


Azure AI Foundry portal

Azure AI Foundry SDK

A single and multi-agent world

Orchestrate and debug workflows



1

Single-agent

Deploy agents with
Azure AI Foundry



Managed agent
micro-services

2

Multi-agent

Orchestrate them together with
AutoGen and **Semantic Kernel**



State-of-the-art
research SDK



Production-ready
and stable SDK

Ideation

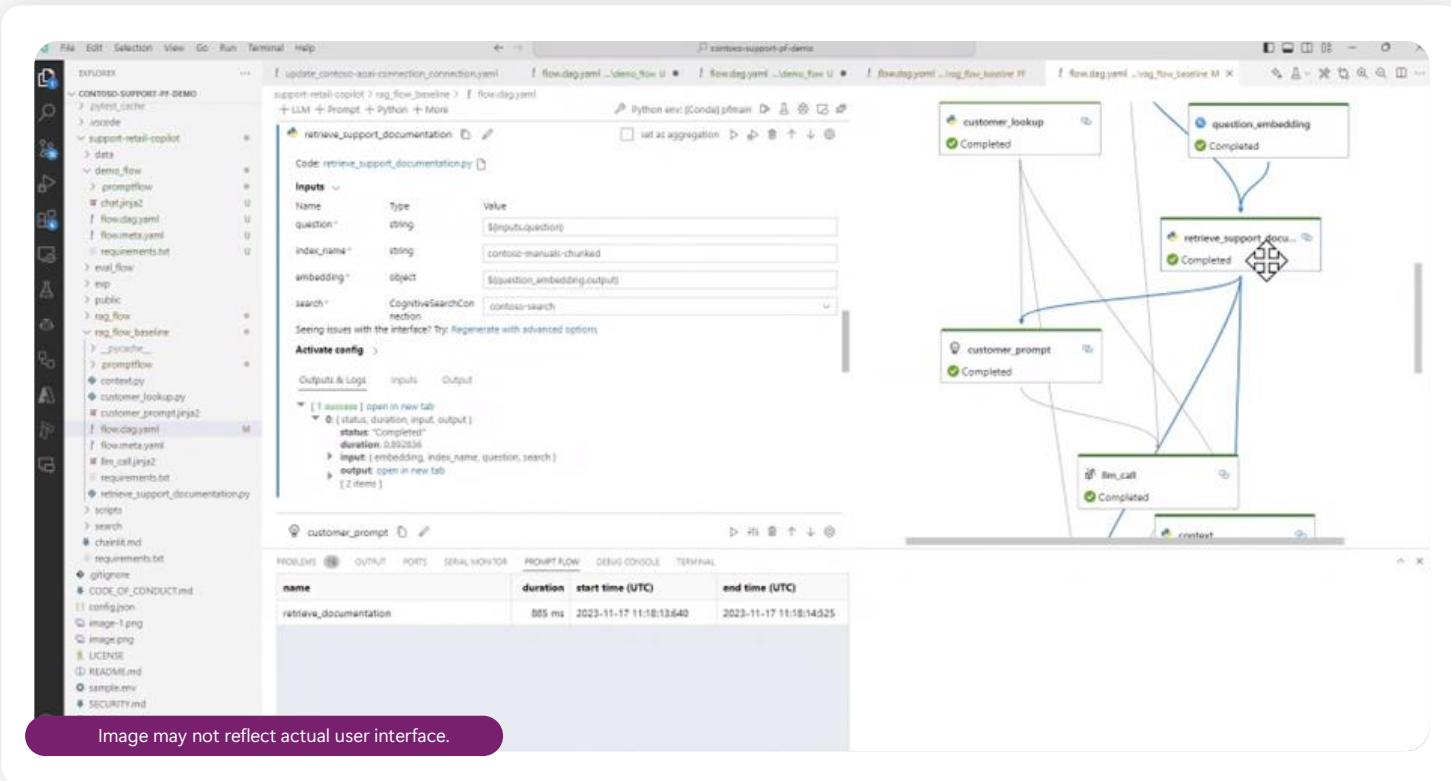
Production

Extended tools and frameworks

Orchestrate and debug workflows

Prompt flow offers a developer-friendly and easy-to-use code first experiences through SDK, CLI, and VS code extension, allowing seamless integration with open-source frameworks and CI/CD pipelines for automation.

- Use your favorite frameworks and editors that allow you to work in your preferred code environments, whether simplified UI or Visual Studio Code/IDE
- Initially build flows with open-source frameworks like LangChain or Semantic Kernel, and use prompt flow to scale the experiments
- Define flows in YAML format, which can stay aligned with the references source files in a folder structure for flow versioning in code repository
- Easily integrate with existing CI/CD pipelines through GitHub Actions and Azure DevOps for GenAIOps
- Smoothly transition from local to cloud by exporting flow folders to local, or uploading folders to the cloud for further authoring, testing, and deployment



Streamlines the development lifecycle of generative AI app with version-controlling, CI/CD integration, and enhanced collaboration among team members.

AutoGen compatibility

Orchestrate and debug workflows

- Multi-agent conversation framework to interactively build LLM workflows
- Use 'GPTAssistantAgent' to create specialized agents that can collaborate on tasks

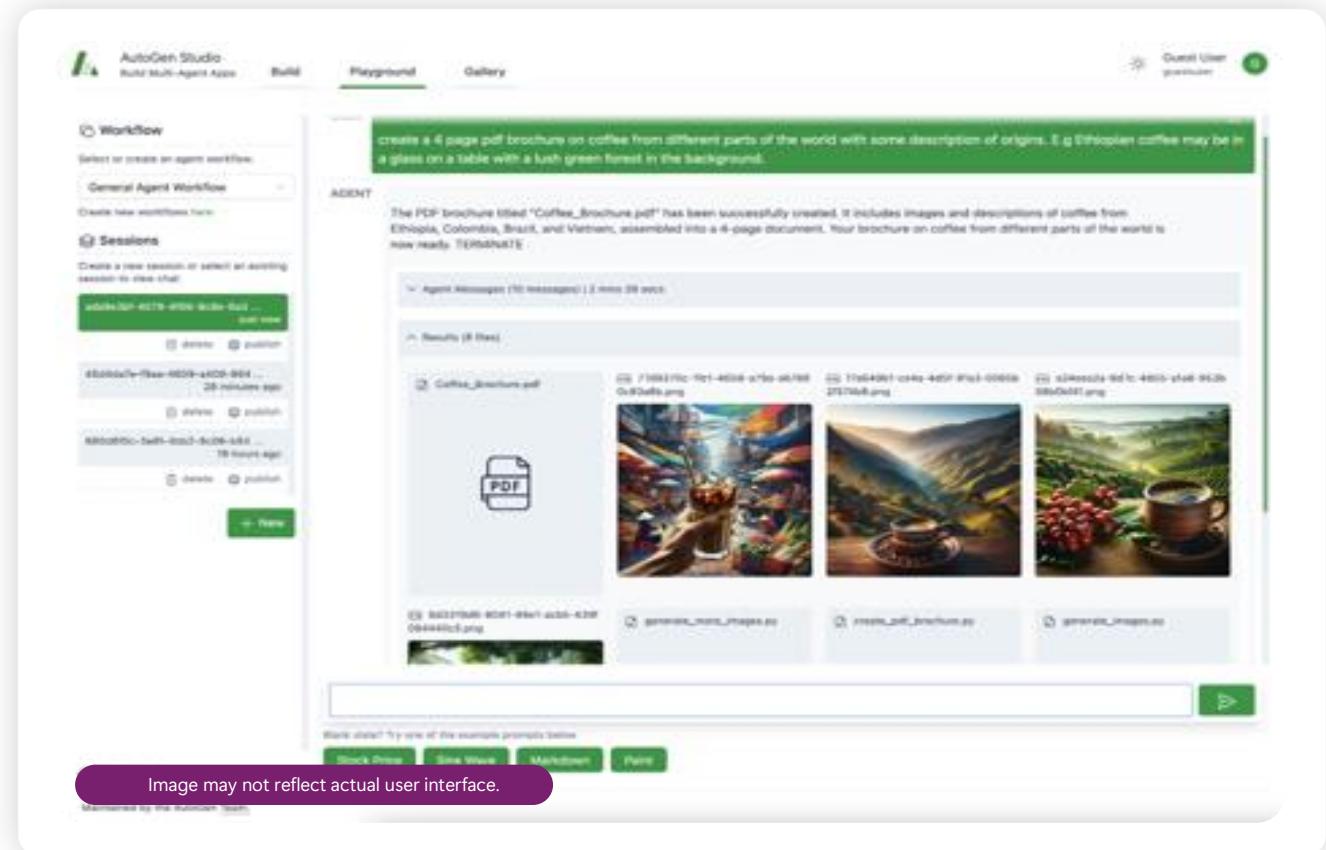


Image may not reflect actual user interface.

Agent Sample #1

Web Search and Extraction

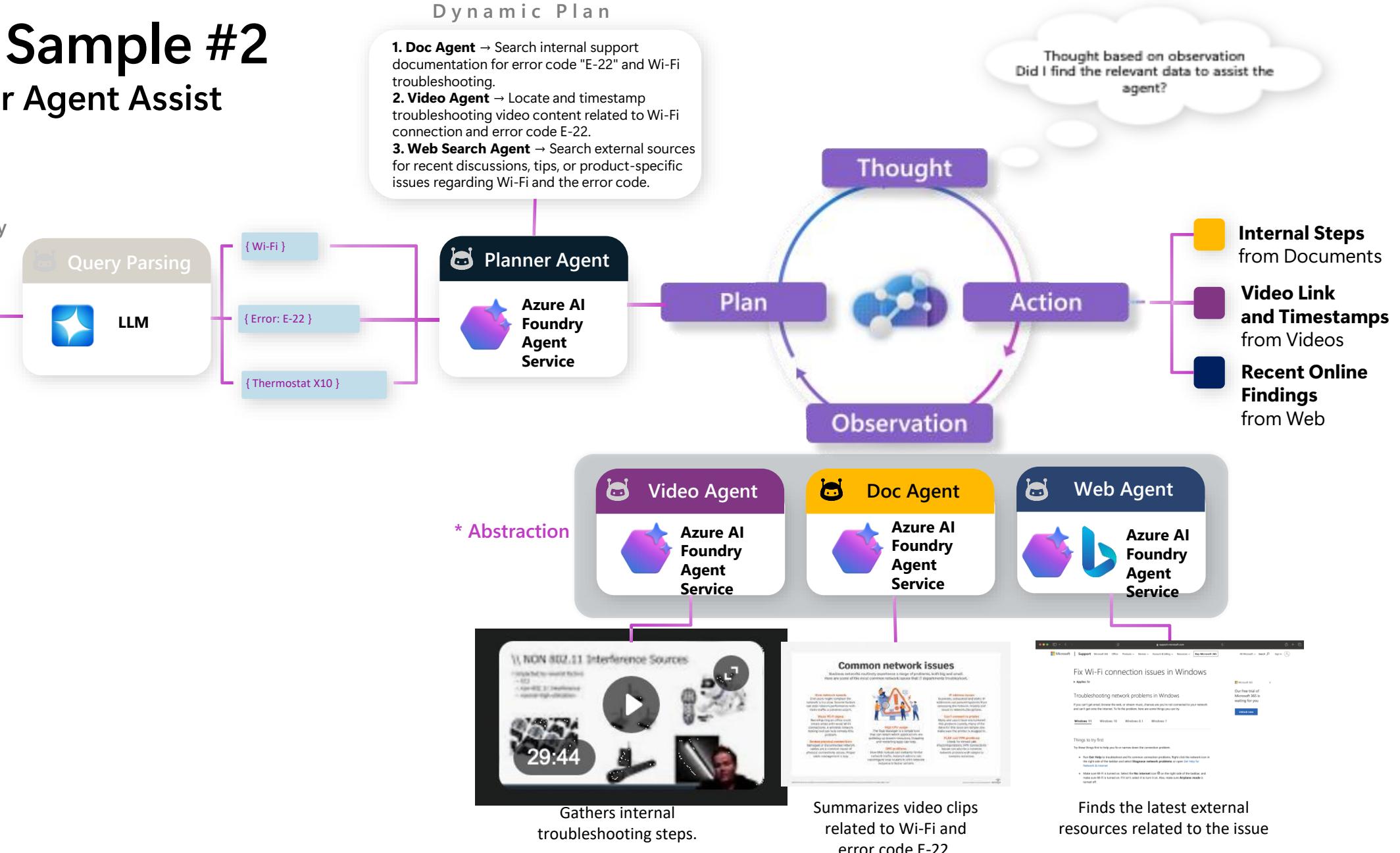


Agent Sample #2

Call Center Agent Assist

Customer Query

My SmartHome thermostat isn't connecting to Wi-Fi, and I keep seeing an error code E-22. How can I fix this?



* We can use the Azure AI Content Understanding service to extract images and videos, and transform them from unstructured to structured data



Module 6

Build your own multi agents with

Semantic Kernel or AutoGen

Agent and Multi-Agent Applications

Agent

- A software entity that communicates via messages, maintains its own state, and performs actions in response to received messages or changes in its state
- These actions may modify the agent's state and produce external effects, such as updating message logs, sending new messages, executing code, or making API calls.

Multi-agent applications

- Software systems, composed of multiple interacting agents that can be modeled as a collection of independent agents that interact with one another.
- Examples include:
 - Sensors on a factory floor
 - Distributed services powering web applications
 - Business workflows involving multiple stakeholders
 - AI agents, such as those powered by language models (e.g., GPT-4), which can write code, interface with external systems, and communicate with other agents.

Characteristics of Multi-Agent Applications



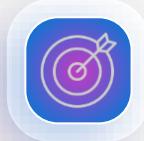
Run within the same process or on the same machine



Operate across different machines or organizational boundaries



Be implemented in diverse programming languages and make use of different AI models or instructions



Work together towards a shared goal, coordinating their actions through messaging



Each agent is a self-contained unit that can be developed, tested, and deployed independently.



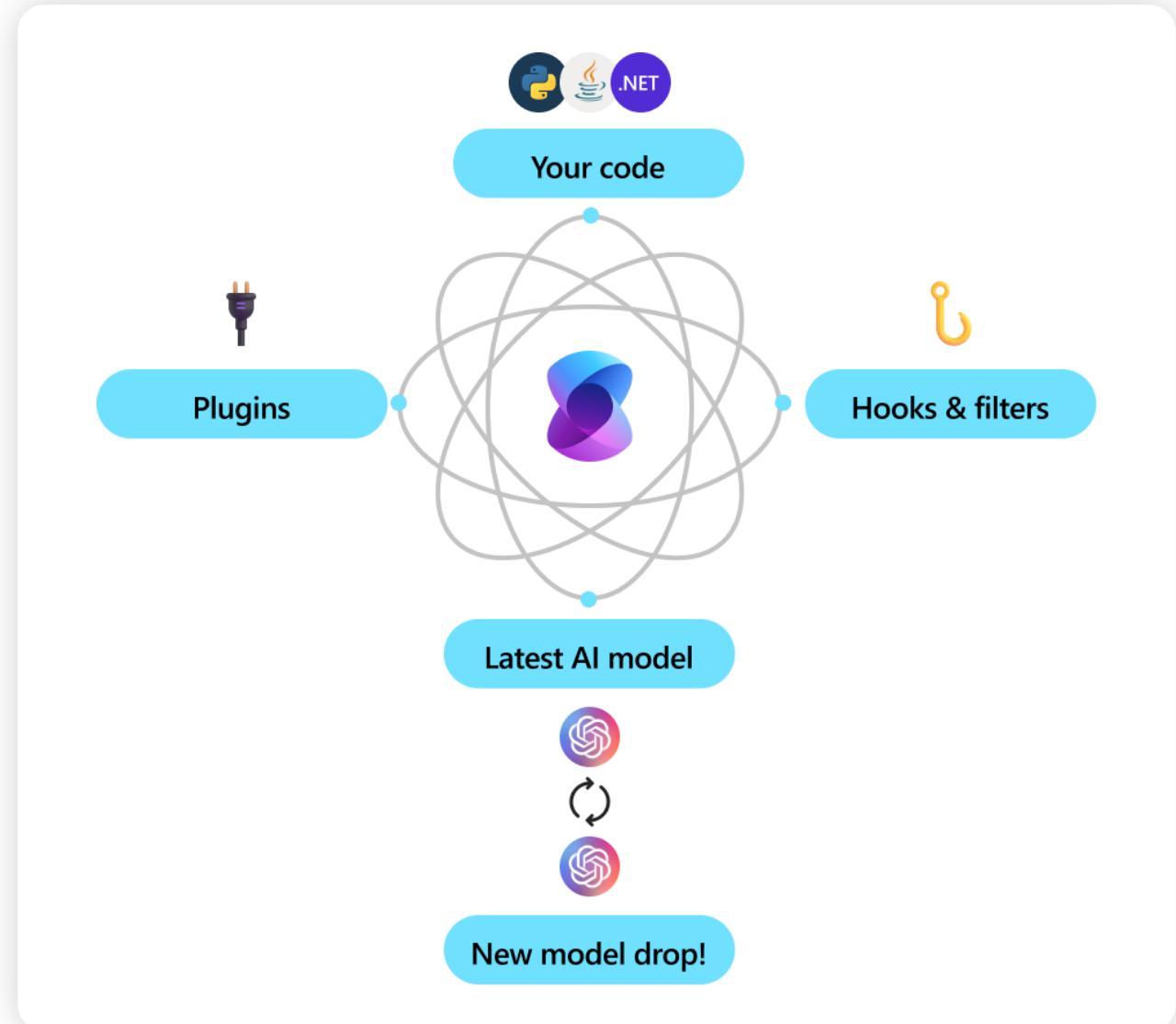
Agents are inherently **composable**: simple agents can be combined to form complex

Understanding Semantic Kernel



Semantic Kernel Introduction

- Semantic Kernel is a lightweight, open-source development kit that lets you easily build AI agents
- Integrates the latest AI models into your C#, Python, or Java codebase
- an efficient middleware that enables rapid delivery of enterprise-grade solutions



Semantic kernel – Key features

Enterprise ready

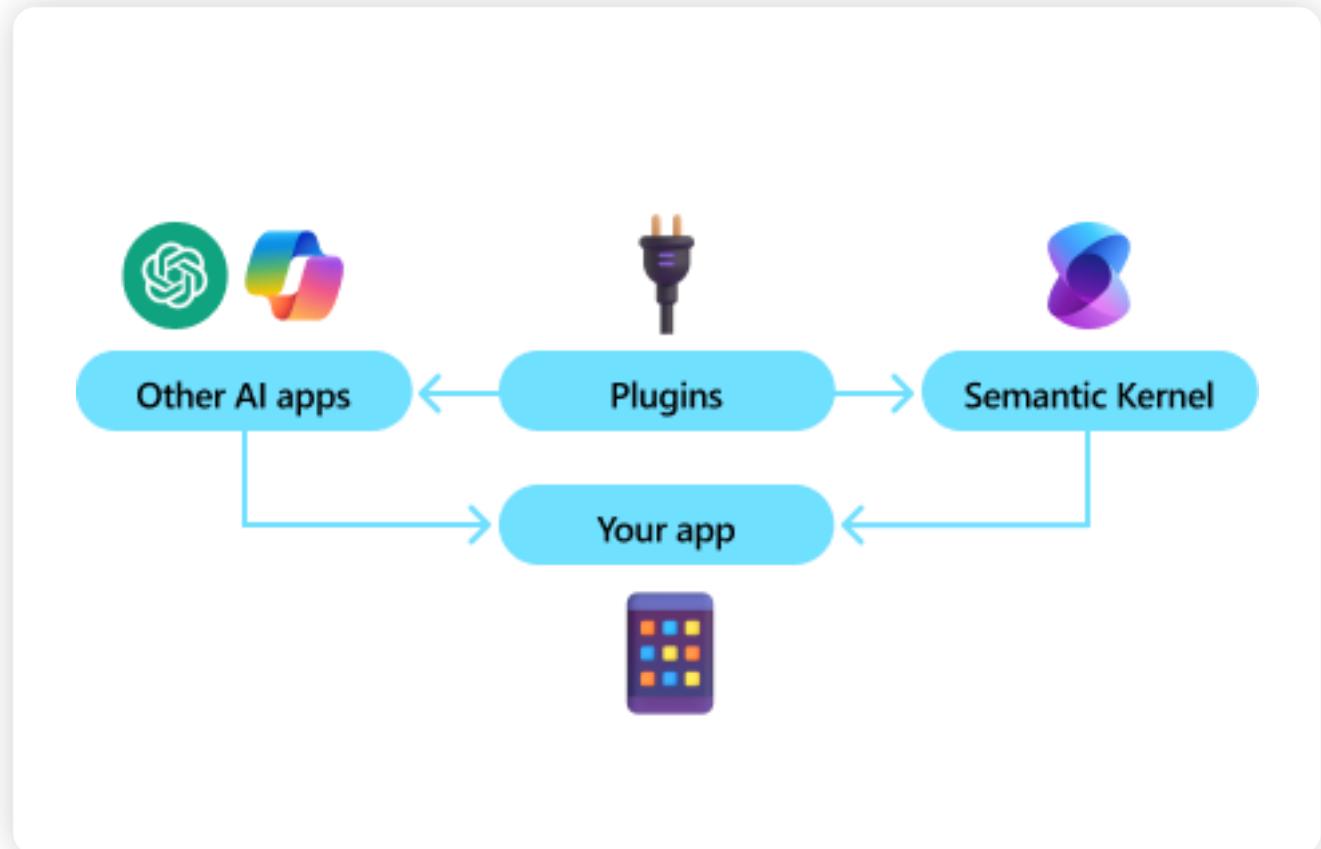
Microsoft and other Fortune 500 companies are already leveraging Semantic Kernel because it's flexible, modular, and observable

Automating business processes

Combines prompts with existing APIs to perform actions

Modular and extensible

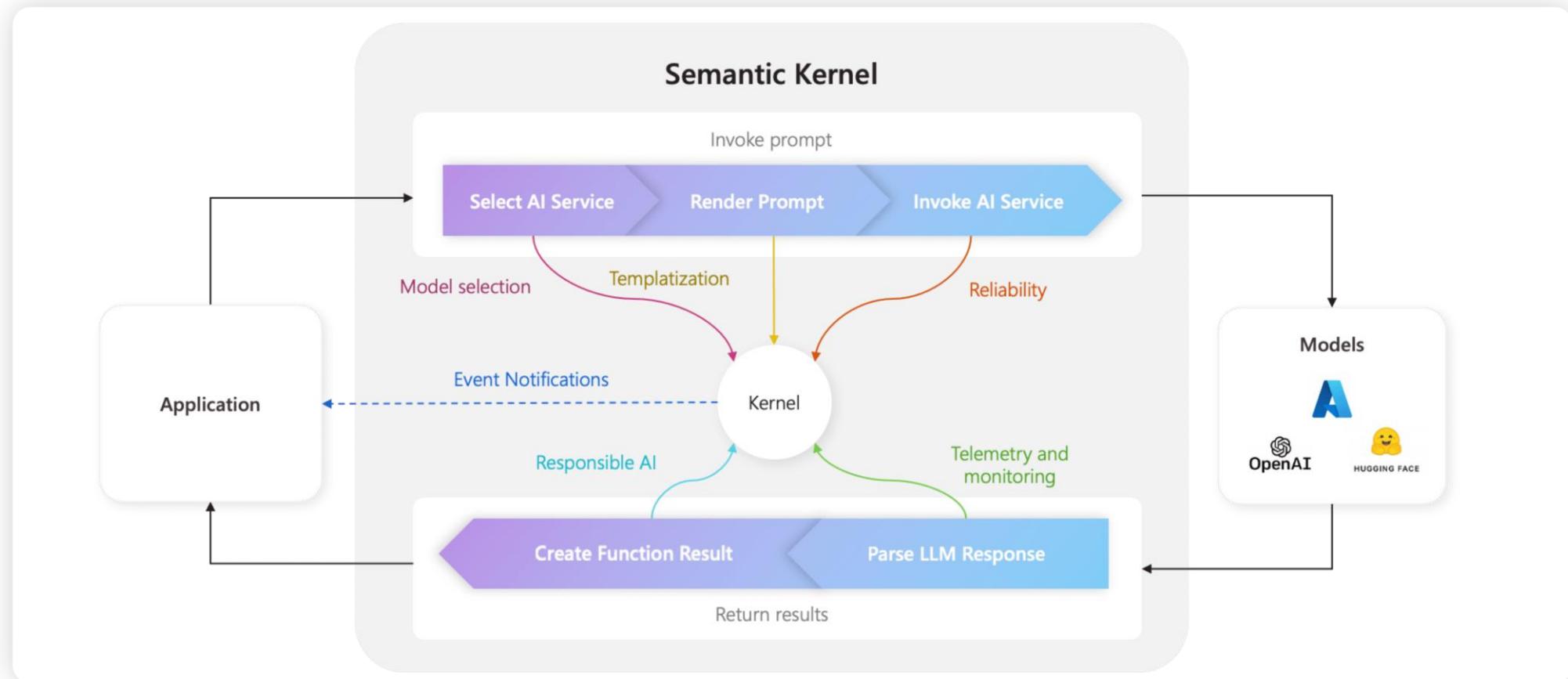
Flexibly integrate AI services through a set of out-of-the-box connectors



Understanding the kernel

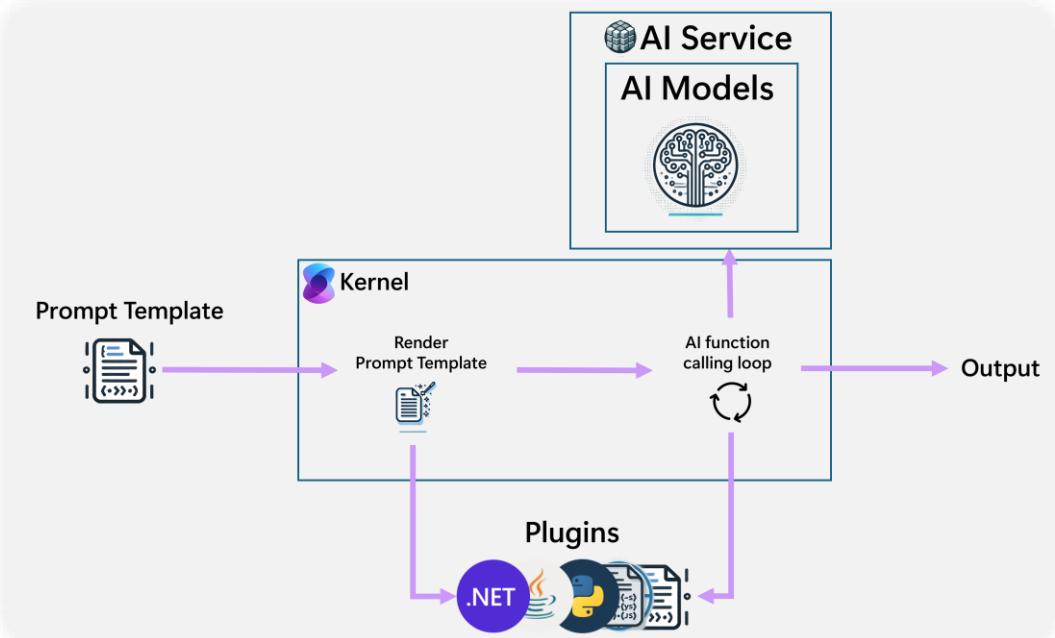
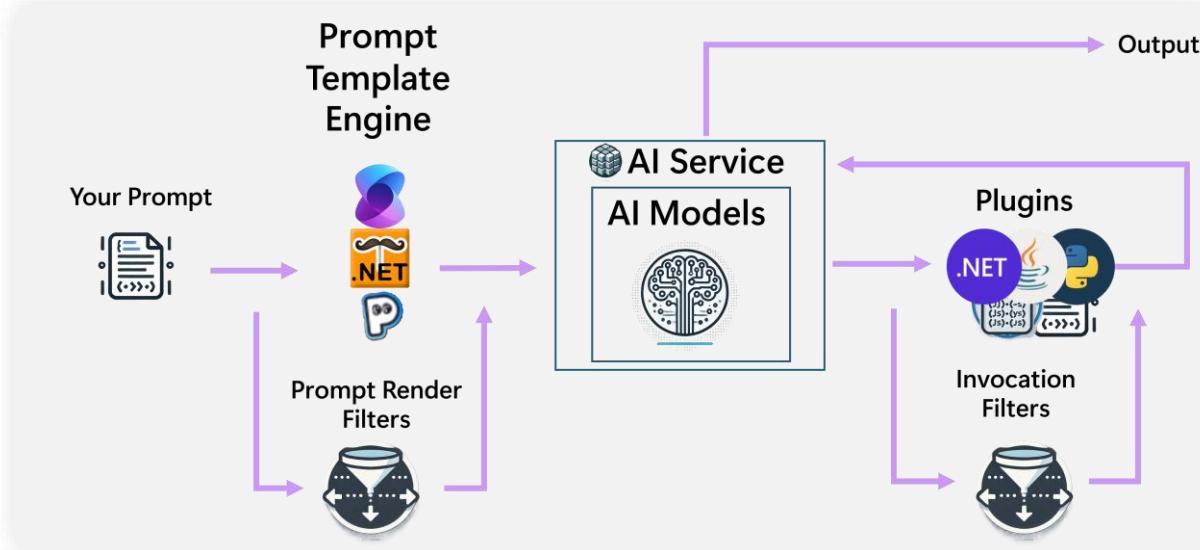
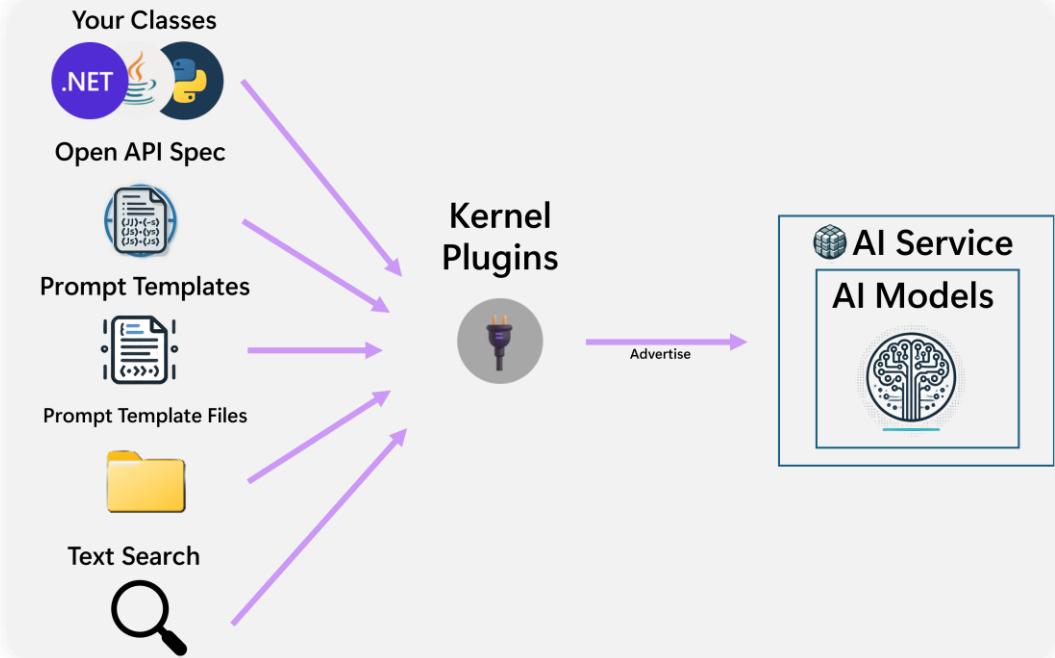
The kernel is at the center

- Kernel has all of the services and plugins necessary to run both native code and AI services
- It is used by nearly every component within the Semantic Kernel SDK to power your agents



Semantic Kernel Components

- AI Service Connectors
- Vector Store (Memory) Connectors
- Functions and Plugins
- Prompt Templates
- Filters



Semantic Kernel Agent Framework

Semantic Kernel Agent Framework provides a platform within the Semantic Kernel eco-system that allow for the creation of **AI agents**

Provides ability to incorporate **agentic patterns** into any application based on the same patterns and features that exist in the core Semantic Kernel framework.

What is an AI agent?

- A software entity designed to perform tasks autonomously or semi-autonomously by receiving input, processing information, and taking actions to achieve specific goals.
- Agents can send and receive messages, generating responses using a combination of models, tools, human inputs, or other customizable components

What problems do AI agents solve?

- Modular Components
- Collaboration
- Human-Agent Collaboration
- Process Orchestration

When to use an AI agent?

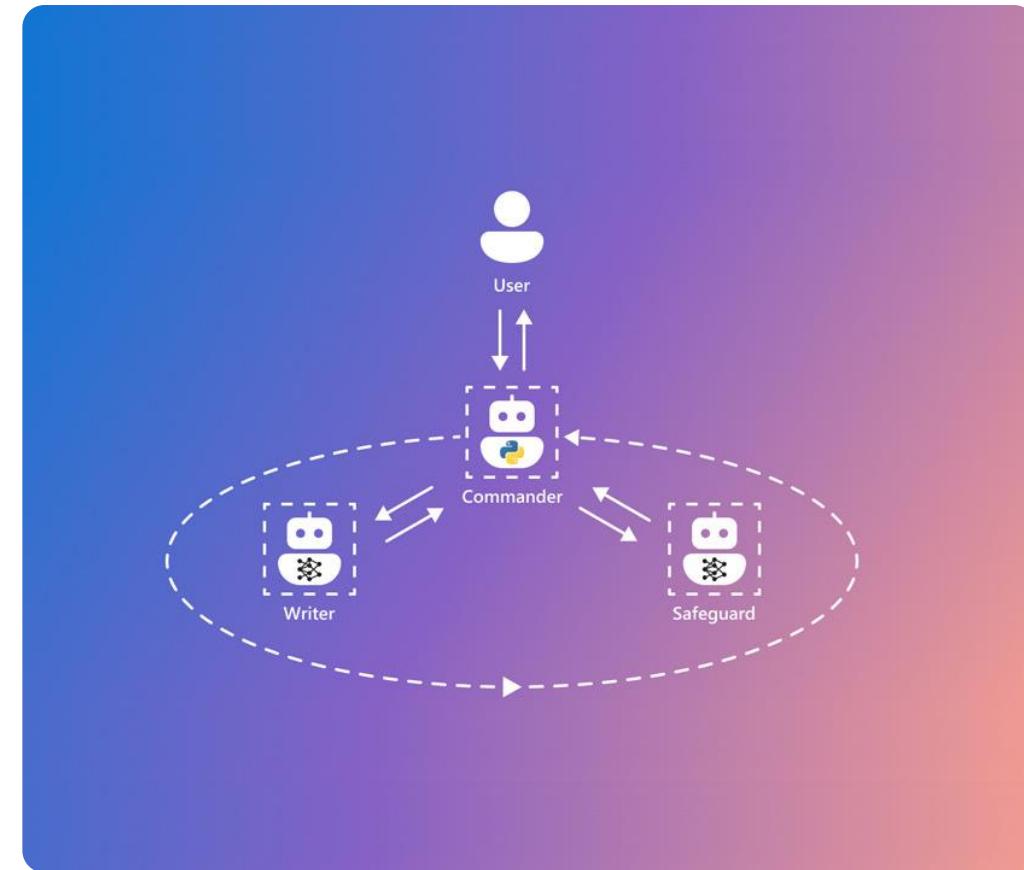
- Autonomy and Decision-Making
- Multi-Agent Collaboration
- Interactive and Goal-Oriented

Understanding AutoGen Agent Framework



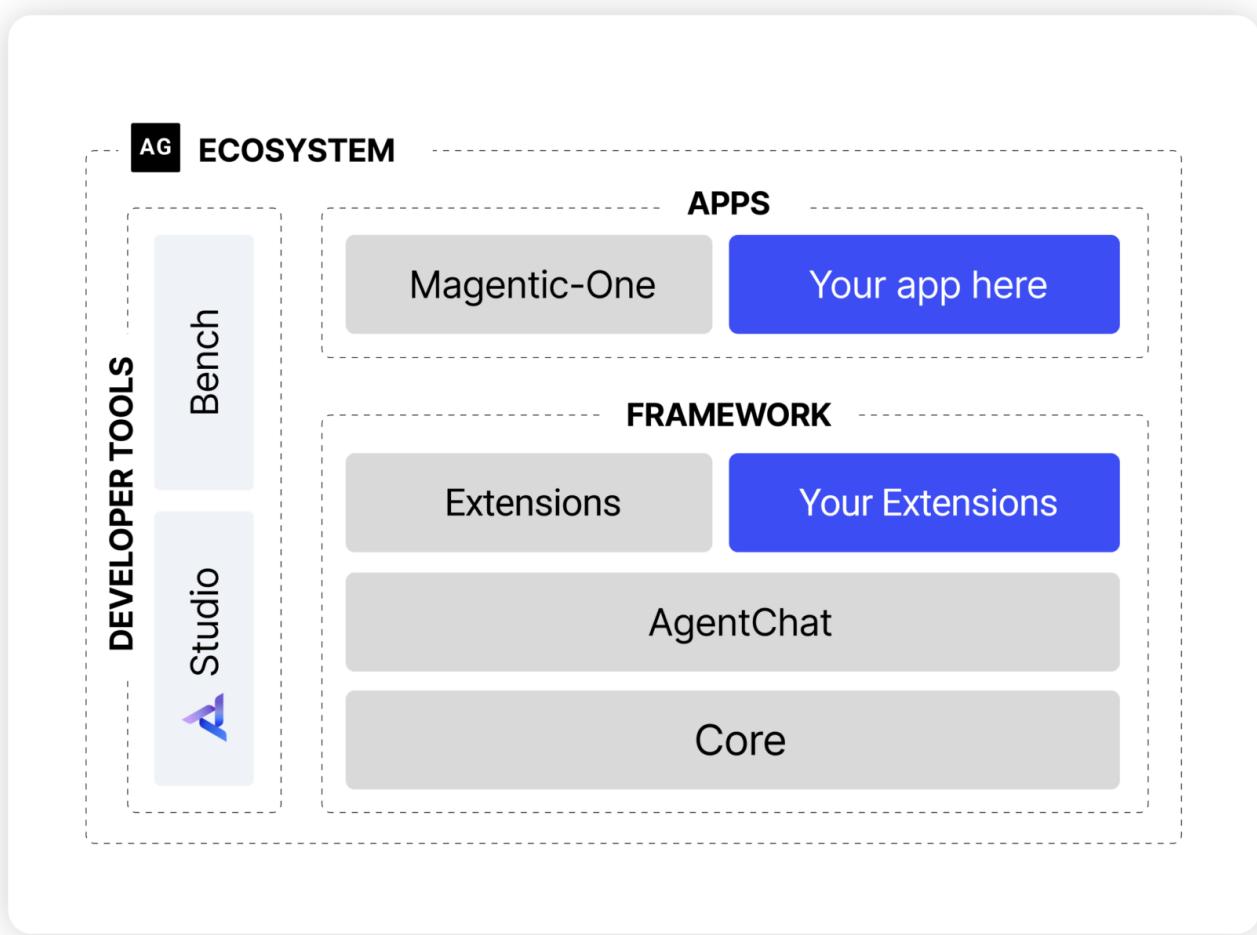
AutoGen Introduction

- AutoGen is an open-source programming framework for building AI agents and facilitating cooperation among multiple agents to solve tasks
- Aims to provide an easy-to-use and flexible framework for accelerating development and research on agentic AI



AutoGen Ecosystem and key Features

- Asynchronous messaging
- Modular and extensible
- Observability and debugging
- Scalable and distributed
- Built-in and community extensions
- Cross-language support
- Full type support



AutoGen - A framework for building AI agents and applications

Magnetic-One CLI

A console-based multi-agent assistant for web and file-based tasks. Built on AgentChat.

Studio

An app for prototyping and managing agents without writing code. Built on AgentChat.

AgentChat

A programming framework for building conversational single and multi-agent applications. Built on Core. Requires Python 3.10+.

Core

An event-driven programming framework for building scalable multi-agent AI systems.

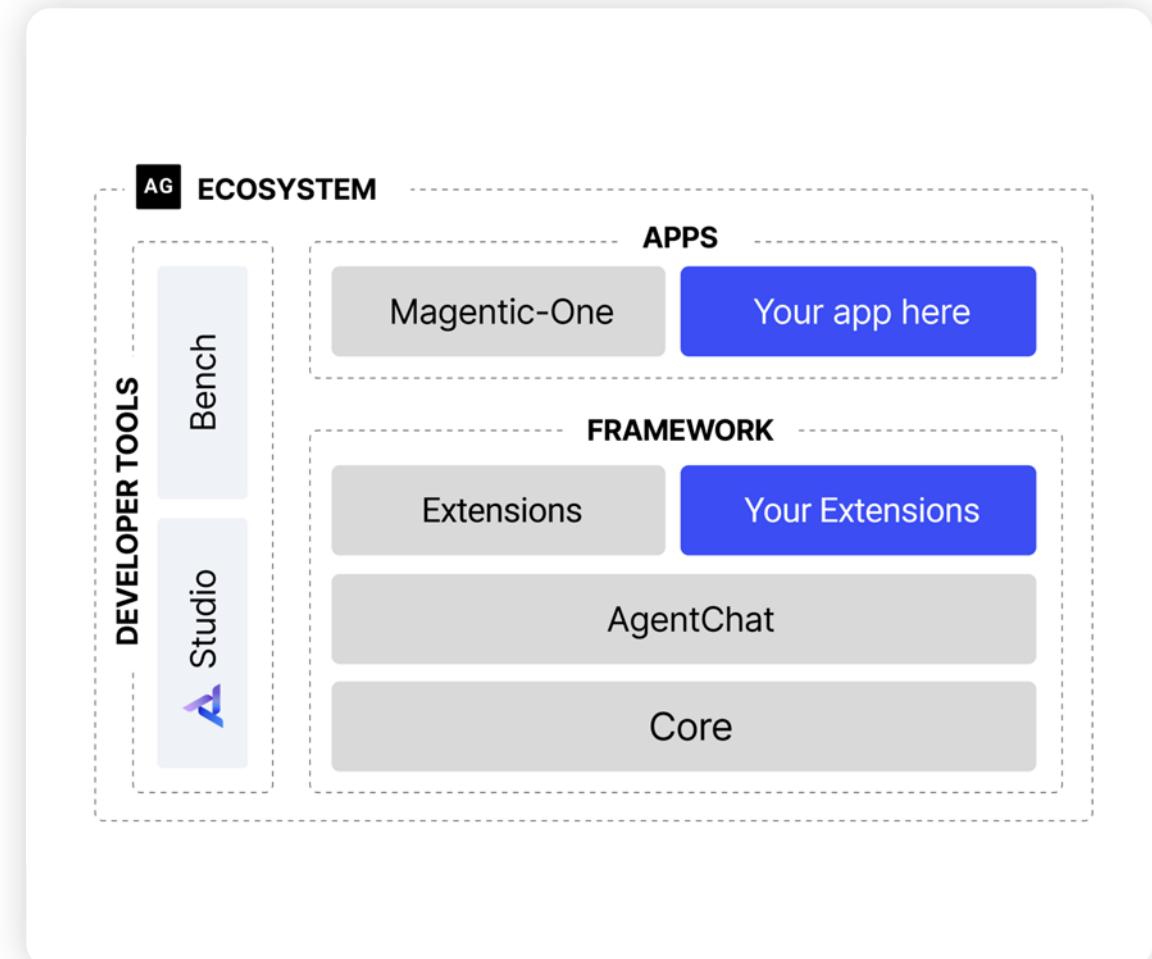
Example scenarios:

- Deterministic and dynamic agentic workflows for business processes.
- Research on multi-agent collaboration.
- Distributed agents for multi-language applications.

Extensions

Implementations of Core and AgentChat components that interface with external services or other libraries. You can find and use community extensions or create your own. Examples of built-in extensions:

- LangChainToolAdapter for using LangChain tools.
- OpenAIAssistantAgent for using Assistant API.
- DockerCommandLineCodeExecutor for running model-generated code in a Docker container.
- GrpcWorkerAgentRuntime for distributed agents.



AutoGen - Core



Asynchronous Messaging

Agents communicate through asynchronous messages, enabling event-driven and request/response communication models.



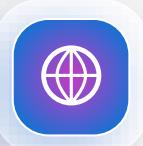
Scalable & Distributed

Enable complex scenarios with networks of agents across organizational boundaries.



Multi-Language Support

Python & Dotnet interoperating agents today, with more languages coming soon.



Modular & Extensible

Highly customizable with features like custom agents, memory as a service, tools registry, and model library.



Observable & Debuggable

Easily trace and debug your agent systems.



Event-Driven Architecture

Build event-driven, distributed, scalable, and resilient AI agent systems.

AutoGen - AgentChat

A high-level API for building multi-agent applications. It is built on top of the **autogen-core** package

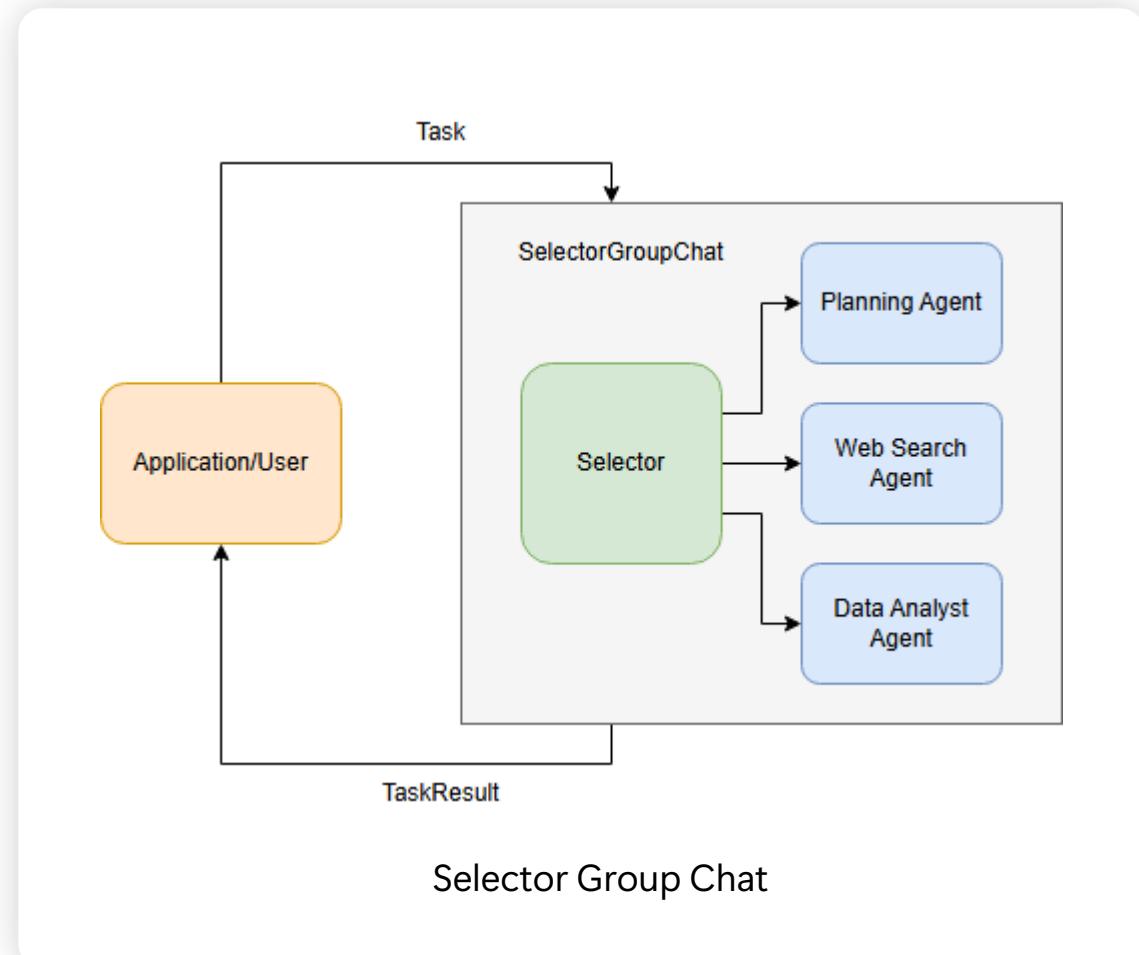
- Provides intuitive defaults, such as Agents with preset behaviors and Teams with predefined multi-agent design patterns

AgentChat > Selector Group Chat

Implements a team where participants take turns broadcasting messages to all other members.

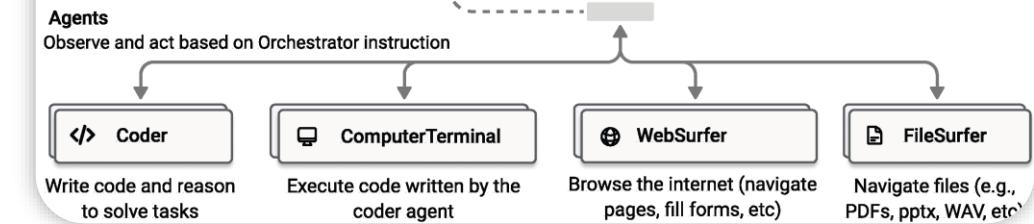
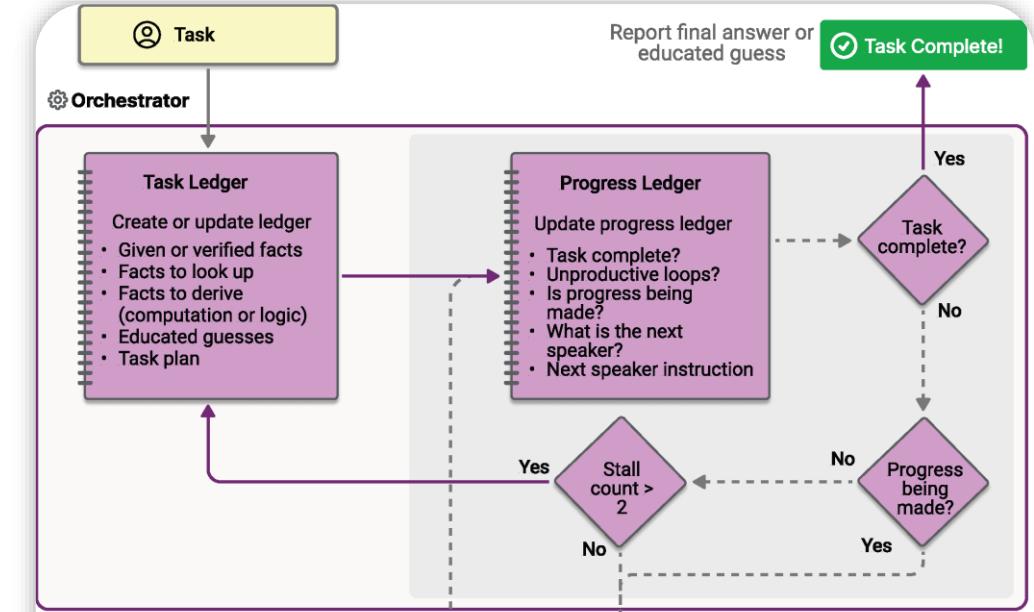
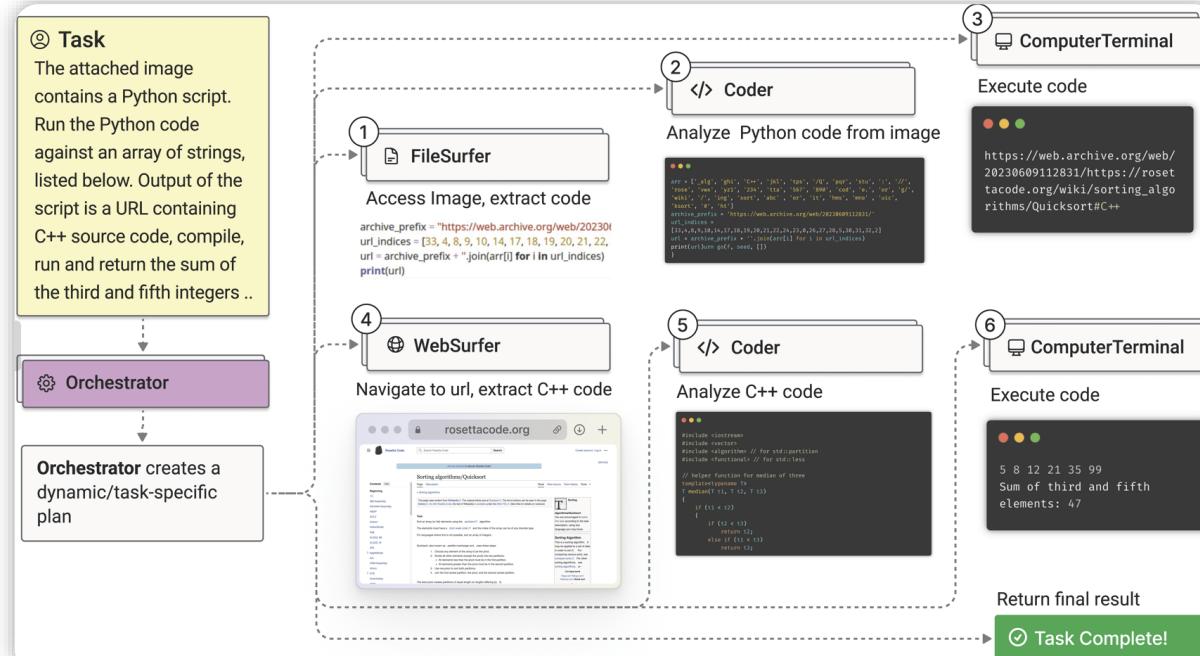
Key features include:

- Model-based speaker selection
- Configurable participant roles and descriptions
- Prevention of consecutive turns by the same speaker (optional)
- Customizable selection prompting
- Customizable selection function to override the default model-based selection
- Customizable candidate function to narrow-down the set of agents for selection using mode



AgentChat > Magentic-One

A generalist multi-agent system for solving open-ended web and file-based tasks across a variety of domains originally released in November 2024 Magentic-One was implemented directly on the **autogen-core** library



```
pip install "autogen-agentchat" "autogen-ext[magnetic-one,openai]"  
  
# If using the MultimodalWebSurfer, you also need to install playwright dependencies:  
playwright install --with-deps chromium
```

Architecture

AutoGen - Extensions

AutoGen is designed to be extensible. The autogen-ext package contains the built-in component implementations maintained by the AutoGen project

Examples of components include:

- `autogen_ext.agents.*` for agent implementations like `MultimodalWebSurfer`
- `autogen_ext.models.*` for model clients like `OpenAIChatCompletionClient` and `SKChatCompletionAdapter` for connecting to hosted and local models.
- `autogen_ext.tools.*` for tools like GraphRAG `LocalSearchTool` and `mcp_server_tools()`.
- `autogen_ext.executors.*` for executors like `DockerCommandLineCodeExecutor` and `ACADynamicSessionsCodeExecutor`
- `autogen_ext.runtimes.*` for agent runtimes like `GrpcWorkerAgentRuntime`

You can discover community extensions and samples or Create your own extensions

AutoGen Studio

The screenshot shows the AutoGen Studio interface in 'Visual builder mode (experimental)'. On the left, there's a sidebar with icons for Teams, Recent, and From GitHub. The main area displays 'Teams' with two entries: 'Web Agent Team (Operator)_17388' (3 agents, 1 hour ago) and 'Default Team_17388' (1 agent, 2 hours ago). Below this is a large blue banner for 'AutoGen Studio v0.4' with the URL <https://github.com/microsoft/autogen>. The central workspace is titled 'Web Agent Team (Operator)_17388' and contains a 'Component Library' where components can be dragged to add them to the team. Components shown include 'MultimodalWebSurfer', 'AssistantAgent', and 'UserProxyAgent'. Each component has a detailed description, model, and tools section.

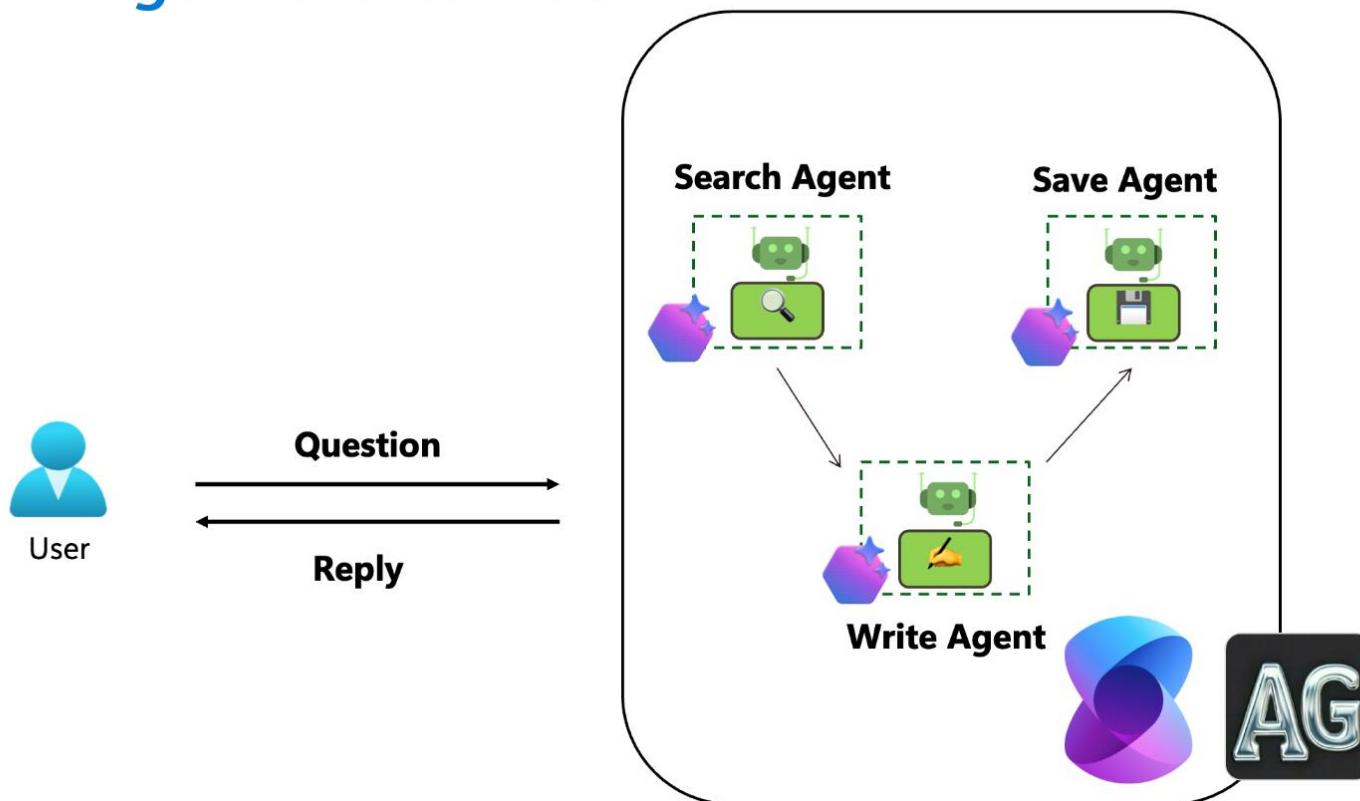
This screenshot shows another view of the AutoGen Studio 'Team Builder' interface. It features a sidebar with 'Teams' and a 'Test Team' button. The main area shows a 'Component Library' with sections for Agents, Models, Tools, and Terminations. Components listed include 'MultimodalWebSurfer', 'AssistantAgent', and 'UserProxyAgent'. A detailed description of each component is provided, along with their respective models and tools. The interface is clean and modern, designed for easy drag-and-drop component placement.

See a video tutorial on
AutoGen Studio v0.4 (02/25)

-
<https://youtu.be/oum6El7wohM>

Multi-Agent Collaboration & Orchestration with AutoGen / Semantic Kernel

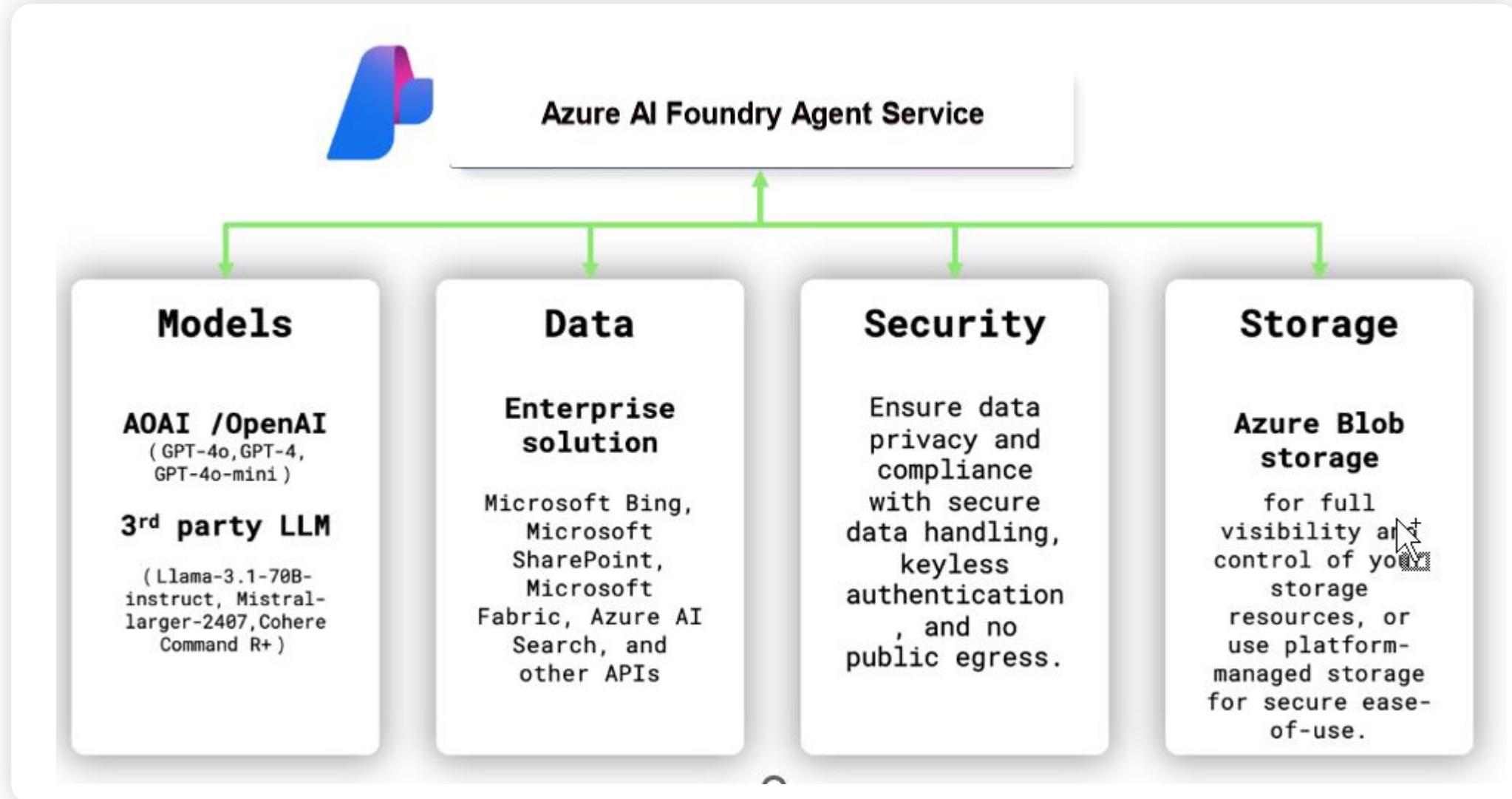
Multi-Agent Orchestration



Multi-Agent Collaboration & Orchestration with AutoGen / Semantic Kernel



Azure AI Foundry Agent Service



Tips for creating AI Agents in Azure AI Foundry SDK

Use the [template](#) to create a project in Azure AI Foundry

Region	gpt-4o, 2024-05- 13	gpt-4o, 2024-08- 06	gpt-4o- mini, 2024-07- 18	gpt-4, 0613	gpt-4, 1106- Preview	gpt-4, 0125- Preview	gpt-4, vision- preview	gpt-4, turbo- 2024-04- 09	gpt-4- 32k, 0613	gpt-35- turbo, 0613	gpt-35- turbo, 1106	gpt-35- turbo, 0125	gpt-35- turbo-16k, 0613
eastus	✓	✓	✓	-	-	✓	-	✓	-	✓	-	✓	✓
francecentral	-	-	-	✓	✓	-	-	-	✓	✓	✓	-	✓
japaneast	-	-	-	-	-	-	✓	-	-	✓	-	✓	✓
uksouth	-	-	-	-	✓	✓	-	-	-	✓	✓	✓	✓
westus	✓	✓	✓	-	✓	-	✓	✓	-	-	✓	✓	-

Check Regions and models availability

You can also use the Azure AI Foundry model catalog

- Llama 3.1-70B-instruct
- Mistral-large-2407
- Cohere command R+

Example Scenario: Blog writing

This scenario involves three AI agents:



One for content collection



Another for writing assistance



A third for content storage and management

These agents can be seamlessly orchestrated using AutoGen or Semantic Kernel

Azure AI Foundry Agent Service defines a single agent

Using Grounding with Bing Search



Grounding with Bing Search

Grounding with Bing Search adds intelligent search results to your AI Agents, expanding the capabilities of AI models with relevant and up-to-date information. It combines the power of billions of publicly available web pages to provide real-time results for the agents to craft the desired output.

This tool can only be used as a knowledge source for the Azure AI Agent Service, in combination with other data sources. By enabling this tool, the Agent is able to search the web for the required information to fulfill the user query. The final output includes the model result grounded with Bing data.

Content collection agent

Add a custom api key connection

← Back to select an asset type

Endpoint *
https://<your-endpoint>

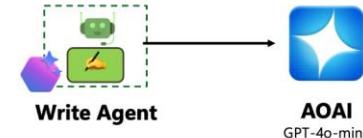
Authentication *
API key

Key *

Connection name * ⓘ
Input connection name

Access ⓘ
Shared to all projects

Write Agent



You are a blog writer, please help me write a blog based on bing search content.

Writer agent

Save Agent



AOAI
GPT-4o-mini

Code Interpreter

You are my Python programming assistant. Generate code and execute it according to the following requirements

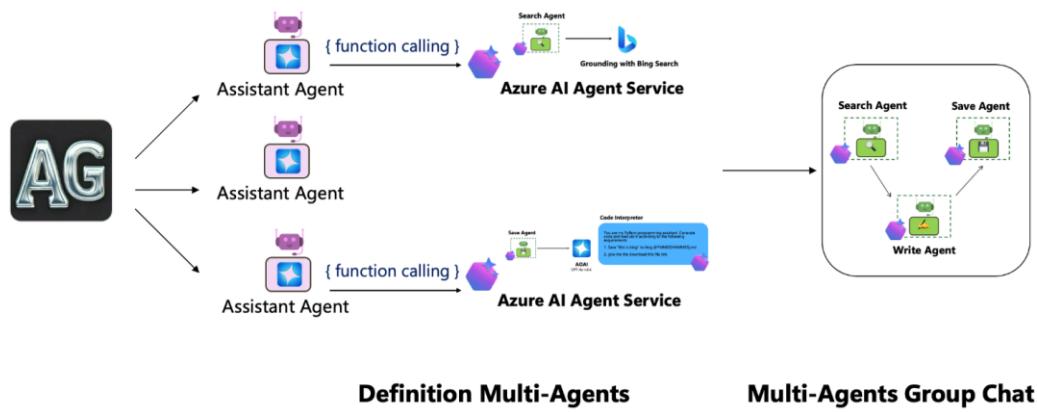
1. Save "this is blog" to blog-(YYMMDDHHMMSS).md
2. give me the download this file link

Save agent

Samples: <https://github.com/kinfey/MultiAIAgent/>

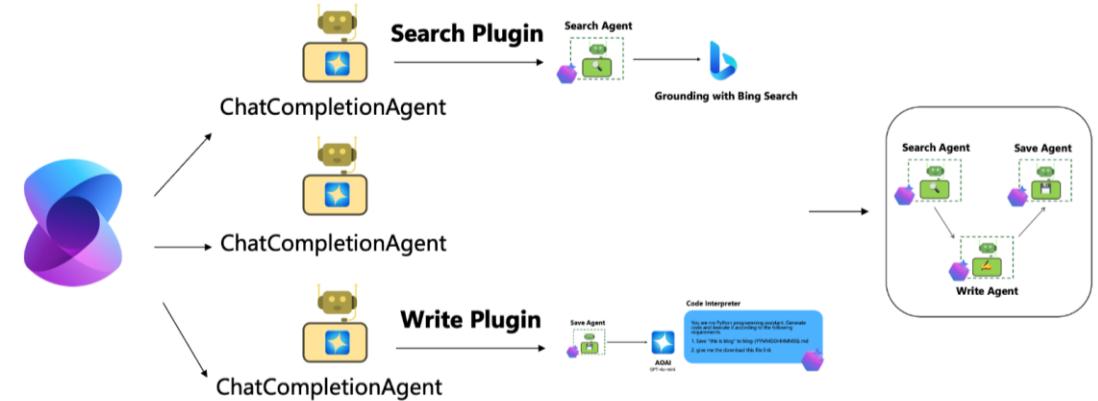
Use AutoGen /Semantic Kernel to orchestrate multi-agents

AutoGen Multi-Agents Orchestration



AutoGen's orchestration

Semantic Kernel Multi-Agents Orchestration



Semantic Kernel's orchestration

AutoGen- Multi-agent Conversation Framework

- AutoGen offers a unified multi-agent conversation framework as a high-level abstraction of using foundation models.
- Features capable, customizable and conversable agents which integrate LLMs, tools, and humans via automated agent chat.
- Simplifies the orchestration, automation and optimization of a complex LLM workflow
- Enables building next-gen LLM applications based on multi-agent conversations with minimal effort.

```
import os
from autogen import AssistantAgent, UserProxyAgent
from autogen.coding import DockerCommandLineCodeExecutor

config_list = [{"model": "gpt-4", "api_key": os.environ["OPENAI_API_KEY"]}]

# create an AssistantAgent instance named "assistant" with the LLM configuration.
assistant = AssistantAgent(name="assistant", llm_config={"config_list": config_list})

# create a UserProxyAgent instance named "user_proxy" with code execution on docker.
code_executor = DockerCommandLineCodeExecutor()
user_proxy = UserProxyAgent(name="user_proxy", code_execution_config={"executor": code_executor})
```

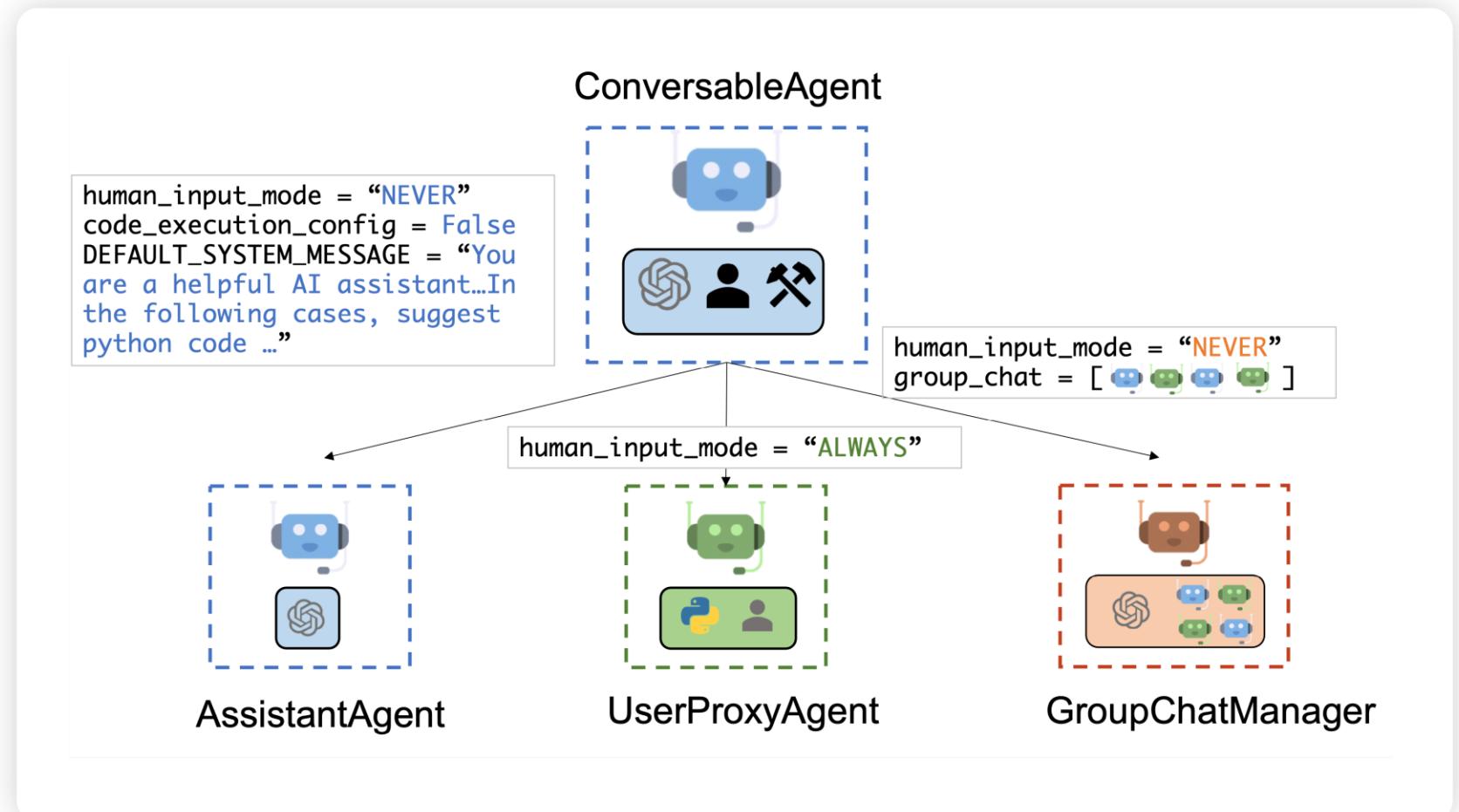
Agents in AutoGen

Conversable

Any agent can send and receive messages from other agents to initiate or continue a conversation

Customizable

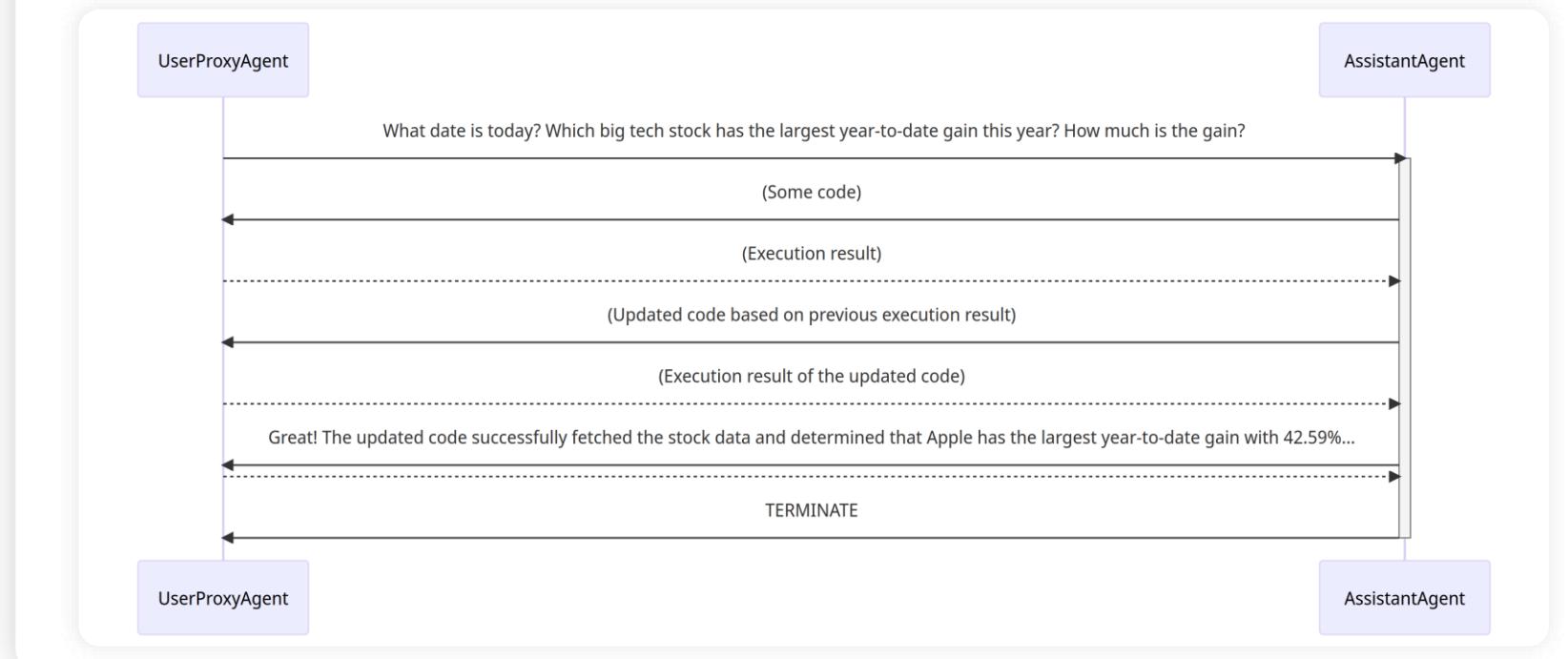
Agents in AutoGen can be customized to integrate LLMs, humans, tools, or a combination of them



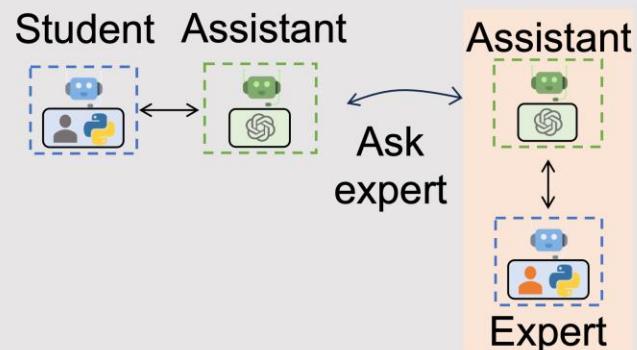
Multi-agent Conversations

A Basic Two-Agent Conversation Example

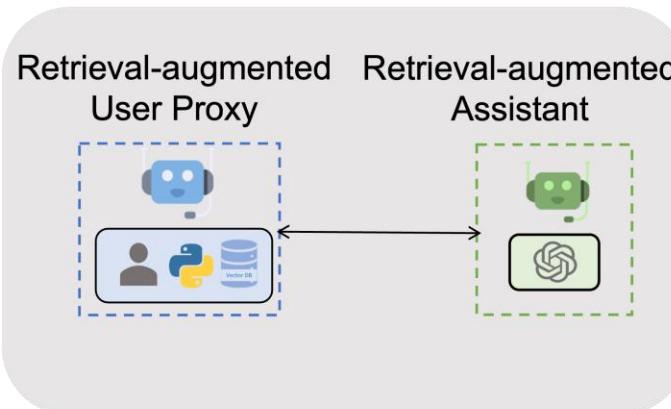
```
# the assistant receives a message from the user, which contains the task description
user_proxy.initiate_chat(
    assistant,
    message="""What date is today? Which big tech stock has the largest year-to-date gain this
year? How much is the gain?""",
)
```



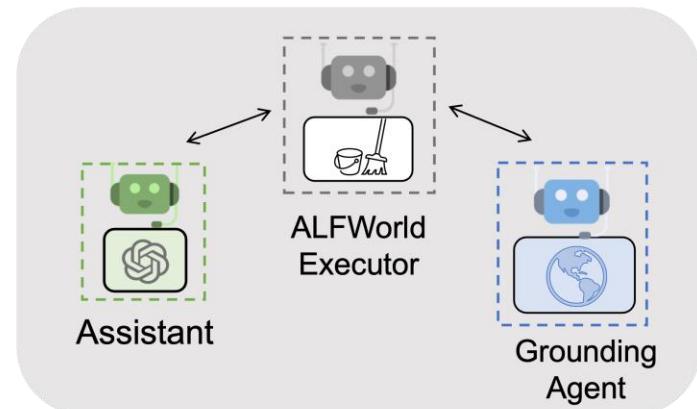
Diverse Applications Implemented with AutoGen



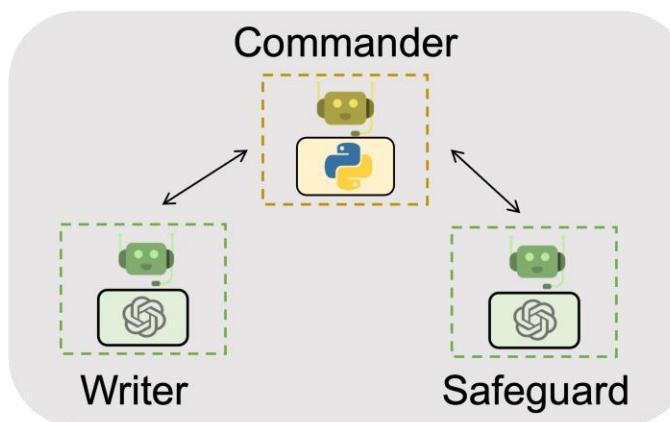
A1. Math Problem Solving



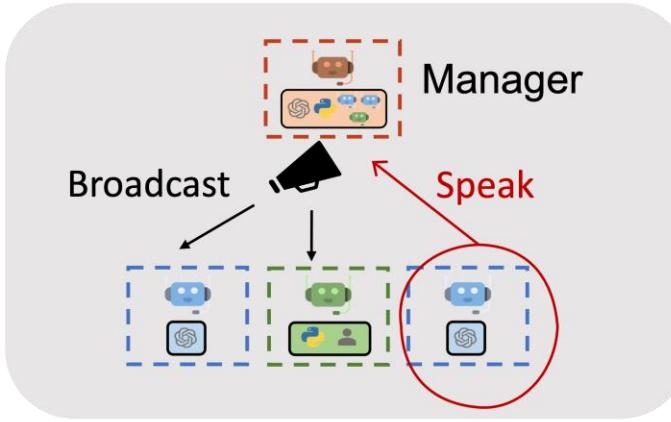
A2. Retrieval-augmented Chat



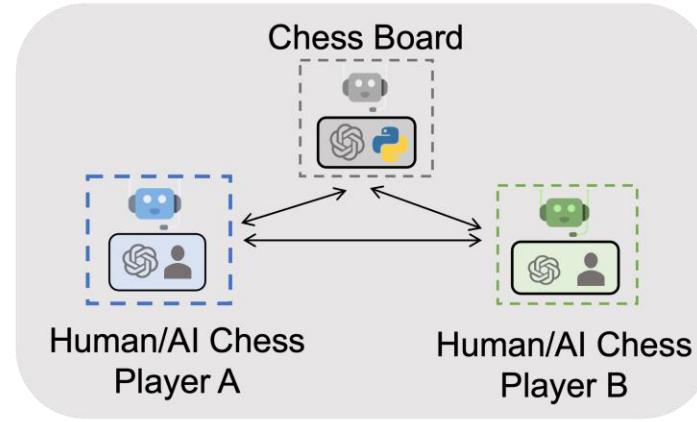
A3. Decision Making



A4. Multi-agent Coding



A5. Dynamic Group Chat



A6. Conversational Chess

Find a list of examples here: [Automated Agent Chat Examples](#)

Orchestration frameworks – Microsoft First-Party



Semantic Kernel

- Full SDK designed to build AI agents with ease, excellent for single agents and can be extended for multi-agents with integrations to AutoGen
- Extensible and compatible with models LLMs or SLMs.
- Ideal for developers looking to leverage AI orchestration patterns similar to those used in Microsoft's Copilot systems in their own applications
- Provides a framework that goes beyond message generation, enabling the calling of existing code to perform productive tasks



AutoGen

- Powerful multi-agent framework with prebuild conversation orchestration patterns for handling complex agent systems
- Extensible and compatible with models LLMs or SLMs.
- Studio offers a no-code low-code approach to designing agents.
- Abstraction layer makes it easy for seasoned developers to build complex agent systems with proven patterns
- Experiment tool – not a product or supported

Hands-on Labs



Building Intelligence: Advanced Agent Development in Azure AI Foundry

⌚ 4 days (5 hours/day)



Module 1: Innovate with Microsoft 365 Copilot and agents

- M365 Copilot and agents
- How Copilot works
- Semantic index for Copilot
- M365 Copilot Chat
- Agents Use cases
- New agents in Microsoft 365
- Unlock more value with SharePoint agents

Module 2: Extend Microsoft 365 Copilot with Agents

- Microsoft 365 Extensibility Planning and approach
- Declarative agents and agent tooling
- Explore Copilot Studio Agent Builder
- Build declarative agents with Microsoft 365 Agents Toolkit
- Build custom agents with Copilot Studio
- Autonomous agents overview
- Agent Governance - Overview
- Gen-AI decision guide – when to build, buy or extend

Hands-on Labs ⌚ 3 hours

- Explore Copilot Studio Agent Builder
- Build HR Assistant Agent with Copilot Studio
- Incorporate actions in HR Agent
- Enable Autonomous Capabilities in Microsoft Copilot Studio for HR Activities

Module 3: Customize Agents with Gen AI in Copilot Studio

- Customizing your agents – Orchestrator, UI, Knowledge, Actions, Autonomy
- Copilot Studio implementation guidance for architects
- Generative AI in Copilot Studio
- Copilot Studio + Power Platform
- Building voice-enabled agents
- AI Foundry integration
- Developing agents using Microsoft 365 Agents SDK

Module 4: Innovate with Azure AI Platform

- How language models work
- AI Foundry and SDK introduction
- AI Foundry Model Catalog
- Azure AI Services
- Azure OpenAI Service and model guidance
- Models-as-a-Service
- Azure AI Foundry Agent Service
- Safeguard with Trustworthy AI

Module 5: Customize, orchestrate and experiment with Azure AI Foundry

- Retrieval Augmented Generation (RAG)
- Customizing models – Fine tuning, distillation
- Responses API (preview)
- Azure AI Foundry Agent Service - Orchestrate and debug AI workflows

Module 6: Build your own multi agents with Semantic Kernel or AutoGen

- Multi-agent applications
- Understanding Semantic Kernel
- Understanding AutoGen Agents Framework
- Multi-Agent Collaboration & Orchestration with AutoGen / Semantic Kernel

Module 7: Enterprise grade production at scale

- Scaling challenges and agent controls
- Manage AI performance in production
- Observability Tools
- Enabling Enterprise governance and management
- Enterprise grade security and data protection
- Monitoring and observability

Module 8: Advanced AI risk evaluation and mitigation

- Identifying risks
- Azure AI Content Safety
- Evaluation and GenAIOps
- Identity and access management
- Network Security for AI apps
- Continuous security for AI



Capstone Project

Objective: Apply your skills to design, develop, and deploy a sophisticated, end-to-end AI solution leveraging Azure AI services, Microsoft Copilot frameworks, and multi-agent orchestration techniques

Hands-on Labs ⌚ 3 hours

- Set Up Azure AI Foundry SDK and Provision Resources
- Build a Retrieval-Augmented Generation(RAG) Pipeline
- Evaluate and Optimize RAG Performance
- Semantic Kernel Fundamentals
- Semantic Kernel Plugins

Hands-on Labs ⌚ 3 hours

- Understanding the Lifecycle of Flow Development
- Building and Customizing Prompt Flows
- Evaluation Flow Setup
- Fine-Tuning Prompts for Optimal Performance
- Implementing Chat Flow and Tool Integration
- Ensuring Responsible AI Practices with Content Safety



Hands-on Lab



Please note that the labs are only open to a limited number of Microsoft partner participants and are offered on a **first-come-first-served basis**.

Step 1 : Click on **Launch** in the event curriculum to launch your labs

Date/Time (PDT)	Title	Track	ICS Training Bag
On-Demand	Spektra Hands-on lab: Agentic AI Accelerator AMER (PDT)	AMER (PDT)	Launch

Step 2 : Join the **Live session for Lab Support and Q&A Session**

Lab Support and Q&A	Lab Support and Q&A		Sign Up	
<p>Abstract: We invite you to join this session to begin your labs. The Trainer will provide you with instructions, demonstrate live scenarios and respond to your questions.</p>				

*For all Lab logistical support issues, please go to: <http://cloudlabs.ai/ms-support/> and contact via chat

Sign-up and launch Hands-on Labs

Note: Please register and launch the lab only when the hands-on lab session is about to begin (not at the start of the day)



- Click the link to complete your registration -
<https://bit.ly/3Hfoc0>



- Enter the Email id and **Activation code** (based on your time zone).

IST	AI2084IST
BST	AI2084BST
PDT	AI2084PDT



- Select the preferred language, accept the **Usage policy** and select **Submit** to complete registration.



- Once registration is completed, click on **Launch Lab** to activate the workshop.

The screenshot shows a registration form titled "Register Now". It includes fields for "Email*" and "Activation Code*". Below these are radio buttons for selecting a language: English (selected), Spanish, Japanese, Korean, Simplified Chinese, Traditional Chinese, and Portuguese. There is also a checkbox for accepting the Microsoft Usage Policy, which links to a Privacy Policy. A large blue "Submit" button is at the bottom. A red arrow points from the "Submit" button to a callout box below it.

Please click on the 'Launch Lab' button to activate your lab environment.

Launch Lab

FAQs - Lab Experience

Lab Guide (First Tab)

Provides step-by-step instructions for performing the lab exercises.

The screenshot shows the 'Guide' tab selected in the top navigation bar. The main content area displays the 'Azure Agentic AI Workshop' overview, which includes an 'Overview' section describing the workshop series and its objectives, and an 'Objectives' section listing ten bullet points about AI development and deployment.

Overview:
This hands-on workshop series is designed to help you deepen your skills in AI-driven development, automation, and building intelligent applications using Microsoft Copilot Studio and Azure AI services. Each day offers practical, guided experiences—from designing AI agents to deploying full-scale AI solutions. By the end of this workshop, you'll walk away with hands-on expertise in creating scalable, intelligent solutions that boost employee engagement, streamline business operations, and enhance customer experiences.

Objectives:

- Learn to design and deploy AI agents with Microsoft Copilot Studio and Azure AI services.
- Gain hands-on experience in multi-agent orchestration using the Azure AI Agent Service SDK and Semantic Kernel.
- Build custom Retrieval-Augmented Generation (RAG) applications with Azure AI Foundry and integrate Semantic Kernel plugins.
- Evaluate, fine-tune, and deploy AI models using Prompt Flow for real-world use cases.
- Construct intelligent escalation systems using conversational interfaces, event-driven architecture, and AI-powered workflows.

Environment (Second Tab)

Lists the credentials required to access the environment and the lab

The screenshot shows the 'Environment' tab selected in the top navigation bar. It displays a table of authentication fields with their values, and a 'Resource Group : lab-vm' table with deployment ID and admin details.

Auth Fields	Value	Action
Username	odl_user_1734975@msa	Copy
Password	xtnt26BED*9K	Copy

Resource Group : lab-vm

Key	Value	Action
Deployment ID	1734975	Copy
Labvm Admin Username	demouser	Copy

Resources (Third Tab)

Allows you to manage resources (e.g., Start, Restart, or Stop VMs.)

The screenshot shows the 'Resources' tab selected in the top navigation bar. It displays a table of virtual machines with columns for Name, Status, and Actions. One row for 'labvm-1734975' is highlighted with a red box.

Name	Status	Actions
labvm-1734975	VM running	

FAQs – Lab Experience

Help (Fourth Tab)

Help section to troubleshoot basic or known issues related to the workshop

The screenshot shows the 'Help' tab selected in a navigation bar. Below it, a section titled 'Troubleshooting known issue' lists common issues and their quick fixes. One issue is about clipboard access, which is highlighted with a red box.

Common issue	Recommended quick fix
Unable to copy paste	Click on SSL certificate symbol → Open pop-up → change the clipboard dropdown to allow. Once clipboard access is enabled, you can use the following shortcuts to paste inside the VM based on the VM OS:

Attendee OS	VM OS	Copy Shortcut	Pas
Windows	Windows	Ctrl + C	Ctrl

VM native Clipboard

Enable clipboard which helps to copy paste the values from Lab Guide to VM

The screenshot shows the 'VM Native Clipboard' toggle switch being turned on, indicated by a red box. Below it, the same troubleshooting section is shown.

Common issue	Recommended quick fix
Unable to copy paste	Click on SSL certificate symbol → Open pop-up → change the clipboard dropdown to allow. Once clipboard access is enabled, you can use the following shortcuts to paste inside the VM based on the VM OS:

Attendee OS	VM OS	Copy Shortcut	Pas
Windows	Windows	Ctrl + C	Ctrl

Access Lab Now – Alternate method

Access the lab in-case of accidental closure of the browser tab

The screenshot shows a summary of lab details. It includes fields for 'Name' and 'Value' (e.g., deployment_id, vm Admin Password, windows VM DNS Name, github User Email, github User Password), a section for 'Licenses', and a prominent blue 'Access Lab Now' button.

On Demand Lab: GitHub Copilot Innovation Workshop

Please use the below details for future use in your labs:

JumpVM-RG-1487427 :

Name	Value
deployment_id	[REDACTED]
vm Admin Password	[REDACTED]
windows VM DNS Name	[REDACTED]
github User Email	[REDACTED]
github User Password	[REDACTED]

Please use the below Licenses details if required during the lab:

Licence Type	Licence
GCW - GitHub UserName	[REDACTED]
GCW - GitHub Password	[REDACTED]

Access Lab Now

If you have any questions, please contact us at cloudlabs-support@spektrasyystems.com

This email is sent by Spektra Systems LLC, on behalf of Microsoft.
You are receiving this message as you have registered for On Demand Lab at <https://experience.cloudlabs.ai>

Troubleshooting common issues

Get guidance to troubleshoot some of the most common errors you may face while performing your labs.

<https://docs.cloudlabs.ai/troubleshooting/Overview>

The screenshot shows a web page titled "Overview" under the "Common Troubleshooting" section. The page includes a sidebar with navigation links for Introduction, Learner, Instructor, Admin, Lab Requester, MCT Pass Program, and Common Troubleshooting (which is currently selected). The main content area displays a table of common issues and their corresponding guides. At the bottom, there are navigation links for "Previous" and "Next".

Common Troubleshooting

Overview

Common Issues	Guide
Troubleshooting Checklist	Start Here
Fix "Websocket connection blocked" to access HTTPS over RDP	Start Here
Copy & Paste functionality not working in lab environment	Start Here
Unable to launch Cloud Shell	Start Here
Add Safe Sender	Start Here
Login with Azure AD - Consent Experience	Start Here

Previous [« Getting Started](#)

Next [RDP: Known Functionality Issues »](#)

Continue your learning deal-readiness journey with more ...



Microsoft Partner Project Ready

Azure OpenAI Workshop

Duration 4 days (4 hrs/day)



Microsoft Partner Project Ready

Build and modernize AI Apps on Azure

Duration 4 days (4hrs/day)



Explore and Register on LevelUp
skillupwithlevelup.com

Enrich your Agentic AI development skills



Register now
aka.ms/AgenticAIHackathon



Participate in the **live sessions** of this Hackathon on **both days** and **complete 2 of 4 projects** with a **minimum of 70% score** in **each project**.



Receive a Digital Badge as a testament to your engagement and learning.

Register now
aka.ms/AgenticAIHackathon



Welcome to Agentic AI Hackathon

The Agentic AI Hackathon is a hands-on, immersive learning experience focused on building intelligent multi-agent AI applications leveraging Azure AI Foundry, Azure OpenAI services, AutoGen Framework, Semantic Kernel, and GitHub. Participants will gain practical expertise in developing and deploying AI-powered solutions, mastering skills such as AI agent customization, orchestration, and automation. This workshop equips developers with real-world capabilities to innovate and accelerate AI adoption using Azure AI and GitHub tools.

Agentic AI Hackathon

- ⌚ Duration - 2 Days (5 hrs/day)
- 📅 Date - June 10-11, 2025
- 👤 Target Audience - Developers

Workload	Summary	Badge
Opening & Orientation	Introduction about the event	
Choice of Hackathon Projects	Build and Deploy Azure AI Agents Build an End-To-End Multi-Agents Enable Multi-Agent Interactions Using Agentic Framework Capabilities Build a Multi-Agent Presentation Builder	
Final show & Discussion	Closing discussion and Q&A	

Coming up next...

Day 1

Showcasing AI Potential with Agentic AI

Module 1: Innovate with Microsoft 365 Copilot and agents

- M365 Copilot and agents
- How Copilot works
- Semantic index for Copilot
- M365 Copilot Chat
- Agents Use cases
- New agents in Microsoft 365
- Unlock more value with SharePoint agents

Module 2: Extend Microsoft 365 Copilot with Agents

- Microsoft 365 Extensibility Planning and approach
- Declarative agents and agent tooling
- Explore Copilot Studio Agent Builder
- Build declarative agents with Microsoft 365 Agents Toolkit
- Build custom agents with Copilot Studio
- Autonomous agents overview
- Agent Governance - Overview
- Gen-AI decision guide – when to build, buy or extend

Hands-on Labs

- Explore Copilot Studio Agent Builder
- Build HR Assistant Agent with Copilot Studio
- Incorporate actions in HR Agent
- Enable Autonomous Capabilities in Microsoft Copilot Studio for HR Activities

Day 2

Architecting Success with Multi-Agent AI Systems

Module 3: Customize Agents with Gen AI in Copilot Studio

- Customizing your agents – Orchestrator, UI, Knowledge, Actions, Autonomy
- Copilot Studio implementation guidance for architects
- Generative AI in Copilot Studio
- Copilot Studio + Power Platform
- Building voice-enabled agents
- AI Foundry integration
- Developing agents using Microsoft 365 Agents SDK

Module 4: Innovate with Azure AI Platform

- How language models work
- AI Foundry and SDK introduction
- AI Foundry Model Catalog
- Azure AI Services
- Azure OpenAI Service and model guidance
- Models-as-a-Service
- Azure AI Foundry Agent Service
- Safeguard with Trustworthy AI

Hands-on Labs

- Setup AI Project and perform Chat Completion from VS Code
- Build a simple AI Agent
- Develop a multi-agent system

Day 3

Multi-Agent AI: Advanced Agent Dev in Azure AI Foundry

Module 5: Customize, orchestrate and experiment with Azure AI Foundry

- Retrieval Augmented Generation (RAG)
- Customizing models – Fine tuning, distillation
- Responses API (preview)
- Azure AI Foundry Agent Service - Orchestrate and debug AI workflows

Module 6: Build your own multi agents with Semantic Kernel or AutoGen

- Multi-agent applications
- Understanding Semantic Kernel
- Understanding AutoGen Agents Framework
- Multi-Agent Collaboration & Orchestration with AutoGen / Semantic Kernel

Hands-on Labs

- Set Up Azure AI Foundry SDK and Provision Resources
- Build a Retrieval-Augmented Generation(RAG) Pipeline
- Evaluate and Optimize RAG Performance
- Semantic Kernel Fundamentals
- Semantic Kernel Plugins

Day 4

Enterprise Grade: Optimization and production at scale

Module 7: Enterprise grade production at scale

- Scaling challenges and agent controls
- Manage AI performance in production
- Observability Tools
- Enabling Enterprise governance and management
- Enterprise grade security and data protection
- Monitoring and observability

Module 8: Advanced AI risk evaluation and mitigation

- Identifying risks
- Azure AI Content Safety
- Evaluation and GenAIops
- Identity and access management
- Network Security for AI apps
- Continuous security for AI

Hands-on Labs

- Understanding the Lifecycle of Flow Development
- Building and Customizing Prompt Flows
- Evaluation Flow Setup
- Fine-Tuning Prompts for Optimal Performance
- Implementing Chat Flow and Tool Integration
- Ensuring Responsible AI Practices with Content Safety

Journey A – Deal-ready

Deal-ready assessment 

Journey B – Solution Design

Solution Architecture assessment 

Journey C – Project-ready

Capstone Project 

Thank You