Article

Oct 2026

Managing your M365 automation estate before Copilot finds your gaps

Here's a question worth asking your team this week: how many Power Automate flows are running in our tenant right now?



Billy Pickin
Digital Productivity Consultant

In our experience, quite often, nobody knows. Other questions that tend to follow include:

- Who owns them?
- Which ones matter?
- What stops working if we disable a leaver's account on Friday?

People built, and continue to build, things to solve real problems. Nobody was asked to keep a register. Low-code allows people to do this, all by design.

Copilot and agent-building tools now drop the barrier again. Building something useful has gone from having to learn the platforms, to simply describing what you want. This often means the volume of "things" being built is going up, the builders are less technical, and the things being built now make their own decisions. This should all sound great, because it genuinely can be. But there's another side to consider: governance.

Perhaps rather unintuitively, the answer is not to lock it down. You'd lose the value of the platforms available, and people will always try to route around things anyway. What's needed is visibility, clear ownership, and guardrails wide enough that nobody notices them, until they should.

Start with one automation

As we covered in our last article, imagine you've built an agent that handles supplier invoice exceptions. It reads invoices that fail the automatic match, works out what the discrepancy is, checks the goods receipt, and either drafts a query to the supplier or proposes a resolution. When it has decided, it calls an existing Power Automate flow to post the transaction. Useful. Well built. Finance like it.

When we break it down, it connects to a lot of different things:

- It reads the finance mailbox.
- It reads the supplier data.
- It has access to the SharePoint site where purchasing keep their documents.
- It calls a flow that writes to the finance system.
- And the connection that flow uses belongs to the finance manager who built the first version of it three years ago.

We've *already* got a few questions on this one. But the reality is there's often even more to consider:

- Procurement built their own version because they didn't know finance had one.
- Someone's created a declarative agent on the finance SharePoint site that answers supplier queries.
- There's a desktop flow on a machine under someone's desk that does a nightly export, and it's been running since before the last two people left.

Deeper dive for more technical folk

When you go looking, you're inventorying more than flows:

- Cloud flows and desktop flows, including ones running on attended machines
- Power Apps, canvas and model-driven
- Copilot Studio agents
- Declarative and SharePoint agents, which multiply fastest because they are easiest to make
- Connections and the accounts behind them
- Custom connectors
- Service accounts, and whether anyone knows their credentials

The Power Platform admin centre gives you most of this. The CoE Starter Kit gives you a fuller picture, though be realistic about it: it's a proper piece of software that needs owning and maintaining, not a report you install once.

The one that bites

Connection ownership is often the one thing that causes the most trouble and gets the least attention

That invoice flow runs on the finance manager's personal connection, not a service account, because that's what she was signed in as when she built it. Why does this matter?

- She leaves in March.
- IT disables the account on her last day.
- The following Monday, invoices stop posting.
- Nobody links the two events for two days, because nobody knew the flow existed. Let alone whose credentials it was using.

We've helped clients resolve these very issues over the years. The fix afterwards can be straightforward. The two days of finance wondering why nothing is working tend to be less straightforward.

There's a usual pattern: a personal connection on something that's quietly become business-critical. If you take away one thing from this article, make it this learning.

Deeper dive for more technical folk: where things should live

Environments. The default environment is where everything lands if you do not intervene, which is how it becomes a bit of a mess. Everyone in your tenant has access to it by default. Anything that matters should be in a named environment with its own security, its own DLP policy and a clear owner. Something touching supplier payment data should not be sat in the default environment next to someone's holiday tracker.

DLP policies. Connectors get classified into business, non-business and blocked, and a flow cannot mix groups. The useful question is not "which connectors do we allow" but "which combinations would we be uncomfortable with". Finance data plus an outbound connector in the same automation is the obvious one.

Sharing limits. You can cap how widely a maker can share an app or flow. Worth setting before someone shares something with an "everyone" group.

Solution checker and managed environments. If you're at the point of promoting things between dev and production, these are how you stop hand-built changes going straight into live.

The one that worries security teams

Agents inherit the permissions of whoever is using them. That sounds fine, until you think about what it means in a real SharePoint estate.

If your finance site has broken permission inheritance somewhere, and a folder that was shared with more people than anyone remembers, a search does not usually surface it. Nobody browses to it. It sits there being quietly over-shared and causing no visible harm.

An agent will find it. Then it will summarise the contents in a friendly sentence and present them to someone who was never meant to see them. It hasn't broken any rules. It's used exactly the access it was given.

So agent readiness is largely data permission readiness. That's less exciting than building agents, but honestly, it's the work that actually needs doing first. Microsoft's oversharing and access reports can help you find it.

Where an agent reads and uses personal data to generate its answers, supplier contacts included, there are GDPR questions worth taking seriously rather than assuming. Is a DPIA needed? Where does processing happen? How long is anything retained?

Deeper dive for more technical folk: lifecycle

The estate needs the same joiners, movers and leavers process as everything else.

- **Joiners**. Makers get an environment and know where to build. Not the default one.
- **Movers**. Ownership transfers when someone changes role. Co-owners on anything that matters, so there is never a single point of failure.
- **Leavers**. Offboarding includes a check for owned automations and connections. This is the one that prevents the March problem above.
- **Retirement**. Flows that haven't run in 90 days get a question asked about them. Some are seasonal and fine. Most are probably no longer needed and can go.
- **Review**. A quarterly look at what's new, what has grown, and what has quietly become important. Half an hour, not a project.

Where to start

If you do nothing else, do these three, roughly in this order.

Find out what you've got. You can't make decisions about an estate you can't see. Expect the number to be higher than you think.

Fix personal connections on anything business-critical. Move them to service accounts with proper ownership.

Sort out the default environment. Decide what's allowed to stay there, and give anything that matters somewhere better to live.

After that you're into environment strategy, DLP and permission clean-up, which are more work but far less urgent than the three above.

The point of all this

None of this is about slowing people down; it's quite the opposite. The organisations getting real value out of Power Platform and Copilot agents are the ones where the guardrails went in early and quietly, so makers never really noticed them.

The alternative is finding out what's in your estate at the worst possible moment, usually two days after someone's last working day.

If you want the background on why agents need different controls to flows, the previous article covers what actually changes when automation stops following a fixed path and starts deciding for itself.

