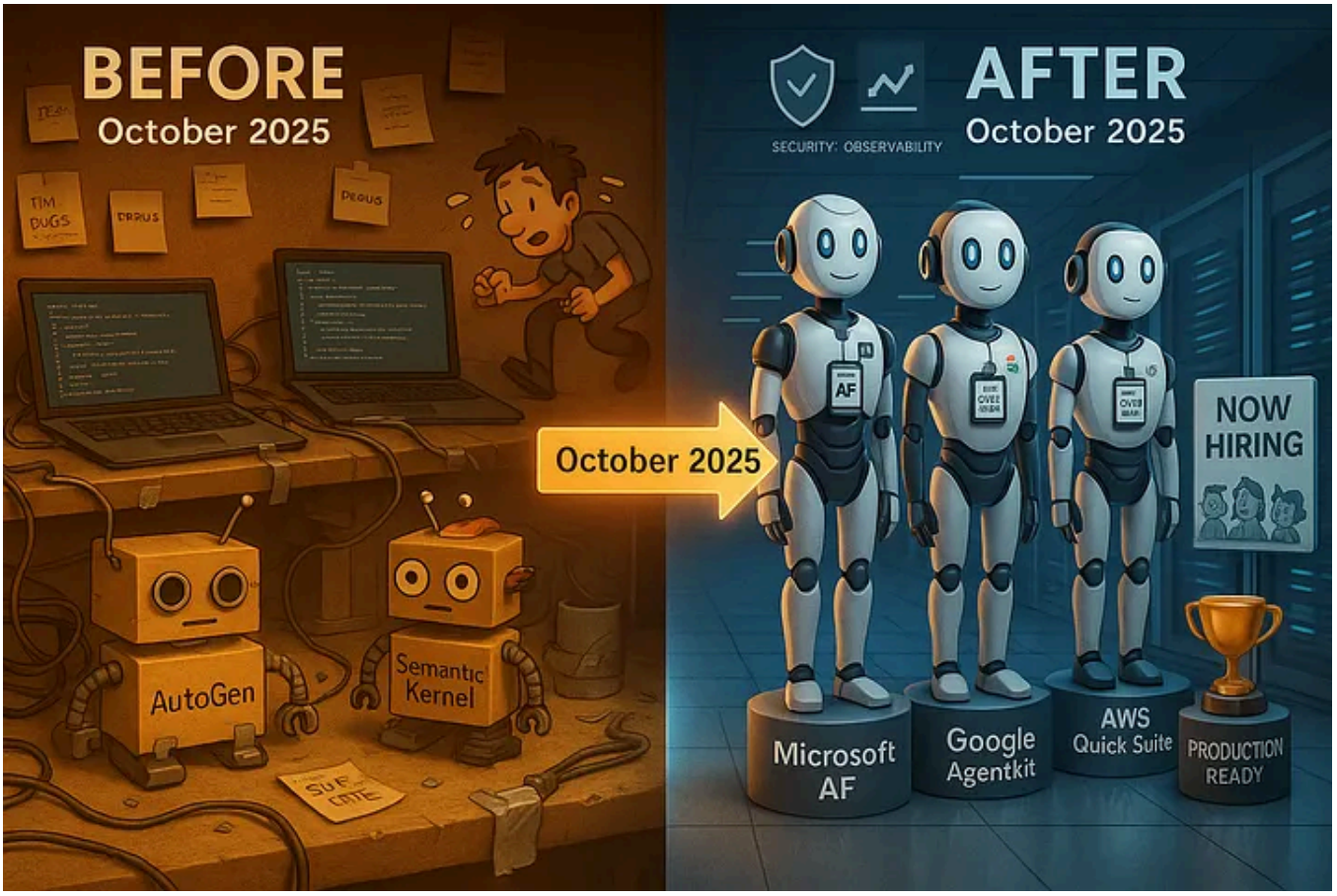
Member-only story

AI Agents Just Went Enterprise: October 2025 Changed Everything

9 min read · Oct 13, 2025

Micheal Lanham Following

Listen Share More



all images generated with gpt-image-1 and agents

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

I've been building with AI agents for the past 30 months, and I'll be honest — it's been messy

You'd cobble together AutoGen for orchestration, add Semantic Kernel for memory, figure out your own observability stack, and pray nothing broke in production. Every demo looked magical. Every production deployment felt like duct tape and hope.

That just changed.

October 2025 will be remembered as the month AI agents stopped being side projects and became enterprise software. In the span of two weeks, Microsoft open-sourced a unified agent framework, OpenAI launched a complete agent development platform, and both Google and AWS shipped products that let you buy AI agents the way you buy Salesforce licenses.

If you've been waiting for the signal to move from proof-of-concept to production, this is it.

The Shift: From "Can We Build This?" to "Which One Should We Buy?"

Remember when building a web app meant hand-rolling your own authentication, database la press gave us the buil th agents.

Here's what dropped in October:

🔷 **Microsoft: Agent Framework (Open Source)**

A unified SDK merging AutoGen's multi-agent magic with Semantic Kernel's production readiness.

🟢 **OpenAI: AgentKit + Apps in ChatGPT**

A complete toolkit from design canvas to embedded UI components — plus a way to ship agents as ChatGPT apps.

🔴 **Google: Gemini Enterprise**

A $30/user/month "front door" for AI at work, with no-code agent building and Computer Use.

🟠 **AWS: Amazon Quick Suite**

The Swiss Army knife of agent platforms at $20–40/user/month, embracing open standards everywhere.

Let me break down what each means for you.

Microsoft Agent Framework: The Dev-First Play

Microsoft basically said: “You know how you’ve been choosing between AutoGen for research and Semantic Kernel for production? Stop choosing.”

Agent Framework (AF) is their answer — an open-source SDK that gives you graph-based workflows, type-safe messaging, and first-class observability from day one. It’s MIT licensed and supports both Python and .NET.

What makes AF different?

Built-in production features that you usually add later (and painfully):

- OpenTelemetry traces for debugging
- Human approval gates for risky actions
- Durable checkpointing so agents can recover
- Thread-based state management

Open standards everywhere:

- MCP (Model Context Protocol) for tool connections
- A2A (Agent-to-Agent) for cross-platform collaboration
- OpenAPI for importing any REST API as a tool

Version control for agents: Agents are defined in YAML/JSON, so you can version them in Gi

When to use

You're a developer who wants fine-grained control and doesn't want to be locked into a proprietary platform. You're comfortable with code and want to build custom multi-agent systems that can run anywhere — locally, on Azure, or eventually ported to another cloud.

```
# Example: Defining an agent in Microsoft AF
from agent_framework import Agent, Workflow

research_agent = Agent(
    name="research_assistant",
    instructions="Find and synthesize technical documentation",
    tools=["web_search", "mcp://documentation-server"]
)

writer_agent = Agent(
    name="content_writer",
    instructions="Create clear technical content from research",
    tools=["markdown_formatter"]
)

workflow = Workflow(
    agents=[research_agent, writer_agent],
    handoffs=[("research_assistant", "content_writer")]
```

OpenAI AgentKit: From Canvas to Shipped Product

If Microsoft's bet is on developers who want control, OpenAI's bet is on teams who want velocity.

AgentKit is a cohesive toolkit that handles the entire lifecycle:

1. Agent Builder (Visual Canvas)

Design multi-agent workflows without writing orchestration code. Think of it as Figma for agent logic — drag nodes, connect them, test runs in preview mode.

2. ChatKit (Embeddable UI)

Drop-in React components that give your product a ChatGPT-quality interface. No more building chat UIs from scratch or explaining why yours feels clunky compared to ChatGPT.

3. Connector Registry

A centralized admin panel to manage data sources and tools across your agents. Yes, it supports third-party MCP servers.

4. Evals for Agents

This is huge. Automated evaluation with trace-level grading and prompt optimization. You can finally answer "did this agent actually improve?" with data.

Apps in ChatGPT: Agents as Products

Here's where it gets interesting. OpenAI turned ChatGPT into a **platform**.

You can now build "apps" that live inside ChatGPT using the new Apps SDK (based on MCP). Users call them by name mid-conversation:

> "Hey ChatGPT, use Expedia to find me flights to Tokyo next month"

Day-one partners include Booking.com, Canva, Coursera, Figma, Spotify, and Zillow. If you're building B2C agent experiences, you can piggyback on ChatGPT's distribution instead of fighting for user attention.

When to use AgentKit

You want to ship a polished agent product quickly. You're okay with being in OpenAI's e mponents, and battle-

The Platform Battle: Google vs AWS

Both Google and AWS launched complete agent platforms this month. Not frameworks — **products** with pricing, admin consoles, and SLAs.

Google Gemini Enterprise ($30/user/month)

Google's pitch: **"One front door for AI at work"**

What you get:

- A no-code workbench for building task-specific agents
- Deep in
- Computer Use capabilities (browser-level automation)
- Partner agents you can instantly deploy
- Connections to external systems like Salesforce

Who it's for: Organizations already in the Google ecosystem who want governed AI distribution to employees without custom dev work.

Amazon Quick Suite ($20–40/user/month)

AWS came in with a broader, more modular answer:

Quick Index & Spaces → Connect data sources + create team workspaces
Quick Research → Deep research with web + internal sources
Quick Sight → Agentic business intelligence
Quick Flows → Routine automations
Quick Automate → Complex multi-system workflows

Plus browser extensions and add-ins for Outlook, Teams, and Word.

Why AWS matters here: They're all-in on **open standards.** Quick Suite connects to thousands of tools via OpenAPI and MCP servers. It's the least locked-in platform

option.

Who it's fo strong automation primitives and don't want vendor lock-in.

Decision Framework: Which Platform Should You Choose?

Here's my honest take after digging through the docs and pricing:

Choose Microsoft Agent Framework if:

✅ You're a developer-first team
✅ You need fine-grained control over agent behavior
✅ You want to avoid platform lock-in
✅ You're comfortable with Python or .NET
✅ You need to run agents locally or across multiple clouds

Choose OpenAI AgentKit if:

✅ You want to ship fast
✅ You're building a product with agent UX
✅ You need visual design tools for non-developers
✅ You want ChatGPT distribution via Apps
✅ Evals and quality measurement matter to you

Choose Google Gemini Enterprise if:

✅ You're a
✅ You wa
✅ Computer Use (browser automation) is valuable
✅ You need deep Google ecosystem integration

Choose AWS Quick Suite if:

✅ You want a complete agent platform with automation
✅ You need to connect to many third-party systems
✅ Open standards (MCP/OpenAPI) are non-negotiable
✅ You want aggressive per-seat pricing
✅ Browser extensions and Office add-ins are important

The Standards That Matter: MCP and OpenAPI

Here's something crucial that happened this month: **every major launch highlighted the same two standards.**

MCP (Model Context Protocol)

A universal way for agents to access tools and data sources. Think of it as USB-C for AI agents — one protocol, any tool.

OpenAPI

The REST API standard that lets agents dynamically discover and use thousands of services.

Why this matters: If you hard-code custom connectors for each tool, you're building technical debt. The industry just standardized on MCP + OpenAPI.

```
// Example: Adding an MCP server to an agent
{
  "name": "customer_support_agent",
  "tools": [
    "mcp://zendesk-connector",
    "mcp://salesforce-connector",
    "https://api.stripe.com/openapi.json"
  ]
}
```

When every platform speaks the same language, you can:

- Switch between frameworks without rebuilding connectors
- Add ne
- Share tool configurations across teams

Security Finally Gets Serious

With great autonomy comes great responsibility (and liability).

The **AI Agent Security Summit** in NYC and SF focused on what enterprises actually need:

Permissioning: Not every agent should access every system
Monitoring: Real-time observability of what agents are doing
Prompt injection defense: Protecting against adversarial inputs
Human-in-the-loop: Approval gates for high-stakes actions

IBM + Anthropic also dropped a partnership with a guide for secure enterprise agents, covering governance, auditing, and safe MCP integrations.

What this means for builders

If you're deploying agents in production, you need:

1. **Observability from day one** (OpenTelemetry traces, not just logs)
2. **Approv**
3. **Permission boundaries** (principle of least privilege)
4. **Audit trails** for compliance

The platforms are building this in. If you're using a framework, you'll need to add it yourself.

The Coding Agent Evolution: GPT-5 Codex

One more thing that shipped: **GPT-5-Codex** with dynamic "thinking time."

The model adjusts how long it thinks based on task complexity:

- Quick fixes → snappy responses
- Complex refactors → sustained reasoning (hours if needed)

This normalizes the idea of agents as **autonomous teammates**, not just autocomplete. You can delegate a feature, walk away, and come back to a working implementation.

What You Should Do Monday Morning

If you're exploring (not committed yet):

1. **Start with Agent Framework** locally (it's free and open source)
2. **Try OpenAI's Agent Builder**
3. **Compare pricing**: Quick Suite at $20/user vs Gemini at $30/user

If you're building a product:

1. Use **AgentKit** for rapid development
2. Build on **MCP and OpenAPI** from day one
3. Plan for **evals and observability** before launch

If you're buying for your company:

1. **Google Gemini** if you're on Workspace
2. **AWS Quick Suite** for broad integrations
3. **Microsoft AF** if you have engineering resources and want control

The Bigger Picture: Agents Become the UI

What really changed this month isn't just that more companies launched agent products.

It's that you can now:

- **Design** agent workflows visually
- **Connect** them to thousands of tools via open standards
- **Ship** them to users with governed UIs and built-in security
- **Measure** their performance with automated evals

The pieces finally fit together.

If you've been building POCs that never make it to production, that friction just dropped significantly.

. . .

One Final Thought

Remember the early days of cloud computing? AWS, Google Cloud, and Azure all launched their own takes on VMs, storage, and databases. Eventually standards emerged (Kubernetes, Terraform, etc.) and the ecosystem stabilized.

We're in the AWS-EC2 era of AI agents right now. Microsoft, OpenAI, Google, and AWS are all making their bets on what the primitives should be.

The good news? They're all betting on **open standards** (MCP, OpenAPI, OpenTelemetry).

That means we get the best of both worlds: platform features without lock-in.

. . .

What are you building with agents? I'd love to hear about your use cases and which platform you're considering. Drop a comment or connect with me — I'm tracking this space closely and happy to share notes.

If this helped clarify the landscape, give it a clap 👏 — and follow for more deep dives on wh

. . .

Resources

📚 **Docs & Repos:**

- Microsoft Agent Framework on GitHub
- OpenAI AgentKit
- Google Gemini Enterprise
- AWS Quick Suite

🔒 **Security:**

- AI Agent Security Summit
- IBM + Anthropic Partnership

. . .

Want a custom comparison for your tech stack? Let me know your cloud provider, identity system, and key use cases and I'll help you navigate which platform makes sense.

Micheal Lanham | Ai Agents In Action | Microsoft Agents | Agentic Ai | Aws Ai Agents

Written by Micheal Lanham

673 followers · 5 following

Micheal Lanham is a proven software and tech innovator with 20 years of experience developing games, graphics and

Responses (2)

Bgerby

What are your thoughts?

Lonnell branch
5 days ago

Great visuals. They really do help understand the differences between the product offerings.

Reply

Terry Yodaiken
Oct 14

Great summary and really helpful. I'm building a AI powered resume builder that does extensive company and interviewer research, then builds out your resume (plus cover letter when needed) based on your factual experience, customised for the role... more

Reply

More from Micheal Lanham

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

Micheal Lanham

Building Smarter AI Agents: 7 Game-Changing RAG Strategies That Actually Work

The insider secrets ML engineers use to create AI agents that don't hallucinate and actually know what they're talking about

Sep 14 97 2

Micheal Lanham

AI Agents

A practical f ds an autonomous AI genius

Oct 21 16 1

See all from Micheal Lanham

Recommended from Medium

In Towards AI by Gao Dalie (*高達烈*)

RAG is Not Dead! No Chunking, No Vectors, Just Vectorless to Get the Higher Accuracy

Over the past two years, I have written numerous articles on how Retrieval-Augmented Generation has become a standard feature in nearly all...

Oct 17 730 6

In Data Science Collective by Simon Greenman

The AI Vibe Coding Paradox: Why Experience Matters More Than Ever

AI can now c[illegible] faster than any developer. But without supervision and [illegible] breaks down in record tim[illegible]

4d ago 17[illegible]

In Dare To Be Better by Max Petrusenko

Unleash Your Inner Wizard: Claude Skills Are Automating Enterprise for Pennies

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

Oct 17 399 6

In Towards Deep Learning by Sumit Pandey

Why Ever ne

I just saw thi long time. (No, it's not Elon.) It's that tiny...

Oct 14 70 11

In AI Software Engineer by Joe Njenga

Why Claude Weekly Limits Are Making Everyone Angry (And $100/Month Plan Will Not Save You)

Yesterday, I finally hit my weekly Claude limit, and I wasn't surprised, since I see dozens of other users online going crazy over these...

Oct 19 130 25

Micheal Lanham

October 2025: The AI Breakthroughs That Changed Everything

Small models beating giants, AI agents taking control, and the security wake-up call that shook the industry

Oct 16 2

See more recommendations