

warning "Backend update (October 2026)" GitHub Models inference was retired on July 30, 2026. Any references below are retained only for historical context while this lab is migrated. Use Lab 013 ? Free LLM backends to configure Foundry Local or Ollama before continuing.

## Lab 002: AI Agent Landscape 2025

Level: L50 Path: All paths Time: ~20 min ? Cost: Free ? No account needed

### What You'll Learn

The Microsoft AI agent ecosystem at a glance When to use each platform: Copilot Studio, Microsoft Foundry, Semantic Kernel, Teams AI Library, AutoGen How MCP fits into all of them The spectrum from no-code to pro-code

### Introduction

The Microsoft ecosystem offers multiple overlapping ways to build AI agents. This can be confusing ? should you use Copilot Studio or Foundry? Semantic Kernel or LangChain? MCP or direct API calls?

This lab gives you a map of the landscape so you can make informed choices.

### The Spectrum: No-Code ? Pro-Code

#### No-Code to Pro-Code Spectrum

There's no "better" end ? it depends on your use case, team skills, and governance requirements.

### Platform Comparison

#### ? GitHub Copilot

|| What it is | AI coding assistant embedded in your IDE and GitHub | Best for | Individual developers, coding productivity | Agent capability | Copilot Chat, GitHub Models, Copilot Extensions | Skill needed | Low (chat) to High (extensions) | Cost | Free tier available |

#### ? Copilot Studio (Low-Code)

|| What it is | Microsoft's no-code/low-code agent builder | Best for | Business analysts, M365 users, Teams agents | Agent capability | Topic flows, connectors, custom actions, Azure OpenAI | Skill needed | Low ? no coding required | Cost | Included with many M365 licenses; free trial available |

#### ? Microsoft Foundry Agent Service

|| What it is | Managed agent runtime on Azure | Best for | Production agents, enterprise scale | Agent capability | Tool calling, Code Interpreter, MCP servers, evaluation | Skill needed | Medium ? Python or C# SDK | Cost | Azure subscription (free tier for prototyping) |

#### ? Semantic Kernel

|| What it is | Open-source agent SDK (Python / C# / Java) | Best for | Developers who want code control with Microsoft stack | Agent capability | Plugins, vector memory, planners, multi-agent | Skill needed | Medium ? Python or C# experience | Cost | Free (open-source); LLM costs depend on backend |

#### ?? AutoGen

|| What it is | Open-source multi-agent framework by Microsoft Research | Best for | Complex multi-agent workflows, research, orchestration | Agent capability | Nested conversations, human-in-the-loop, code execution | Skill needed | High ? Python, advanced agent concepts | Cost | Free (open-source); LLM costs |

#### ? Teams AI Library

|| What it is | SDK for building AI-powered Teams bots | Best for | Teams-native apps, enterprise collaboration | Agent capability | Conversational AI inside Teams channels, M365 data access | Skill needed | Medium ? Node.js or C# | Cost | Free SDK; requires M365 tenant

|

question "? Check Your Understanding" What is the key difference between Copilot Studio and Semantic Kernel in terms of who should use them?

??? success "Answer" Copilot Studio is designed for citizen developers and business analysts who need no-code/low-code agent building. Semantic Kernel is designed for professional developers (Python/C#) who want full code control over agent logic, plugins, and memory.

Where Does MCP Fit?

Model Context Protocol (MCP) is not a platform ? it's a connector standard. Think of it as the USB-C of AI tools: one standard interface that any agent can use to plug in to any data source or tool.

Where MCP Fits

MCP works with all the platforms above ? and also with Claude Desktop, OpenAI, and any other MCP-compatible host.

question "? Check Your Understanding" MCP is described as "USB-C for AI tools." What specific problem does this analogy highlight that MCP solves?

??? success "Answer" MCP solves the  $N \times M$  integration problem. Without MCP, connecting 5 agents to 5 tools requires 25 custom integrations. With MCP as a universal standard, each tool publishes one MCP server and every MCP-compatible agent can connect to it ? reducing integrations to  $N+M$ .

Decision Cheat Sheet

Situation | Recommended tool | "I want an agent in Teams for my team, no coding" | Copilot Studio | "I want to add AI to my VS Code extension" | VS Code Chat Participant API | "I want a production agent backed by Azure, with monitoring" | Microsoft Foundry Agent Service | "I want to write Python/C# code to build a sophisticated agent" | Semantic Kernel | "I want multiple AI agents collaborating on complex tasks" | AutoGen | "I want to connect my existing tool/API to any AI agent" | Build an MCP Server | "I just want to experiment with LLMs for free" | GitHub Models |

question "? Check Your Understanding" A developer wants to build a system where a "researcher" agent, a "writer" agent, and a "reviewer" agent collaborate on producing a report. Which Microsoft tool is best suited for this?

??? success "Answer" AutoGen. It is specifically designed for orchestrating multiple specialized agents that collaborate on complex tasks through nested conversations. Semantic Kernel excels at building single sophisticated agents, while AutoGen excels at multi-agent coordination.

? Knowledge Check

question "Q1 (Multiple Choice): You are a citizen developer with no coding experience. You need to build a Teams chatbot that answers HR policy questions from SharePoint. Which tool should you choose?"

- A) AutoGen - B) Semantic Kernel - C) Copilot Studio - D) Microsoft Foundry Agent Service

??? success "? Reveal Answer" Correct: C ? Copilot Studio

Copilot Studio is the no-code/low-code option designed for citizen developers and IT pros. It integrates natively with Teams and Microsoft 365, can point at SharePoint as a knowledge source, and requires zero code. AutoGen and Semantic Kernel require Python/C# development skills. Foundry is for developers building custom backends.

question "Q2 (Multiple Choice): What does MCP (Model Context Protocol) solve in the AI agent ecosystem?"

- A) It provides a GUI builder for agents without coding - B) It optimizes LLM token usage

to reduce API costs - C) It defines a universal standard so any agent can connect to any tool/data source through a common interface - D) It manages authentication and role-based access control for agents

??? success "?" Reveal Answer" Correct: C

MCP is described as "USB-C for AI tools" ? a universal plug standard. Without MCP, connecting 5 agents to 5 tools requires 25 custom integrations. With MCP, each tool publishes one MCP server and every MCP-compatible agent can use it. It solves the N×M integration problem across the entire ecosystem.

question "Q3 (Multiple Choice): What is the primary difference between Semantic Kernel and AutoGen?"

- A) Semantic Kernel is open-source; AutoGen is Microsoft-proprietary - B) Semantic Kernel builds single sophisticated agents with plugins; AutoGen orchestrates multiple specialized agents collaborating on complex tasks - C) AutoGen only works with GPT-4o; Semantic Kernel supports any LLM - D) Semantic Kernel is for Python only; AutoGen supports Python and C#

??? success "?" Reveal Answer" Correct: B

Semantic Kernel excels at building one deeply capable agent ? with plugins, memory, planners, and structured tool use. AutoGen excels at orchestrating \*multiple\* agents ? a researcher agent, a writer agent, a reviewer agent ? each doing a specialized subtask and passing results between them. Both are open-source and support multiple LLMs.

Summary

The Microsoft ecosystem has tools for every skill level and use case? from no-code Copilot Studio to pro-code AutoGen. MCP is the universal connector that works across all of them.

In the next lab, we'll help you choose the right tool for your specific situation.

Next Steps

? Lab 003: Choosing the Right Tool

Phase 3 modernization

warning "Verified 2026-10-27" Use Microsoft Agent Framework 1.0 for new agent work.

Semantic Kernel remains a legacy migration source; AutoGen is maintenance-oriented. Remove obsolete planner patterns and use explicit workflows, tools, memory, checkpoints, and stop conditions.

[diagram or code sample]

Microsoft Agent Framework 1.0 GA AutoGen to Agent Framework migration

Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and every supporting file declared for this lab.

Download generated PDF Download generated ZIP

Supporting file: CHEATSHEET.md

Supporting files: Lab 002: AI Agent Landscape 2025

Start with the documented local or free path. Record assumptions and validation evidence.

Keep hosted services optional unless the lab marks them required.

Supporting file: README.md

Lab 002: AI Agent Landscape 2025

Use these files with the lab guide and generated kit.