



Multi-Tenant Copilot Studio Agents for MSPs

We Built an Agent. Here's What Worked, What Broke
and How to Govern It.

pax8.com

Executive Summary

Microsoft Copilot Studio now supports multi-tenant agents in public preview. In plain terms, one AI agent, built and hosted inside your Microsoft 365 environment, can be installed and used by any of your customers' tenants — without any Microsoft Power Platform setup required on their side. For MSPs, this is the first time the platform makes a “build once, serve many” agent strategy possible inside the Microsoft 365 stack.

We built an agent end-to-end to find out how real the capability is. It worked, but the path is narrower than the marketing suggests. Turning a working agent into a sustainable offering means accepting responsibilities that look a lot more like running a software company than running a managed service.

There's also a second shift happening at the same time. Agent 365, Microsoft's governance layer for AI agents, became generally available on May 1, 2026. It gives every agent its own identity inside Microsoft Entra, treats agents as first-class digital workers alongside employees and gives MSPs the control plane needed to run agents responsibly at scale. For any MSP thinking about multi-tenant agents, Agent 365 isn't optional. It's the governance foundation underneath the offering.

This document walks through what we built, the four classes of problems we hit, the single configuration choice that unlocks cross-tenant access, what changes when you become the host and where Agent 365 fits in your governance story.

The Build:

One Agent, Two Tenants and One Source of Truth

We deployed a Copilot Studio agent — codenamed Poseidon — as a finance advisor that answers questions about a curated set of financing institutions. The agent is locked to a single, controlled data source and is configured to never use the open web or make up answers it can't ground. Customers reach the agent through Microsoft Teams in their own tenant, exactly as they would any other Teams app.

The deployment was a three-step process: create a Microsoft identity for the agent that's allowed to operate across tenants, build and configure the agent in Copilot Studio and publish it as a Teams app — first into our tenant, then into the external customer tenant.

The final result is what truly matters. We wanted a user from our organization and a user from a different company to both interact with the same agent through their own Teams environment, receive identical responses based on the same data source and return later to continue their conversations seamlessly. This setup provides genuine cross-tenant functionality rather than just a demonstration.

The Four Things That Broke

A clean walkthrough of Microsoft's docs makes this look like a wizard you click through. It isn't. We hit four distinct problems, each of which would stall an MSP rolling this out at a customer site if the team isn't ready for it.

1. A Back-End Data Issue We Couldn't Fix from the Front End

The agent threw an error during testing that turned out to be a bad record deep in Microsoft's platform, somewhere we couldn't see or touch. After exhausting reasonable troubleshooting, the fastest path forward was to rebuild the agent from scratch and reuse the same identity.

Lesson: When something breaks below the surface in Copilot Studio, expect to rebuild rather than debug. Build that into your time estimates.

2. The Teams App Package Wouldn't Pass Microsoft's Own Validator

The file Copilot Studio generated used an older manifest file that Teams Admin Center now rejects. We had to make a small manual edit to this file before Teams would accept it.

Lesson: Microsoft's tooling is moving fast, and the pieces don't always agree with each other. Keep a deployment runbook with the workarounds, because new ones will appear.

Note: Issues like this are characteristic of a public preview, where the tooling is still catching up to the platform. We expect the rough edges around package generation, validation and tenant onboarding to smooth out as multi-tenant Copilot Studio moves into general availability.

3. External Users Got Stuck in a Sign-In Loop

When customers tried to log in for the first time, the sign-in screen wouldn't render properly and they were dead-ended. The fix was a configuration change on the agent's identity.

Lesson: The authentication setup has very little tolerance for "almost right." A single missed setting and your customer's experience is broken — with no useful error message to point them at the cause.

4. The Agent Kept Failing to Find Data That Was Clearly There

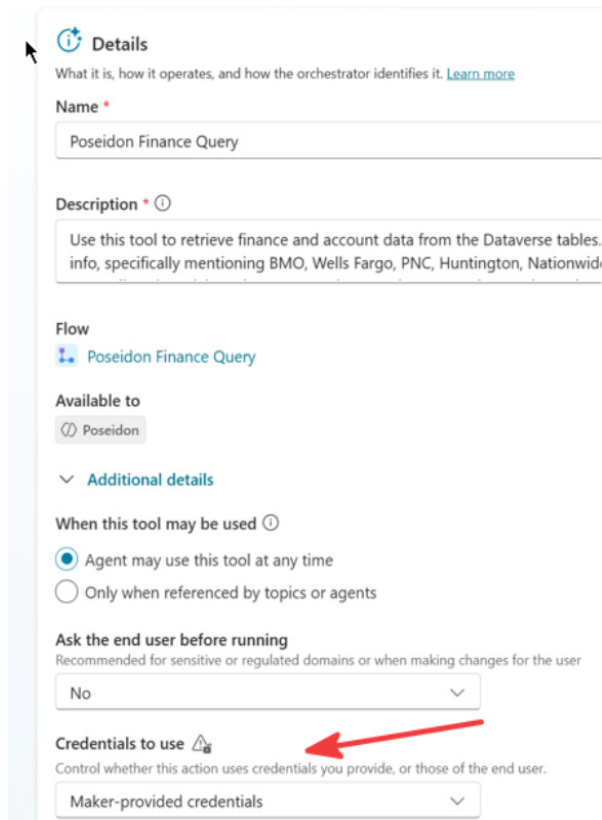
It was sending the user's full sentence into the data lookup as if the whole sentence were a search term, instead of pulling out the company name and looking that up. We fixed it by tightening the agent's instructions and making the lookup more forgiving.

Lesson: The connection between the AI and the underlying data needs to be tuned as a system, not as separate pieces. Plan time for it.

None of these are showstoppers. They're the kind of thing that turns a half-day deployment into a half-week if you hit them at a customer site.

The One Thing That Matters Most:

How the Agent Signs Users In



Details
What it is, how it operates, and how the orchestrator identifies it. [Learn more](#)

Name *
Poseidon Finance Query

Description * ⓘ
Use this tool to retrieve finance and account data from the Dataverse tables. info, specifically mentioning BMO, Wells Fargo, PNC, Huntington, Nationwid

Flow
Poseidon Finance Query

Available to
Poseidon

▼ **Additional details**

When this tool may be used ⓘ
☒ Agent may use this tool at any time
☐ Only when referenced by topics or agents

Ask the end user before running
Recommended for sensitive or regulated domains or when making changes for the user
No

Credentials to use ⓘ
Control whether this action uses credentials you provide, or those of the end user.
Maker-provided credentials

If you take only one lesson from the build, take this one. Copilot Studio offers two ways to handle sign-in for an agent, and only one of them works across tenants.

The default option — the easy, recommended-looking one — works perfectly for users in your own tenant and silently blocks every external user. It is not a permission you can grant your way around. It's a hard limit on how that option is built.

The working option requires a more deliberate setup that explicitly tells Microsoft this agent is meant to be used by people from other organizations. It's more configuration up front, but once it's in place, the customer experience is clean. They receive a one-time sign-in prompt the first time they use the agent, and from then on, they just chat with it like any other Teams app.

There is one caveat that MSPs should internalize before they sell anything. In the current preview, the agent acts under your identity, not the customer's. That's a clean security boundary — your agent can't accidentally reach into a customer's mailbox or files — but it also means the agent can't personalize answers based on who the user is or what they have access to in their own tenant. Microsoft has flagged user-level authentication on the roadmap, so until it ships, you should design around it.

What This Looks Like at Scale:

You're Not Running a Managed Service Anymore

Cross-tenant deployment is the easy part to demo. The harder conversation is what changes when you go from a proof of concept to a real offering. Microsoft has effectively handed MSPs the technical capability of a software vendor without handing over any of the supporting infrastructure. Everything below sits on you, the host.

Cost. All of it.

Every conversation in every customer's tenant runs on your Microsoft 365 capacity and your bill. There is no built-in mechanism to pass usage costs through to the customer. If your capacity runs out, every customer breaks at once. Any commercial recoup has to be structured in your contracts, such as bundled into a managed service, an add-on subscription or a fixed fee, because Microsoft isn't billing your customers for you.

Release discipline you may not have today.

Microsoft's recommended setup includes a second test environment in a separate tenant, because cross-tenant behavior is what you're shipping and you can't validate it from inside your own walls. That doubles your test footprint and adds a step to every release.

Terms, privacy and support are your problem.

Publishing a multi-tenant agent requires you to provide your own terms and conditions, privacy policy and supporting documentation. Customers have no visibility into the back end, which means every support ticket comes to you, not Microsoft.

Data residency is your problem too.

In the preview, an agent lives in a single geography and there's no way to host it in multiple regions. If a customer has a data residency requirement that doesn't match where your agent is hosted, the answer right now is "not yet."

Adoption can outrun you.

If the agent reaches a broad audience, growth is unpredictable, and unpredictable growth means unpredictable costs. The defensive moves are simple but non-negotiable: distribute privately to known customers at first, restrict which tenants are allowed to reach the agent, build usage limits into your contracts and put alerts on your capacity before you onboard the second customer — not the tenth.

That covers the commercial and operational shifts. The governance question deserves its own section.

Who Runs Your Agents? Agent 365 and the Governance Question

The moment you ship a multi-tenant agent, you have a new kind of thing in your environment: a piece of software that acts in the world, that has access to data, that talks to people and that, until very recently, had no proper place in your identity directory. Multiply that across every customer who has Copilot Studio, every business user who decides to build an agent for their team and every ISV pushing agents through the Teams Store, and you arrive at the problem that's been quietly emerging across Microsoft 365 environments — agent sprawl.

Three questions surface quickly and most organizations can't answer them today:

Who built it?

Agents are being created across the organization with no central visibility.

What can it access?

Agents inherit permissions without formal access controls.

Who is accountable?

No clear ownership model exists when an agent acts on its own.

Agent 365, generally available since May 1, 2026, is Microsoft's answer. It sits alongside the tools you already use to build agents (Azure AI Foundry for pro-code, Copilot Studio for low-code) and serves a different purpose. It's where you govern and operate them. In the same way that Entra is the control plane for users, Agent 365 is the control panel for agents.

The Headline Change:

Every Agent Gets an Identity

Agent 365 makes each agent a first-class digital worker inside Microsoft Entra, which means:

- **A directory entry.** Your agent appears in Entra alongside employees and service principals — not hidden inside a Power Platform environment where only the maker can see it.
- **A mailbox and Teams presence.** Agents can send email, participate in Teams conversations and be contacted directly. They show up where work already happens.
- **A place in the org chart.** You can see who an agent reports to and where it fits in your structure.
- **One identity per agent.** No shared accounts, no shadow identities, no agents running under a person's credentials with no clean way to revoke when the person leaves.

For an MSP shipping a multi-tenant agent, that last point matters more than it might sound. The agent you're hosting is now a named, identifiable entity in your directory and is auditable, governable and revocable, like any other digital worker. That's a meaningful step up from the current preview reality where the agent runs under a generic service-principal credential.

The Sponsorship Model:

Every Agent Has a Human Behind It

Agent 365 introduces a two-role accountability structure that MSPs should adopt as standard practice for any agent they build:

- The Sponsor is the business owner. They answer for what the agent does. They cannot be bypassed.
- The Owner is the technical lead. They build, configure, maintain and troubleshoot it.

Both are humans with Entra identities. When the Sponsor leaves the organization, agent access is automatically revoked. That single mechanism solves a category of risk that’s otherwise very hard to manage manually — orphaned agents continuing to act in the world long after the person who authorized them has moved on.

The Five Governance Pillars Agent 365 Enables

For MSPs building governance offerings around this, the framework breaks down into five areas.

Pillar	What It Covers
Registry	A central inventory of every agent in the tenant — who created it, what data it touches and when it was last used.
Access Control	What data each agent is allowed to reach, with sensitivity labels and DLP enforced at the agent level.
Visualization	Activity and data-flow visibility — which agent did what, when and on whose behalf.
Interop	Cross-service and third-party agent governance, not just first-party Microsoft agents.
Security	Conditional Access policies applied to agent identities, the same way they’re applied to users.

The SMB Stack:

Business Premium + Purview + Agent 365

For most SMB customers, the foundation for governed AI agents is closer than it looks. Business Premium provides Conditional Access, which now extends to agent identities. The Purview Suite provides DLP and sensitivity labels that govern what data agents can touch. Agent 365 ties it all together with the registry, sponsorship and visualization layer.

The customer conversation is straightforward: “You already have Business Premium, so you’re halfway there. Add Purview and Agent 365, and you have governed AI agents.” That reframes the discussion from a tech upsell into a compliance and risk posture conversation, which lands very differently in a boardroom than another line item on a Microsoft bill.

What This Means for Your Multi-Tenant Agent Specifically

When you publish a multi-tenant agent, both sides of the relationship benefit from Agent 365.

- **In your host tenant,** your agent has a named identity, a Sponsor, an Owner, an audit trail and Conditional Access policies you control. You can manage it, decommission it and prove governance to any customer who asks.
- **In your customer’s tenant,** your agent appears as a third-party application in their inventory. If they have Agent 365, it shows up in their registry the moment a user installs it, with their access controls and DLP policies applied. They can govern your agent inside their environment without you having to do anything special.

That last point is the missing piece that makes the multi-tenant story commercially safer: governance becomes a shared responsibility model rather than a “trust me” situation. Customers with mature security postures will start asking about it. MSPs who can answer the question fluently will win those conversations.

Multi-Tenant Management Made Easier

Multi-tenant monitoring and management is a cornerstone of any scalable practice. Microsoft announced a whole set of new capabilities at Microsoft Build and Pax8 Beyond that make multi-tenant management easier and more robust:

- **Agent management across all tenants directly in Microsoft Admin Center (admin.cloud.microsoft).**
- **A tenant-switcher for fast, seamless navigation between managed tenants using GDAP.**
- **A unified Agent 365 registry view spanning all managed tenants.**
- **Deployment of agents across multiple tenants from a single interface.**
- **Consolidated reporting on agent activity, operational state and risk posture.**

These new capabilities are a major leap forward for multi-tenant management. Partners can now see unified insights on agent activity and risk posture across their entire customer base and move between tenants with the kind of speed and efficiency that previously wasn't possible.

A Decision Framework for MSPs

Multi-tenant agents are real, they work and the build path is now proven. But before any MSP gets to "should we, where and how?", there's a more fundamental question to settle.

The Threshold Question:

Are You Ready to Operate Like a Software Vendor?

The moment you publish a multi-tenant agent, you've crossed a line. You are no longer deploying technology into a customer's environment, you are running technology for customers, from your own environment, on your own dime and under your own brand. That's the operating posture of a SaaS company or ISV, and it's a meaningful shift away from how most MSPs work today. Five dimensions of the business must change with it.

1. Billing

Traditional MSP work charges per tenant, per seat, per ticket — costs and revenue track cleanly to specific customers. Multi-tenant agents don't work that way. Your costs are pooled in your environment, growing with total usage across all customers, with no native per-tenant attribution from Microsoft. That means your pricing model must be designed up front, not derived from a vendor invoice. Whether it's a flat monthly fee per customer, a usage-banded subscription or a per-seat add-on, the commercial structure must live in your contract paper because it doesn't live in the platform. Pricing this without thinking like a SaaS company — modeling usage curves, gross margin per customer, capacity break-even — is how MSPs end up subsidizing their largest customers without realizing it.

2. Data Isolation

A traditional MSP deployment isolates data by construction so that each customer's data lives in their own tenant. Multi-tenant agents flip that. Your hosted data sources are reached by every customer who uses the agent, which means data isolation is now an architecture decision you make, not a property of the platform. If customer-specific data ever flows through your agent, you have to design and document exactly what's stored, what's logged, what's shared and what crosses tenant boundaries. Enterprise customers will ask. Mature security teams will ask. Your own legal counsel will ask. Have an architecture diagram and a written answer ready before they do.

3. Capacity and Reliability

A SaaS provider monitors capacity, sets service-level commitments, alerts on degradation and absorbs the operational burden when something breaks. An MSP with a multi-tenant agent now has to do the same, except for an undefined population of users, with limited telemetry during the preview and a single failure mode (capacity exhaustion) that takes every customer down at once. Plan to staff for it. Capacity exposure isn't a quarterly review item for a SaaS company, it's a daily one.

4. Release Management

With single-tenant deployments, you can roll changes out customer by customer, on each customer's pace and testing window. With a multi-tenant agent, every change ships to every customer simultaneously. There's no staggered rollout, no per-customer change control and no "let's wait until next quarter for that one." Mature SaaS companies handle this with feature flags, environment promotion and a published release cadence everyone subscribes to. MSPs adopting this model need at least a lightweight version of that same discipline, or quality issues will surface as cross-customer incidents that damage multiple relationships at once.

5. Contracts and Liability

Once you're providing software-as-a-service, you're providing it under your own terms, your own SLA, your own privacy commitments and your own data processing language. The shape of that paper looks more like a SaaS subscription agreement than a managed services agreement. Build it before you sign your first customer, not after. Include usage caps, force majeure language for capacity issues during preview, explicit data handling clauses and a clear delineation between what your agent does and what Microsoft is responsible for underneath it.

The MSPs who succeed with this will be the ones who recognize they're crossing into ISV territory and act accordingly — investing in the pricing models, isolation architecture, capacity practice, release discipline and contract paper of a software vendor. The ones who try to retrofit traditional MSP operations onto a fundamentally different delivery model will find out the hard way that the economics don't work and customers won't accept it.

With this framework established, here are examples of what you can start right now and what might be better to take your time on:

Good Fit Today

- A repeatable knowledge or workflow problem that several customers share, using data you control.
- A small number of known customer tenants you can onboard one at a time.
- A pricing conversation you can have outside the Microsoft billing model.
- An internal team that can absorb a more involved deployment process and a steeper learning curve than you're used to.
- Customers (or your own organization) on Business Premium plus Purview, where Agent 365 governance can be turned on without a foundational license uplift.

Wait for General Availability

- Use cases that depend on personalizing answers to the individual user from their own tenant data.
- Customers with strict data residency requirements that don't match where your agent is hosted.
- Distribution strategies that need self-serve sign-up and per-customer billing.

Always

- Plan capacity and set alerts before the second customer goes live.
- Put usage limits in your contracts. "Fair use" is not a financial control.
- Treat the second test tenant as a permanent part of your environment, not an optional extra.
- Write the privacy policy, terms and data processing addendum before you publish, not when a customer asks.
- Stand up Agent 365 in your host tenant before the first agent ships. Define the Sponsor and Owner for every agent. Decide what your registry, access and Conditional Access policies look like for agents before you have a fleet of them to retrofit.

What We're Watching on the Roadmap

Two of the biggest gaps in the multi-tenant preview have clear roadmap items: user-level authentication and detailed analytics. Once those are generally available, the offering shifts from “narrow private distribution under contract” into something an MSP can confidently put on a price sheet. Multi-geo support and broader distribution channels are also flagged.

In the meantime, the right posture is the one that matches our build: a tightly scoped agent, a known data source, a small set of explicitly enabled customer tenants, a contractual structure that protects you from the bill and Agent 365 as the governance layer underneath it all. The technology to ship multi-tenant agents arrived in preview. The technology to govern them properly arrived on May 1. Together, that's a complete enough picture for MSPs to start building real offerings carefully and with the operating mindset of a software vendor, not a managed service shop.

About Pax8 Professional Services

Pax8 Professional Services helps MSPs design, build and operate Microsoft 365 and Azure solutions at scale. From AI strategy and Copilot Studio agent development to governance frameworks and Agent 365 enablement, our team works alongside partners to turn emerging capabilities into repeatable offerings.