

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 003: Choosing the Right Tool

Level: L50 Path: All paths Time: ~15 min ? Cost: Free ? No account needed

What You'll Learn

A practical decision framework for choosing your AI agent tool Understanding the key trade-offs (control vs. simplicity, cost vs. power) Suggested learning routes based on your role and goals

Introduction

After reviewing the landscape in Lab 002, the natural question is: where should I start?

Use the decision flowchart and role-based guides below to find your path.

Decision Flowchart

Decision Flowchart

question "? Check Your Understanding" According to the decision flowchart, what tool should you use if your primary goal is to connect an existing database or API to AI agents?

??? success "Answer" You should build an MCP Server. MCP (Model Context Protocol) provides a universal connector standard so any MCP-compatible AI agent can access your tool or data source through a common interface.

By Role

? Business Analyst / Power User Goal: Automate workflows, create agents without writing code.

Recommended path: 1. Lab 001 ? Lab 011 ? Lab 069 ? Lab 075

Tools: Copilot Studio, Declarative Agents, Power BI Copilot, M365 Copilot

??? Developer (Python / C#) Goal: Write agents in code, integrate with existing systems.

Recommended path: 1. Lab 013 ? Lab 076 ? Lab 020 ? Lab 082 ? Lab 084

Tools: Agent Framework (SK), MCP, Guardrails, GitHub Models

? Integration / Platform Engineer Goal: Expose existing systems (databases, APIs) to AI agents.

Recommended path: 1. Lab 012 ? Lab 020 ? Lab 031 ? Lab 054 ? Lab 064

Tools: MCP, A2A Protocol, pgvector, Azure API Management

?? Solution Architect Goal: Design production multi-agent systems with governance and observability.

Recommended path: 1. Lab 076 ? Lab 049 ? Lab 050 ? Lab 074 ? Lab 084

Tools: Agent Framework, Foundry Agent Service, OpenTelemetry, A2A + MCP

? Data Engineer / Analyst Goal: Build AI-powered analytics, data agents, and enrichment pipelines.

Recommended path: 1. Lab 047 ? Lab 052 ? Lab 053 ? Lab 067 ? Lab 075

Tools: Fabric IQ, Work IQ, GraphRAG, Power BI Copilot

? Enterprise Admin / IT Governance Goal: Govern, secure, and monitor AI agent deployments across the organization.

Recommended path: 1. Lab 063 ? Lab 065 ? Lab 066 ? Lab 064 ? Lab 046

Tools: Entra ID, Purview DSPM, Copilot Studio Governance, APIM, Agent 365

? Student / Learner Goal: Understand AI agents and build something real, for free.

Recommended path: 1. Lab 001 ? Lab 004 ? Lab 013 ? Lab 078 ? Lab 076 ? Lab 022

Tools: GitHub Models, Foundry Local, Agent Framework ? everything free!

question "? Check Your Understanding" A solution architect needs to design a production

multi-agent system with observability and governance. Which combination of tools does this lab recommend?

??? success "Answer" Foundry, Semantic Kernel, AutoGen, and App Insights. The recommended learning path is: Foundry Agent MCP ? Agent Observability ? Multi-Agent SK ? AutoGen Multi-Agent. This covers managed runtime, agent logic, multi-agent orchestration, and monitoring.

question "? Check Your Understanding" What does "more control = more responsibility" mean in the control vs. simplicity trade-off?

??? success "Answer" Pro-code tools like AutoGen and Semantic Kernel give you full flexibility over agent logic, but you must handle more yourself ? error handling, deployment, security, scaling. No-code tools like Copilot Studio are faster to build but less customizable. The right choice depends on your team's skills and requirements.

The Two Key Trade-offs

Control vs Simplicity, Free vs Paid

More control = more flexibility + more responsibility. More simplicity = faster to build + less customizable.

question "? Check Your Understanding" Can a student with no Azure subscription and no budget still build a working AI agent using the tools in this hub?

??? success "Answer" Yes! GitHub Models and Semantic Kernel are completely free. The L50 conceptual labs and L100?L200 labs using GitHub Models require no Azure subscription. Students can build real agents, run MCP servers locally, and learn the full agent development lifecycle at zero cost.

2. Free vs. Paid

The SVG above includes the full Free vs. Paid comparison. Start free ? add Azure only when you need production features.

? Knowledge Check

question "Q1 (Multiple Choice): A developer wants to build a VS Code extension that responds to @mybot in GitHub Copilot Chat. Which tool/API should they use?"

- A) Copilot Studio - B) VS Code Chat Participant API (Lab 025) - C) Microsoft Foundry Agent Service - D) Azure Bot Service

??? success "? Reveal Answer" Correct: B ? VS Code Chat Participant API

The Chat Participant API registers a @yourextension participant directly inside VS Code's Copilot Chat interface. It runs entirely inside VS Code ? no Azure subscription, no server required. Copilot Studio is for Teams/M365 non-code agents. Foundry is for server-side hosted agents with full cloud scale.

question "Q2 (Multiple Choice): Which factor is MOST important when choosing between Copilot Studio and Semantic Kernel?"

- A) The programming language you prefer (Python vs C#) - B) Whether you need cloud deployment or local deployment - C) Your role and how much code control you need ? citizen developer vs. professional developer - D) The LLM provider (OpenAI vs Anthropic)

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

The primary decision axis is code control vs. speed. Copilot Studio targets citizen developers and IT pros who need a functional agent fast with no code. Semantic Kernel targets professional developers who need full control over logic, tool schemas, memory patterns, and production behavior. Both support multiple LLMs and cloud deployment.

question "Q3 (Multiple Choice): The 'least privilege' principle says your agent should have access to exactly what it needs ? no more. Which of these violates least privilege?"

- A) A product search agent that can call search_products() and get_product_details() - B)

A customer service agent given read-only database access - C) An order-status agent given full admin credentials to the orders database - D) A weather agent that can only call the public weather API

??? success "?" Reveal Answer" Correct: C ? Full admin credentials violates least privilege
An order-status agent only needs to *read* order records. Giving it admin credentials means a prompt injection attack or logic error could delete orders, modify prices, or access all customer data. The correct setup is a read-only database user scoped to the specific tables the agent needs. Options A, B, and D all follow least privilege correctly.

Summary

There's no single "right" tool ? it depends on your role, goals, and constraints. The good news: everything in this hub starts free, and you can always level up. The decision framework above points you to the most efficient path.

Next Steps

Pick your path and dive in!

No-code route: ? Lab 011 ? Copilot Studio Developer route (free): ? Lab 013 ? GitHub Models

MCP route: ? Lab 012 ? What is MCP? Full Azure route: ? Lab 030 ? Foundry + MCP Want to understand LLMs first? ? Lab 004 ? How LLMs Work

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 003: Choosing the Right Tool

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 003: Choosing the Right Tool

Use these files with the lab guide and generated kit.