

Whitepaper

Why Copilot requires a cleanup, and how we made cleaning obsolete

A field guide to improve adoption of Copilot — the retrieval mechanics behind disappointing answers, and the architecture that fixes them without touching SharePoint.



**Keep Microsoft. Keep the mess.
Fix the answer.**



THE STARTING POINT

The 36% problem

Microsoft Copilot has one of the fastest enterprise rollouts in software history. But sustained adoption tells a different story: across many deployments, only around a third of licensed users keep using it after the novelty fades.

The reason is rarely dramatic. People don't abandon Copilot because it's frightening or hard. They abandon it because **the answers disappoint**. A user asks for the latest pricing model, gets last year's draft. Asks for the signed contract, gets three versions and no clarity on which is final. After a few of these, they quietly go back to searching SharePoint by hand.

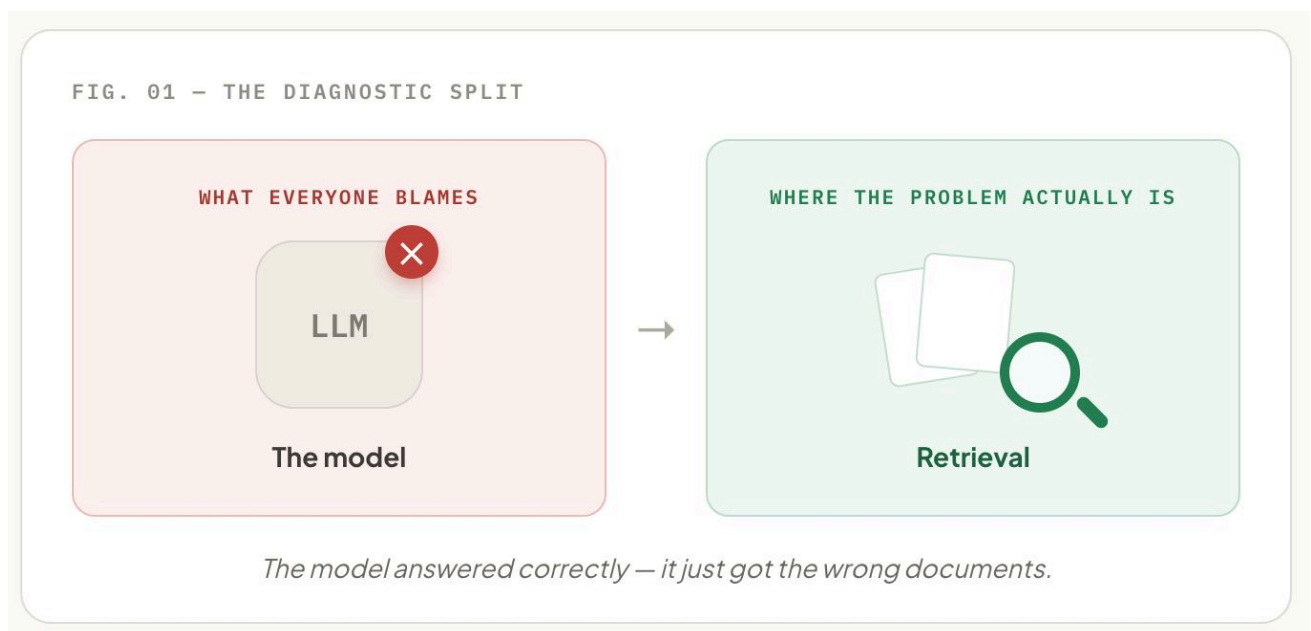
So the first question every manager should ask is the diagnostic one:

The diagnostic question

When Copilot gives a bad answer — is the model wrong, or did it never see the right document?

Almost always, it's the second. The model is fine. It answered faithfully based on what it was handed. The problem is **what it was handed** and that is a retrieval problem, not an intelligence problem.

This paper does two things. First, it explains why Copilot pushes organizations toward an expensive SharePoint cleanup. Then it explains why Aspected makes that cleanup obsolete.



Keep Microsoft. Keep the mess. Fix the answer.

PART 01

Why Copilot requires a cleanup

How Copilot actually finds an answer

Copilot doesn't read your whole tenant when you ask a question. It works on a **semantic index**, and at the core of that index is a **vector database**. Understanding the steps of how this works is the entire game, once you see the steps, you'll see exactly where things break.

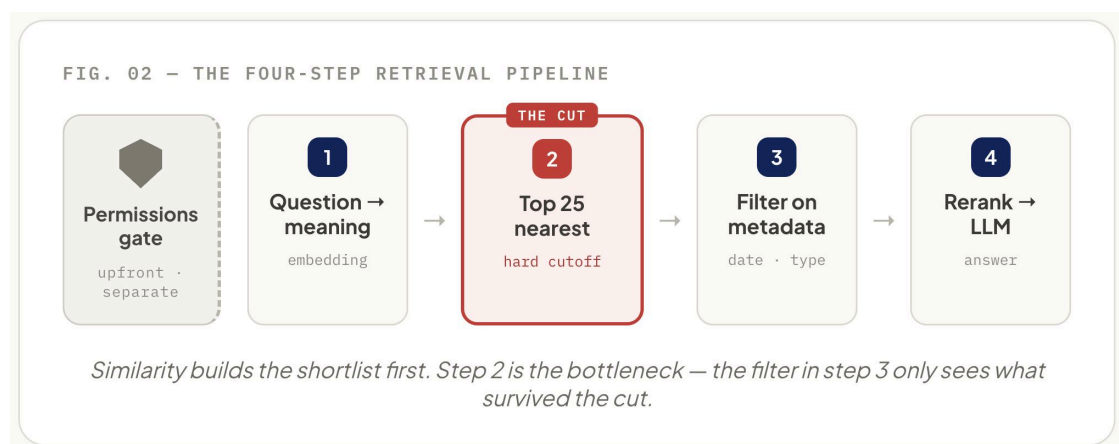
Here is what happens when a user asks a question:

- 1. Search by similarity.** The vector database converts the question into a mathematical representation of its meaning, then finds documents whose meaning is closest. It matches "wooden meeting table" to "MDF conference furniture" even though they share no words.
- 2. Take the top 25.** From everything similar, the system keeps roughly the **25 nearest neighbours**. This is the shortlist. Everything outside it is invisible for the rest of the process.
- 3. Filter on metadata.** Now the system applies filters: date, document type and author, to narrow that shortlist down to what's relevant.
- 4. Rerank and send to the LLM.** The survivors are reordered by relevance, and the best of them are handed to the language model, which writes the answer.

A note on permissions

Who is allowed to see what is enforced separately, up front. Copilot only ever surfaces content the user already has access to. That's a security boundary that applies before and around all of this, it is not one of the relevance filters in step 3, and it's not where retrieval quality goes wrong.

The order matters enormously. **Similarity comes first. Filtering comes second.** Hold onto that, it's the crack everything falls through.



Keep Microsoft. Keep the mess. Fix the answer.

Where it breaks: the mess pollutes the shortlist

A sense-check from outside the model

Independent LLM-spend reviews routinely find 22–48% of consumption recoverable through retrieval and prompt optimization (Enterprise LLM Cost Comparison, Feb 2026). The Base result below, 25% of total spend, lands inside that independently-cited band.

Now picture a real SharePoint tenant. Not a demo but a real one, with twenty years of history in it. A user asks: *"What's our latest company update on Project Falcon?"*

In a clean world, the latest update is one of the 25 nearest neighbours, the date filter picks it, done. But your tenant isn't clean. There are **thousands** of company updates in there — drafts, finals, Word versions, PowerPoint versions, last year's, the year before's, the reorganized copy someone saved to their personal folder. Every single one is semantically similar to "company update on Project Falcon." They all match on meaning.

So when the system takes the top 25 by similarity, **the shortlist fills up with old, near-identical clutter**. The genuinely latest update — the one the user actually wants — may sit at position 38. It never made the cut.

And here's the part that catches people out: the **date filter in step 3 can't save you**. The filter only operates on the 25 documents that survived step 2. If the right document didn't make the shortlist, no filter, no rerank, and no amount of model intelligence can pull it back. It's gone. The LLM writes a confident, fluent answer from polluted inputs — and the user gets last quarter's numbers presented as current.

This is the failure mode that erodes trust. Not a wrong model. A right model fed a contaminated shortlist.

“With Aspected Database, we have improved the quality of our RAG results by making metadata part of the search index. Traditional RAG systems search purely on text and only apply filters afterward. As a result, there is a chance that not all relevant data is included in the output. This makes AI unpredictable, slow, and expensive.

Aspected includes all context directly in the search query upfront. This enables immediate governance, cuts search times in half, and delivers maximum accuracy at lower cost.”

Wim Verheij, Managing Director at Novadoc Netherlands

Keep Microsoft. Keep the mess. Fix the answer.



"So just filter first" — and why that doesn't work either

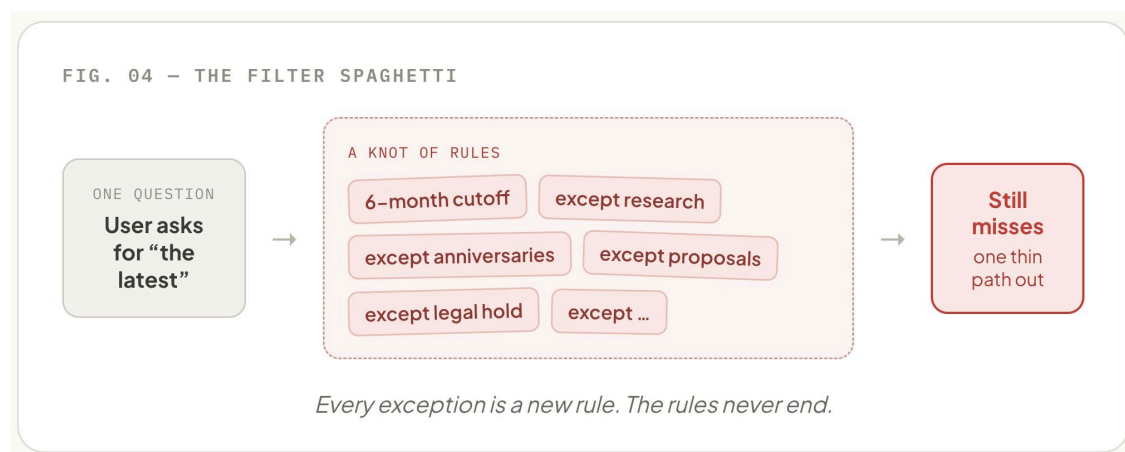
The obvious fix is to flip the order: **filter first, then search**. Apply the date and type constraints up front, so only fresh, relevant documents ever reach the similarity search. This sounds right. In practice it creates a worse problem, and it's worth understanding why, because sophisticated buyers will propose exactly this.

Filtering first requires you to decide the cutoffs in advance, for every kind of question. And there is no cutoff that's correct for everyone. Imagine your IT admin sets a sensible default: for general questions, only look at documents from the last **6 months**. Reasonable. But:

- What about something **7 months** old? Just missed the window, invisible.
- What about a **research project from three years ago** that you deliberately want to reuse?
- What about that **10-year anniversary speech**, the most relevant document is, by design, a decade old?

There is no single cutoff. So you start writing cutoffs per use case. Within months you have a **spaghetti of brittle, overlapping filter rules** that a team has to maintain, that breaks the moment someone asks an unanticipated question and that still misses the mark. Filtering first doesn't beat the structural problem. It just moves the pain from "polluted results" to "endless configuration."

Keep Microsoft. Keep the mess. Fix the answer.



So clean it up, right?

At this point the advice writes itself: **clean SharePoint**. Remove the duplicates. Archive the old versions. Fix the metadata. Then the shortlist won't be polluted and the whole problem evaporates. And it's true, cleaning would help enormously. If the thousands of near-identical company updates simply weren't there, the latest one would make the top 25 easily. So why doesn't everyone just do it?

PART 02

Why nobody actually cleans, the Xillio evidence

This is where we stop theorizing, because we have twenty-two years of data on exactly this question. Aspected is the brainchild of Xillio. For 22 years, **Xillio** has migrated companies onto SharePoint at roughly 50 migrations a year, that's around **1,000 organizations** moved into clean, new environments. A migration is the single best opportunity an organization will ever get to clean house: everything is being touched, moved, and inspected anyway. Across all 1,000, we saw the same three things every time.

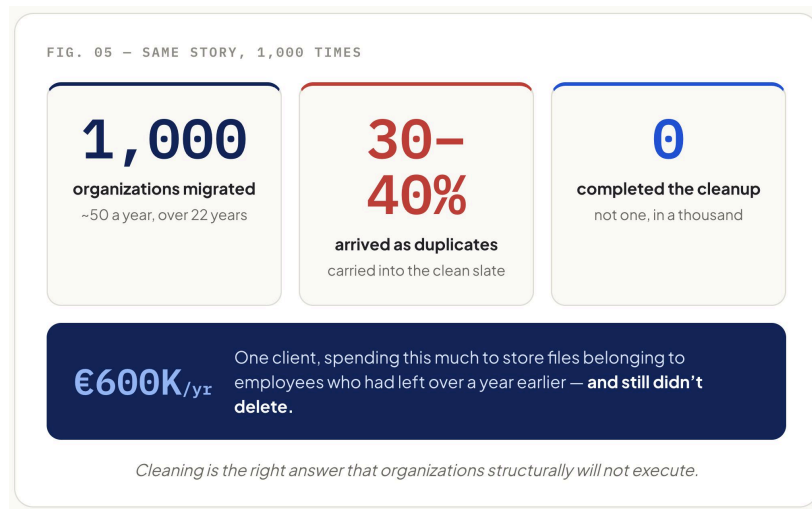
One: they all brought the mess with them. On average, **30–40% of what each client migrated into their pristine new SharePoint was duplicate file structure** carried over from the old system. The clean slate got dirty on day one.

Two: they all asked the right question. Every single client looked at that duplication and said some version of “*shouldn't we just delete this?*” They knew. They saw it. The instinct was correct.

Three: not one of them pulled the trigger. Every client hesitated, then stopped — and the reasons are rational: “How do we know which one is the original?” “I don't actually know what this is or who needs it.” “What if something breaks, or someone needs it for compliance, and we deleted it?”

The fear of deleting the wrong thing always outweighs the cost of keeping everything. The lesson from a thousand migrations is blunt: **cleaning is the right answer that organizations structurally will not execute**. It's not a tooling gap. It's a risk-and-ownership gap that no amount of good intentions closes. So we stopped trying to make cleaning happen. We made it unnecessary.

Keep Microsoft. Keep the mess. Fix the answer.



PART 03

How Aspected makes cleaning obsolete

Go back to the crack in the pipeline. The whole problem came from the order of operations: search for similarity first, take the top 25, then filter. The right document gets buried before the filter ever gets a chance to find it. Aspected fixes this at the root. **We include the filter inside the index itself.**

We don't first select the 25 nearest neighbours and then try to filter down to the latest. We make the metadata, recency, document type, filename, author, anything, **part of the very first query**. The vector and the metadata are fused into a single index, so retrieval happens in **one pass instead of two competing steps**. The system returns the 25 nearest neighbours that score high on **both** the text and the recency at the same time.

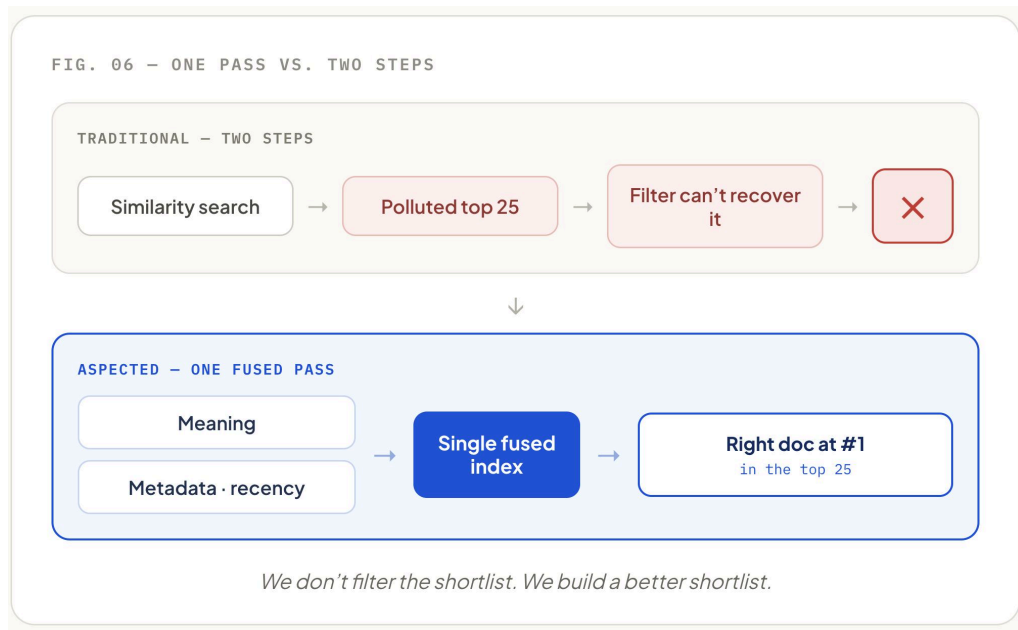
Now ask "what's our latest company update on Project Falcon?" again. The thousands of old versions are still sitting in SharePoint, **you didn't clean anything**, but they no longer crowd out the right answer, because the latest update scores high on both meaning and recency, and the old ones don't. That's what "cleaning is obsolete" means. Not that your tenant is clean, **that its messiness no longer degrades the answer**.



Aspected has added the context Microsoft Copilot needed. The answers are now perfectly aligned with how we organize and use our information."

Rob Kasslack, IT Business Partner, AI & Innovation at Agfa

Keep Microsoft. Keep the mess. Fix the answer.



Why this unlocks far more than "latest"

Fusing metadata into the index isn't just a recency trick. Because any metadata field can be part of the first query, you can ask multi-dimensional questions that traditional retrieval simply can't serve:

"Get me the latest proposal about Project X."

We return the 25 nearest neighbours that score high on **recency**, high on **document type = proposal**, and high on **the topic X**, all in one pass. Three constraints, no spaghetti, no missed cutoff.

"A couple of years ago I read a whitepaper — something like 50 Reasons to Clean SharePoint Before AI."

The kind of half-remembered, fuzzy query that breaks every conventional system. Aspected scores **document type = whitepaper**, **filename matching** the rough title, and the **semantic topic** even though the user got the title slightly wrong and doesn't remember the date. Vague human memory, precise retrieval.



Keep Microsoft. Keep the mess. Fix the answer.

Why this matters beyond Copilot answers — agents

There's a second, bigger consequence for anyone thinking past Q&A. Because Aspected can **isolate the filename — and any other metadata field — as its own queryable dimension**, separate from the document body, you can build **complex agents** on top of it. An agent that needs to reason over "the latest proposal, of type X, owned by team Y, excluding archived items" needs those dimensions to be independently addressable. In a traditional vector store where everything is blurred into one similarity score, that's fragile guesswork. With Aspected, each aspect is a clean, separate signal the agent can target precisely. That not only works for retrieval but also to steer consistency in answering. AGFA demanded accuracy AND consistency in answers as the answer was part of an agent chain. This consistency is possible as each item of the answer is an aspect.

Agfa Labs Search Result:

The document **IMPALA_VN_PA098631_XP_FA-ALK.DOCX** is an analysis report for a raw material called **MEG (Ethylene glycol)** provided by [REDACTED]

Summary

- **Client:** [REDACTED]
- **Contact:** [REDACTED]
- **Analysis conducted by:** AgfaLabs (Contact: Leo Timmermans)
- **Objective:** Analyze MEG for purity and content using:
 - NMR (for purity & DEG content)
 - ICP (for Fe content)
- **Sample ID:** PA098631-001-001 (Label: CP0200 MEG)
- **Method:** Potentiometric titration with TBAOH 0.1N on ~10 g of sample dissolved in a CH₂Cl₂/CH₃OH (13/2) mixture
- **Result:** Acid number ≤ **0.1 mg KOH/g**
- **Detection limit:** 0.1 mg KOH/g

The report confirms the MEG sample has a **very low acid number**, indicating **high purity regarding acidity**.

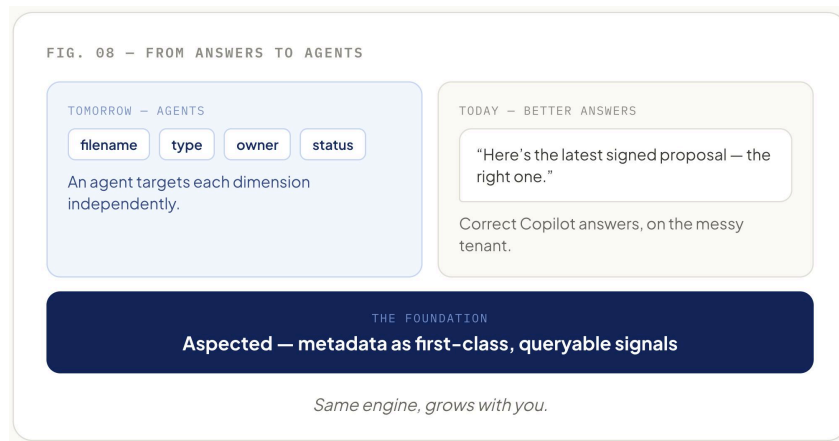
“ Aspected stood up in 30 minutes and gave us something we didn't have: a way to actually see retrieval quality side by side instead of guessing, which immediately exposed a gap in our own baseline. What stands out isn't a single 'it's better' number; it's that the aspect model gives you, and an agent, real levers to shape relevance per query that a plain vector store simply can't.”

Andy Copland, Director at CGA Management

So the same architecture that fixes today's Copilot answers is also the foundation for tomorrow's agentic workflows. That's the difference between a patch and a platform.

Aspected is based on a patented new database design. It is deployed within your own Azure tenant. We communicate through MCP and as such you can simply add other sources.

Keep Microsoft. Keep the mess. Fix the answer.



THE TAKEAWAY

How to relay the message

The whole argument in four sentences.

- 1. Copilot's disappointing answers are a retrieval problem, not a model problem** — the right document often never makes the shortlist.
- 2. Cleaning SharePoint would fix it, but a thousand migrations prove organizations won't finish the cleanup** — the risk of deleting the wrong thing always wins.
- 3. Aspected fuses metadata into the index so the right document makes the shortlist without any cleanup** — the mess stays, but stops mattering.
- 4. Every metadata field becomes an independent, queryable signal, so the same foundation lets you build real agents** — the fix for today and the platform for what's next.

About Aspected

Aspected is a new kind of database for enterprise AI developed by Xillio.

Its mission is simple:

Improve the reliability of enterprise AI by improving the knowledge layer underneath it.

<https://aspected.com>



Keep Microsoft. Keep the mess. Fix the answer.



**Learn more or schedule a session with our experts?
Contact us:**

 sales@aspected.com

 +31 (0)35 622 95 45

 www.aspected.com

 www.linkedin.com/company/aspected/