

Grounding trust is the decisive commitment in enterprise Copilot for financial services.



- Many financial-services firms have deployed Copilot onto a decade-old enterprise estate, and adoption is stalling.
- The instinct is to blame the model, the training, or often, the rollout, but the real constraint sits one layer below, in the estate Copilot reasons over.
- On an ungoverned estate, unpredictable failures like oversharing exposure, confidently wrong answers, provenance blindness, and governance drift each carry a supervisory edge in a regulated environment.

- A five-layer architecture—a grounding trust flywheel—establishes trust, and content earns it only when it is correctly permissioned, sensitivity-labeled, provenance-tracked, and governed, all at once.
- **ADAM**, our AI Accelerator Platform, operationalizes this architecture, turning grounding trust into governed production.

The limits of licensed but ungoverned Copilot deployments

Over the last two years, financial-services firms have moved quickly to put Microsoft Copilot in front of their people. Licenses have been assigned, apps rolled out, pilots run. Access to AI, at the point of work, has genuinely improved. And yet, beneath that progress, a structural problem persists. The Copilot programs sitting on top of all that connectivity keep disappointing. Answers underwhelm or arrive with a confidence the underlying content does not justify. Adoption climbs during the pilot and then quietly plateaus. And in regulated environments, a more serious pattern emerges ever so often. Copilot surfaces something it should never have surfaced. An unlabeled deal file, a client record from a team the user was never meant to see, or a document whose permissions no one has audited in years.

The diagnosis most executive teams reach is that the model is not good enough, or the training was thin, or adoption needs another push. Each of these is treated as a fixable program issue. But they are symptoms, not causes. The actual constraint sits one layer below, in the estate Copilot reasons over. This exposes a structural limitation. The Microsoft 365 estate was architected to manage content, collaboration and access, not to govern machine reasoning over that content at scale. It optimizes how people store and share information. It does not, by default, govern what an AI is allowed to infer from it, or whether the answer it produces can be trusted, scoped and evidenced. In financial services, outcomes are not determined by whether Copilot is deployed. They are determined by whether the estate beneath it is grounded.

‘Deployed’ does not mean ‘trusted.’ Rolling Copilot onto an ungoverned estate is not a productivity move. In a regulated firm, it is a huge strategic risk.

Shifting from deployed to grounded Copilot estates

Deployment answers the question of access: can our people reach Copilot? Grounding answers a different and more consequential question: can we trust what Copilot reaches, and what it does with it? These are not the same program and treating them as one is the root cause of most stalled Copilot initiatives.

What is grounding?

Grounding is the mechanism by which Copilot connects a user’s request to the organization’s own content, retrieving from SharePoint, OneDrive, Teams and connected systems to compose an answer. It is the feature that makes Copilot enterprise-grade rather than a generic chatbot. But grounding is only as good as the

estate it draws on. If that estate carries 15 years of accumulated permissions, unlabeled sensitive content and duplicate, conflicting records, Copilot faithfully reflects all of it back, at machine speed, to whoever asks.

Consider what ‘Copilot-ready’ means in most firms today.

Licenses assigned. Apps live. But the SharePoint and OneDrive estate underneath was built over a decade and a half, with access controls layered on by successive teams and rarely reviewed. An M&A workspace is inherited by 400 people through nested group membership no one has unpicked. A confidential client file carries no sensitivity label. A retention policy lapsed two reorganizations ago. Copilot does exactly what it is designed to do: it grounds its answer in all of it.

Now consider a single client, as they actually exist across a bank’s systems.

- In the CRM, they are flagged ‘high net worth.’
- In the compliance system, they are a politically exposed person, Tier 1.
- In onboarding, they exist only as an untagged PDF.
- In the trading system, they carry a risk code with no human-readable description.

These are four representations of one person’s reality. Absent a governing layer, Copilot has no way to know they are the same client, which record is authoritative, or which of them a given user is entitled to see. It learns the encoding quirks of each source system, not the truth beneath them.

That is not an edge case. It is the standard condition of enterprise content at rest. The connectivity problem, getting content into one platform, has largely been solved. The consistency and governance problem, making that content safe and coherent for an AI to reason over, has barely been touched. The move that matters now is from deployed to grounded.

Failure patterns: Four ways an ungoverned estate breaks Copilot

When Copilot is switched on over an ungoverned estate, the failures are not random. They follow a predictable pattern, and in a regulated firm, each one carries a supervisory edge that turns an inconvenience into an exposure.

1. Oversharing exposure

Copilot surfaces content a user can technically reach but should never see. Because grounding respects existing permissions, a single over-permissioned site, one bad access control set years ago, becomes an enterprise-wide leak the moment Copilot makes that content easy to find and summarize. Deal terms, client PII, or material non-public information that was effectively hidden by obscurity is now one natural-language question away. The permission was always wrong; Copilot simply makes the consequence visible and instant.

2. Ungrounded, confidently wrong answers

When Copilot grounds on stale, duplicate or conflicting documents, it returns an answer that is fluent, plausible and wrong. In everyday productivity this is an annoyance. In a suitability assessment, a disclosure, or a client communication, a confidently wrong answer is materially worse than no answer at all, because it carries the authority of the system and invites action. Without a single authoritative version of the truth to ground on, Copilot cannot be relied upon where reliability matters most.

3. Provenance blindness

Ask most Copilot deployments to show their working, to trace an answer back to the specific sources it drew on, and they cannot. There is no durable audit trail from output to origin. In a regulated firm, this is disqualifying. An answer you cannot evidence is a control gap waiting for an examiner's question, and 'the AI said so' is not a defensible position in front of a supervisor. Explainability is not a nice-to-have here; it is the difference between a usable system and an unusable one.

4. Governance drift

Sensitivity labels, DLP rules and retention policies are not set-and-forget. They decay as the organization changes, as teams reorganize, as content migrates, as new sources connect. An estate that was compliant last quarter silently is not this quarter, and Copilot keeps reasoning over it as though nothing has changed. Without continuous governance, grounding trust erodes invisibly, and the first sign of the erosion is often the incident itself.

None of these are model failures. A better model does not fix an over-permissioned site, reconcile a duplicate client, generate a missing audit trail, or refresh a lapsed policy. They are data-governance failures, and they require architectural investment, not another round of model evaluation or user training.

The precise idea: What grounding trust actually is

Grounding trust is not a general aspiration about 'good data hygiene,' and it is not a product you can buy. It is a precise, testable property of an individual piece of content. A document, record or data element has grounding trust when it satisfies four conditions at once:

The four conditions of grounding trust

- **Correctly permissioned:** Only the people who should reach it can, with no inherited or accidental oversharing.
- **Sensitivity-labeled:** Classified and protected against a live policy, not left unmarked and undefended.
- **Provenance-tracked:** Carries auditable lineage: source, owner, freshness and version, so any answer built on it can cite it.
- **Governed and current:** Its labels, DLP and retention are kept live as the estate and the rules around it change.

When all four conditions are met, the content is fit for Copilot to ground on. When even one fails, that content is a liability, and Copilot, doing exactly what it is designed to do, will find it. Grounding trust, then, stops being a slogan. It is a standard you can hold every element of the estate against, and the architecture that follows exists to enforce that standard at scale.

A five-layer architecture: The Grounding Trust Flywheel

The response to a grounding-trust problem is not a product purchase. It is an architectural posture: a deliberate sequence of layers that ensures every signal entering the estate is grounding-trusted before Copilot, or any downstream agent, is permitted to reason over it. We call this the Grounding Trust Flywheel. It has five layers, and the sequence is not optional, each layer is a prerequisite for the one above it. Deploy out of order, and you spend on capability while the foundation beneath it silently degrades.

Layer 1: Capture (discover the estate)

Everything begins with knowing what you have. The first layer inventories every relevant source, SharePoint, OneDrive, Teams, mailboxes and the connected line-of-business systems Copilot can be extended to reach and captures the permissions and labels each currently carries. The grounding-trust problem begins the moment content exists, so nothing above this layer can be trusted until the estate has been mapped. You cannot remediate, and you certainly cannot govern, what you have not seen.

Layer 2: Gateway (where trust is established)

This is the heart of the architecture, the layer that sits between raw content and every downstream consumer. Before Copilot is switched on, four things happen here, in sequence and at scale: oversharing and over-permissioned sites are remediated; sensitivity labels are applied across the estate; DLP and retention are stood up and enforced; and provenance is attached to every element, its source, owner, freshness and the policy version under which it was classified. Critically, this runs against live policy, not a one-off snapshot. When the organization's rules change, the gateway changes with them. The gateway is where content earns its grounding trust or is held back until it can.

Layer 3: The governed grounding layer

Downstream of the gateway sits the curated, continuously validated view of the estate that Copilot is permitted to ground on, and nothing else. No content reaches this layer without passing the four conditions. It is the single authoritative representation of what the enterprise knows, normalized, reconciled and provenance-tagged, and it exists to make one guarantee: that whatever Copilot reasons over is worth reasoning over. This layer is the difference between an AI that reflects the mess of the estate and one that reflects its governed truth.

Layer 4: Assistive Copilot on trusted ground

With a governed layer beneath it, Copilot finally delivers what it has been promising. Answers are grounded in authoritative content, scoped to the permissions the user actually holds, and, because provenance travels with every element, each response can cite where it came from. The compliance team can evidence why Copilot said what it said. Productivity gains arrive without the accompanying anxiety, because the same architecture that makes Copilot useful also makes it defensible. This is the layer most firms are trying to reach by buying licenses; it is only reachable by building the three layers below it first.

Layer 5: Agentic workflows, governed by design

The apex of the architecture is agentic execution: autonomous agents that act on behalf of teams, KYC refresh, suitability checks, first-line servicing, assembled and governed in Copilot Studio. Each agent action generates new signals: an approval, a completed check, a servicing record. Those signals flow back through the gateway, are normalized and provenance-tagged, and re-enter the grounding layer enriched. This is what makes it a flywheel rather than a pipeline. The system does not merely run; every cycle improves the data foundation for the next, and the organization's grounding trust compounds over time instead of decaying.

Capture → govern → ground → assist → act → refine.

The sequence is the contribution. Skip a layer and Copilot spends the organization's money while the estate beneath it quietly erodes.

What the architecture unlocks: one question, answered with evidence

The value of grounding trust is easiest to see in a question a relationship manager, or a first-line servicing agent, might reasonably ask:

“Which of my clients hold the fund we are about to reclassify, have a suitability profile that no longer fits it, and have not been contacted in the last 90 days?”

On an ungoverned estate, Copilot cannot answer this safely. The holdings, the suitability records and the contact history live in different systems, with inconsistent client identifiers and no permissions model Copilot can trust. Any answer it produces is a guess wearing the costume of authority. On a grounding-trusted layer, the same question becomes a scoped, explainable query, resolved against reconciled data, filtered to the clients this user is entitled to see, and an agent can go one step further and draft the outreach, with every step evidenced back to source. The answer is available today, and, just as importantly, it is one the firm can defend.

Stage 1	Connected	The Microsoft 365 estate and Copilot licences are in place.
Stage 2	Deployed	Copilot is switched on. Governance is ad hoc, applied where someone happened to think of it. Most firms are here.
Stage 3	Grounding gateway	Oversharing remediated, content labelled, provenance tracked. The prerequisite for everything above. This is the unlock.
Stage 4	Trusted assist	Copilot grounded on authoritative content, explainable, and genuinely adopted because it can be trusted.
Stage 5	Agentic	Governed agents that act, and a flywheel that keeps improving the foundation with every cycle.

As regulatory frameworks around AI use crystallize, traceability stops being good practice and becomes a requirement, at which point the organizations that built provenance in from Stage 3 will already hold the asset their competitors are scrambling to retrofit.

From architecture to production with ADAM: The execution engine for grounding trust

Many financial-services organizations discover that their Copilot ambitions stall not for want of a diagram, but for want of the operational scaffolding to turn the diagram into production. An architecture on a slide is not the same as a remediated estate, a live governance framework, and a working set of agents. Bridging that gap, repeatably, in a regulated, high-complexity environment, is where execution frameworks matter.

We built ADAM (Agentic Data and Application Management), precisely to operationalize AI in environments like this. Rather than treating grounding trust as a one-off remediation project, ADAM makes it executable and continuous, embedding the flywheel's layers into the places where decisions actually happen. It brings together a set of accelerators that reduce delivery risk and shorten time to value:

- A readiness and remediation layer that scans permissions, tenant settings, and content-exposure risk across the estate before Copilot is enabled, so activation is safe by design rather than by hope.
- A governance framework that applies sensitivity labels, DLP and responsible-AI guardrails at scale, and keeps them live against changing policy, so security signs off instead of stepping in.
- A telemetry and ROI layer that measures adoption depth by persona, hours saved and productivity lift, so ROI is evidenced rather than asserted, and grounded in data the CFO will accept.
- An agent factory that provides reusable patterns, blueprints and orchestration components to build, govern and deploy custom Copilot agents in Copilot Studio, so Stage 5 is an engineered step, not a leap of faith.

The emphasis throughout is integration, not replacement. ADAM works with the Microsoft estate the firm already owns; it does not force a rip-and-replace. Intelligence is modular, explainable and auditable, ownership remains with the client, and as outcomes improve, investment scales with value rather than with lock-in. In the context of grounding trust, ADAM is the engine that turns the flywheel: it is what moves a Copilot program beyond pilots, proofs of concept and isolated use cases, and into governed production at enterprise scale.

Four actions to operationalize grounding trust for the board

Grounding trust is ultimately a board-level architectural commitment, not a task to be delegated wholesale to the Copilot pilot team. Four actions turn the commitment into motion:

- **Assess the estate, not the tool.** Establish whether your Copilot rollout sits on a governed grounding layer or merely on assigned licenses, and measure oversharing exposure before it surfaces, not after. The estate is the risk; the tool is not.
- **Invest in the gateway first.** Prioritize remediation, labeling and provenance ahead of scaling seats. The gateway is the prerequisite for every capability above it, not a parallel workstream to be funded later.
- **Scope Copilot to trusted domains.** Deploy where grounding trust is established first, capturing defensible early wins while the rest of the estate is remediated. Sequenced scope beats big-bang exposure.
- **Build the regulatory asset now.** Implement provenance and audit trails from day one, so every Copilot and agent output is explainable and evidenced against future regulatory scrutiny. Built now, it is an asset; retrofitted later, it is a cost.

Three crucial considerations about Copilot and governance

Adoption and governance are one program, not two

The instinct to run a ‘Copilot adoption’ track and a ‘data governance’ track separately costs the organization on both fronts. Investment in the grounding gateway directly improves Copilot’s output; Copilot’s own usage patterns, in turn, reveal where governance effort should focus next. Split them, and you create two teams working against each other’s assumptions. Unify them, and each compounds the other’s value.

Agents raise the stakes; they do not lower them

A common argument holds that smarter, more capable models make governance less necessary, that the AI will simply handle the mess. The opposite is true. An assistant that surfaces the wrong document is a problem; an autonomous agent that acts on that wrong document, refreshing a record, submitting a request, closing a case, is a far larger one. Grounding trust is not an alternative to governance in an agentic world. It is its most demanding application, and its precondition.

This is about sequencing, not new technology

The components of grounding trust, sensitivity labels, DLP, permissions remediation, provenance, the semantic index, are established Microsoft capabilities. The contribution here is not a new tool but the correct sequence for assembling and governing them, and the insistence that the sequence is non-negotiable. Deploy Copilot before the grounding layer exists, and the firm spends heavily on capability while the foundation quietly erodes beneath it. The technology is available today; what is scarce is the discipline to build it in order.

A new foundation for enterprise AI in financial services

Deployment helped financial services firms open the door to Copilot. Grounding trust is what will let them walk through it safely. By treating the estate as the thing that must earn trust, before the model is ever asked to reason over it, firms can finally align productivity, control and defensibility in a single architecture, rather than trading one against the others.

The flywheel starts with trust, and trust needs a governed estate. Governing the estate is the most consequential architectural commitment in enterprise Copilot today. Made deliberately, at board level, it is what separates the firms that get productivity and a defensible audit trail from the firms that get an incident. In an environment where regulators, clients, and boards are all sharpening their questions about AI, that's the only competitive (and not technical) distinction that matters.

Four moves that decide your Copilot outcome

- 'Deployed' does not mean 'trusted'. Access to Copilot is not the same as trust in what it reaches. In a regulated firm, rolling it onto an ungoverned estate is a strategic risk, not a productivity move.
- Fund the gateway before the seats. Remediation, labeling, and provenance are the prerequisites for every capability above them. Sequenced scope beats big-bang exposure.
- Agents raise the stakes, not lower them. An assistant that surfaces the wrong document is a problem; an autonomous agent that acts on it is a far larger one. Grounding trust is governance's most demanding application.
- Build the regulatory asset now. As AI traceability requirements crystallize, the firms that built provenance in from Stage 3 will already hold the asset their competitors are scrambling to retrofit.

Start with a free two-hour Copilot check-in

A live introduction to Copilot and agents, an ADAM-based grounding-trust and readiness snapshot of your estate, and quick-win scoping. It is stage one of our 'Ready for Take-Off' journey, from pilot to governed enterprise scale. [Link](#)

About Brillio

Brillio is The Enterprise AI Accelerator helping Fortune 1000 companies move from AI ambition to scaled impact, faster. Powered by our AI accelerator platform – Agentic Data and Application Management (ADAM), Brillio is one of the fastest-growing digital technology service providers, delivering transformation across five core workstreams: business-led transformation, customer experience transformation, AI and data engineering, digital engineering, and infrastructure engineering.

With 14 delivery locations across North America, Europe, and Asia and a team of over 6,000 customer-obsessed professionals, Brillio combines deep industry expertise, modern engineering, and accelerators to deliver measurable outcomes. Headquartered in Dallas, Texas, Brillio serves clients globally with a commitment to speed, scale, and measurable impact.



<https://www.brillio.com/>

Contact Us: info@brillio.com

