

★ Member-only story

Microsoft Just Solved the AI Agent Problem (And Open-Sourced the Solution)

6 min read · Oct 14, 2025



Micheal Lanham

Following ▾

▶ Listen

📄 Share

⋮ More



all images generated with gpt-image-1 and agents

How the new Agent Framework unifies Semantic Kernel and AutoGen to bring enterprise-grade AI agents from prototype to production

You're building an AI agent. You start experimenting with AutoGen because the multi-agent orchestration is incredible. Everything works beautifully in your demo.

Then your boss asks: "Can we put this in production?"

Suddenly you're scrambling. Where's the monitoring? How do you handle authentication? What about compliance? You realize you need Semantic Kernel's enterprise features, but now you're rewriting everything.

Sound familiar?

Microsoft just ended this nightmare. They've released an **open-source Agent Framework** that finally gives us both worlds: AutoGen's experimental freedom *and* Semantic Kernel's production reliability. No more choosing. No more rewriting.

Let's dive into what makes this framework a game-changer.



The Problem: Innovation vs. Reliability

Here's the dilemma every AI developer has faced lately.

Semantic Kernel gave you stability. Enterprise connectors, robust memory management, content filtering, telemetry — all the boring stuff that actually matters

in production. It was the responsible choice.

AutoGen gave you magic. Multi-agent systems where AI agents debate, reflect, and collaborate in group chats. It was the exciting choice that made demos pop.

Each had passionate followers. Each had critical gaps.

Developers kept asking Microsoft: *“Why can’t we have both?”*

Now we can.

One Framework to Rule Them All

Microsoft’s Agent Framework merges these two philosophies into a single, unified toolkit.

Think of it like this: you get AutoGen’s Ferrari engine with Semantic Kernel’s Volvo safety features. Speed *and* reliability in one package.

What does this mean in practice?

You can now build AutoGen-style multi-agent systems (with all those cool collaboration patterns) while getting Semantic Kernel's thread-based state management, type safety, and plugin architecture. Your experimental agent behaviors run on a robust, controllable backbone.

Best part? It's **100% open source** on GitHub with an MIT license.

Getting Started Is Almost Too Easy

Microsoft promises you can create your first AI agent with “just a few lines of code.”

Let’s test that claim:

```
# Install the framework
pip install agent-framework

# Create a simple agent
from agent_framework import Agent

agent = Agent(
    name="ResearchAgent",
    model="gpt-4",
    instructions="Help users research topics thoroughly"
)

# Run it
response = agent.run("What are the latest developments in quantum computing?")
```

That’s it. You’re up and running locally in minutes.

But here's where it gets interesting: the framework **scales with you**. Start with one agent. Add multi-agent workflows. Integrate tools. Add monitoring. Deploy to Azure. All without switching frameworks or rewriting code.

No more "this is just a prototype" disclaimers in your demos.

Python and .NET: Choose Your Weapon

Most AI frameworks force you to pick a language. Python or nothing.

Not this one.

The Agent Framework supports **both Python and .NET** with full feature parity. Python developers get their familiar `pip install`. .NET developers get a NuGet package (`Microsoft.Agents.AI`).

This is huge for enterprise teams where different groups use different languages. Your ML engineers can prototype in Python while your backend team builds in C#, all using the same framework.

Workflows: AI Meets Infrastructure as Code

Here's my favorite feature: **declarative workflows**.

You can define entire multi-agent systems using YAML or JSON. Yes, really.

```
workflow:
  name: "ResearchWorkflow"
  agents:
    - name: "Researcher"
      role: "Gather information"
      handoff_to: "Analyst"

    - name: "Analyst"
      role: "Summarize findings"
      handoff_to: "Reporter"

    - name: "Reporter"
      role: "Draft final report"
      requires_approval: true
```

Why is this brilliant?

Version control. Your AI workflow is now code. Commit it. Review it. Roll it back.

Visualization. Microsoft provides a Visual Studio Code extension that shows your agents as nodes in a graph. You can literally *see* how your agents interact.

State management. Workflows support checkpointing (save and resume), human-in-the-loop interactions (pause for approval), and persistent state for long-running processes.

This bridges the gap between “autonomous AI” and “deterministic business process.” You get AI flexibility within a structured, auditable flow.

Enterprise Features That Actually Matter

Let's talk about the unsexy stuff that keeps CIOs awake at night.

Observability: Know What Your Agents Are Doing

Every agent action is traced via **OpenTelemetry** and integrates with Azure Monitor.

Model calls, tool invocations, messages between agents — all logged and traceable. Microsoft even contributed OpenTelemetry enhancements specifically for multi-agent systems.

You can now treat your AI agents like any microservice. That's a big deal.

Security: Because “We’ll Figure It Out Later” Isn’t a Strategy

The framework includes:

- **Entra ID integration** for authentication
- **Access control** for enterprise APIs
- **Human-in-the-loop checkpoints** for sensitive operations
- **Responsible AI safeguards** like prompt filtering

Your agents can securely access company data while respecting identity and policy rules.

Open Standards: No Vendor Lock-In

The framework supports:

- **OpenAPI** for tool plugins (integrate any REST API)
- **Model Context Protocol (MCP)** for dynamic tool integration
- **Agent-to-Agent (A2A)** messaging for inter-agent communication

Your agents aren't locked into Microsoft's ecosystem. They can talk to anything.

Real Companies, Real Results

This isn't vaporware. Organizations are already using it.

KPMG built “KPMG Clara AI” on the Agent Framework, connecting specialized audit agents to enterprise data. In a regulated industry where governance is everything, the framework's built-in compliance features were non-negotiable.

Commerzbank is piloting AI agents for customer support. Their take? It “simplifies coding, reduces effort, and fully supports MCP for agentic solutions.”

BMW, Fujitsu, Citrix, and TeamViewer are exploring use cases from vehicle telemetry analysis to IT automation.

Early signs suggest this could become the de facto standard for enterprise AI agents.

From Prototype to Production in One Framework

Here's the bottom line.

Microsoft's Agent Framework solves the fundamental tension in AI development: innovation versus production-readiness.

You no longer choose between AutoGen's experimental flexibility and Semantic Kernel's enterprise stability. You get both. In one framework. Open source.

Whether you're a solo ML engineer building your first agent or an enterprise team deploying regulated AI systems, you now have a toolkit that scales from "Hello World" to mission-critical production.

The framework provides:

- **Multi-agent orchestration** (group chats, debates, handoffs)
- **Declarative workflows** (YAML/JSON definitions)
- **Cross-language support** (Python and .NET)
- **Enterprise observability** (OpenTelemetry + Azure Monitor)

- **Security & governance** (Entra ID, human-in-the-loop, compliance)
- **Open standards** (OpenAPI, MCP, A2A protocols)
- **Visual tooling** (VS Code extension, Azure AI Foundry)

All open source. All under one roof.

Your Move

The Agent Framework is available now on GitHub. MIT licensed. Ready to use.

Start with a simple agent locally. Add complexity as you need it. Deploy to production when you're ready. All without switching tools.

Eight in ten enterprises now use agent-based AI. The question isn't whether your organization will build AI agents. It's whether you'll build them on a fragmented stack of tools or a unified, production-ready framework.

Microsoft just gave you the answer.

What will you build?

. . .

Want to dive deeper? Check out the [Agent Framework documentation](#) or explore the [GitHub repository](#). And if you're building something cool with AI agents, I'd love to hear about it in the comments.

Tags: #ArtificialIntelligence #MachineLearning #Microsoft #OpenSource #AgenticAI #Python #DotNet #EnterpriseAI

Micheal Lanham

Ai Agents In Action

Microsoft Ai Agents

Agent Framework

Microsoft Ai



Following ▾

Written by Micheal Lanham

672 followers · 5 following

Micheal Lanham is a proven software and tech innovator with 20 years of experience developing games, graphics and machine learning AI apps.

Responses (3)



Bgerby

What are your thoughts?



Seth Chandler

2 days ago



I hate when people post stuff on Medium that does not work. Like this. ImportError Traceback (most recent call last)

```
/tmp/ipython-input-2063330466.py in <cell line: 0>()
```

```
----> 1 from agent_framework import Agent
```

```
ImportError: cannot import name 'Agent'... more
```



1

[Reply](#)



Sudhakar Varadharaj

2 days ago



Do not put the untested code in the blog. The above sample agent_framework code not working. Looks like AI generated code or concept.



1

[Reply](#)



ddmn

5 days ago



Tried if a simple use case, potential is there with the framework but how it expands purely depends on the community stability.

<https://python.plainenglish.io/building-smart-conversational-applications-with-microsoft-agent-framework-a-complete-walk-through-c99767d4f0b9>



1

[Reply](#)

More from Micheal Lanham



Micheal Lanham

The 10 AI Agent Platforms Revolutionizing Software in 2025

A hands-on guide to choosing the right framework for building autonomous AI agents—from open-source powerhouses to enterprise solutions



Oct 8




79



3




 Micheal Lanham

AI Agents Just Went Enterprise: October 2025 Changed Everything

How Microsoft, OpenAI, Google, and AWS turned “agentic AI” from GitHub experiments into products your company can actually buy

✦ Oct 13 🖱 22 💬 2




 Micheal Lanham

Building Intelligent RAG Systems with AI Agents: 25 Game-Changing Tips

Your complete guide to creating production-ready Retrieval-Augmented Generation applications that actually work

★ Sep 24 🖱 34



 Micheal Lanham

Building Production-Ready RAG Agents: A Developer's Complete Guide


From keyword search to vector embeddings—everything you need to know about implementing Retrieval-Augmented Generation that actually...

★ Jun 23 🖱 28



See all from Micheal Lanham

Recommended from Medium

 In Towards AI by Teja Kusireddy

We Spent \$47,000 Running AI Agents in Production. Here's What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic's Model Context Protocol (MCP) are revolutionary. But...

Oct 16  995  15




 In Dare To Be Better by Max Petrusenko

Claude Skills: The \$3 Automation Secret That's Making Enterprise Teams Look Like Wizards

How a simple folder is replacing \$50K consultants and saving companies literal days of work

✦ Oct 17 🖱 398 💬 6



 In Coding Nexus by Code Coup

Claude Desktop Might Be the Most Useful Free Tool You'll Install This Year

I didn't expect much when I first saw the announcement for Claude Desktop. Another AI wrapper, I thought. Maybe with a shiny UI.

✦ 6d ago 🖱 246 💬 6





In AI Software Engineer by Joe Njenga

This Viral DeepSeek OCR Model Is Changing How LLMs Work

This DeepSeek OCR model hit an overnight success not seen in any other release—4k+ GitHub stars in less than 24 hours and more than 100k...



Oct 23



272



4



In Artificial Intelligence in Plain English by Surendra Pandar

Every Paid AI—Now FREE & UNLIMITED (100% Legal)

The founder raised \$5.7M to make this possible



Oct 21




47



4



 Agen.cy

20+ Genius Ways Power Users Are Using Claude Code Right Now

Here are 10+ ways power users are using Claude Code 

Oct 23  29  1



See more recommendations