

PRODUCT CONTENT IN THE AGE OF AI

From description to intelligence.



Stephan Delbos

Sr. Director, Product Knowledge Activation ·

Mews

Grab the paintbrush.



Stephan Delbos

Sr. Director, Product Knowledge Activation · Mews

Twenty years in content and digital marketing, most of it spent making complicated products legible to the people using them.



Product Knowledge Activation

One team at Mews. We own product content across four surfaces, where every customer, every Mewser, and every AI assistant meets what the product can do.

01

Knowledge Base

The single source of truth. Every other surface cites this one.

02

Mews University

Structured learning. Onboarding, certifications, role-based paths.

03

In-product guidance

Contextual help where the work happens. Walkthroughs and tips.

04

Ada

AI resolution. Reads the KB, answers the customer.

One day, we got Ada.

An AI chatbot, fed on our knowledge base. I sat with the rollout brief on my desk and felt the floor move; the job of product content was about to change, and I had to decide whether to lead the change or be moved by it.

Before · 2023

Docs written for the customer reading.

Tone. Voice. Discoverability through search and navigation.
Humans interpreting structure with their eyes.

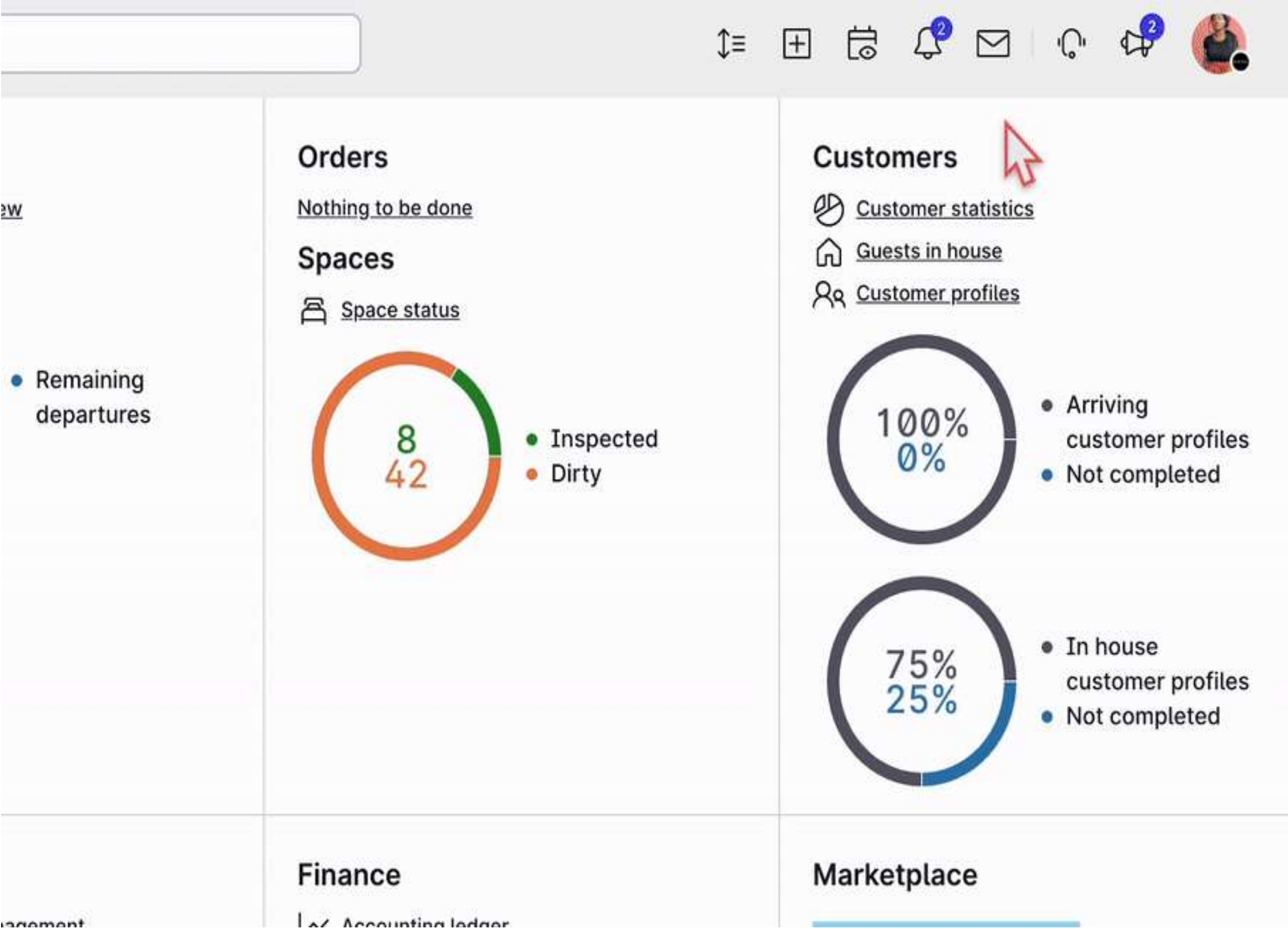


After · Ada goes live

Content read by a model first, the customer second.

Retrieval. Disambiguation. Whatever Ada surfaces is what the customer sees. Structure is no longer optional.

How Ada works.



Six things to audit in a knowledge base before you give it to AI.

01 One topic, one home.

Non-overlapping categories. Every fact has exactly one canonical location.

03 Write for retrieval.

Plain language. Front-loaded answers. No buried leads, no decorative preambles.

05 Metadata earns its keep.

Tags, summaries, version stamps. Anything the retrieval layer can use to disambiguate.

02 Structure the model can parse.

Descriptive H1 / H2 / H3 hierarchy. Proper semantic HTML. Beautiful prose isn't enough.

04 Priorities, set by data.

Support tickets, user journeys, conversation logs — not what someone thought to ask.

06 Audit relentlessly.

Stale content is worse than missing content. A confident wrong answer is the failure mode.

1,500 articles. We couldn't do it all.

So we triaged. We ranked by ticket volume, retrieval likelihood, and structural debt — and rebuilt the slice that mattered first.



It worked. Then it kept working.

EXPECTED

20 %

What we'd have called a win on day one of the rollout.

AT LAUNCH

40 %

Out of the gate. Already double what we'd projected.

WHERE WE GOT

90 %

Containment, against tickets Ada could plausibly handle.

AI for KBs isn't necessarily a seismic shift, if you've been tending your KB well.

Good hygiene compounded. The work we'd done over years — clean taxonomy, one topic per article, consistent metadata — had a new payoff overnight. The same discipline that made articles findable for humans made them retrievable for a model.

The traffic mix flipped.

01

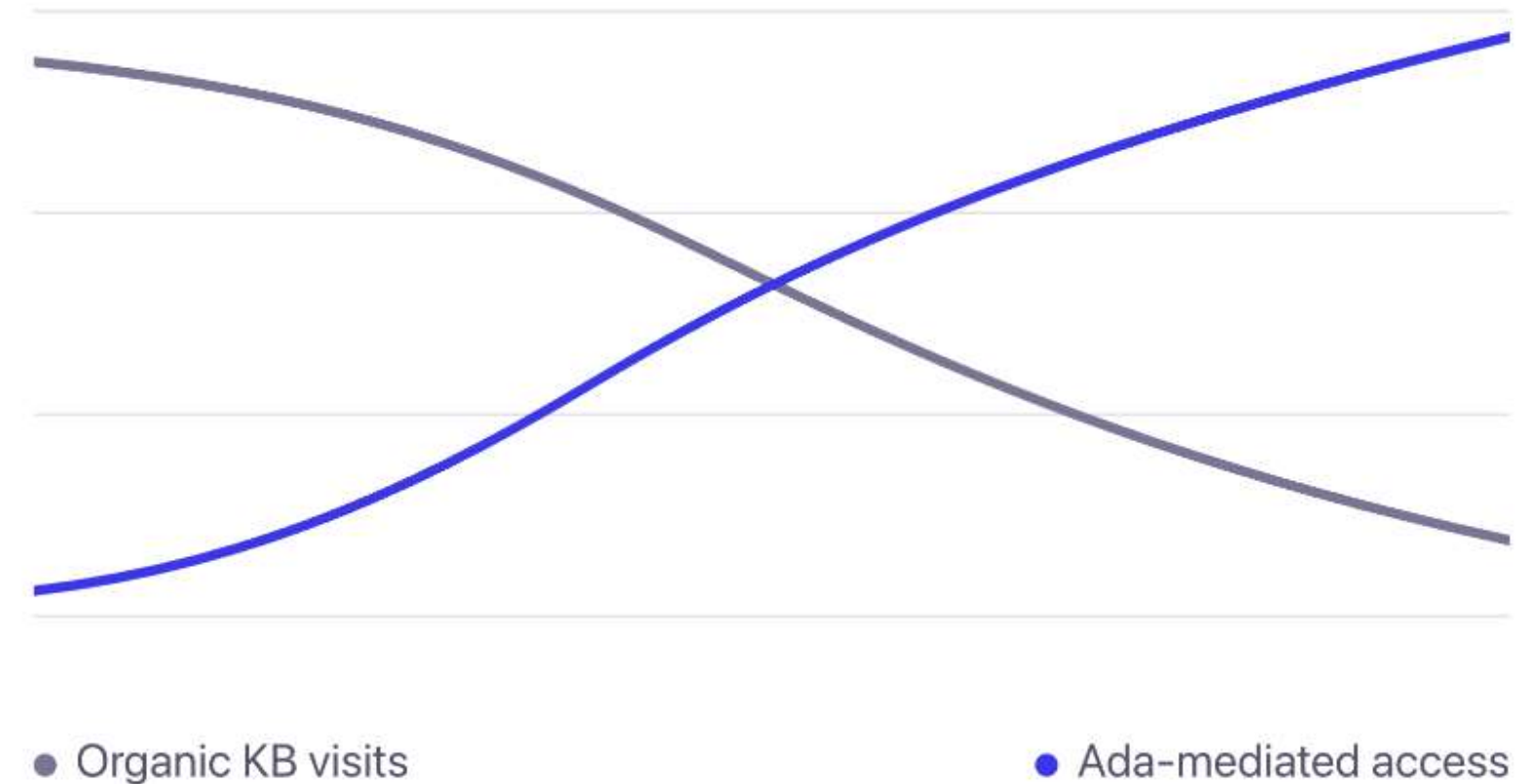
Organic KB traffic started **dropping**. Fewer customers were arriving at the docs through search and navigation.

02

More were reaching the same content **through Ada** — never seeing the article we wrote, just the answer it produced.

The KB stopped being a destination. It became a substrate.

Q1 2024 → Q1 2026 · ILLUSTRATIVE
Where customers meet our knowledge.



The past was forgiving. The future won't be.

Good KB hygiene got us to 90%. That's a real number. But it was bought against a generation of AI that's already over.

We're not designing content for people anymore. We're designing it for the machines that read on people's behalf. And soon, for the machines that talk to other machines.

WHAT THIS MEANS FOR PRODUCT CONTENT

Same content. New job.

Content used to be a deliverable — published, indexed, done. Now it’s the layer every assistant in your stack is reading from.

SINGLE SOURCE

The Knowledge
Base

- A Ada
- B Mews University
- C In-product guidance
- D Internal tools

Once content was the
substrate, we started
building on top of it.

Three tools, all of them AI, all of them feeding the same knowledge layer from a different angle. The next three slides are what we ship and what each one does.

AI analyzing AI's own failures.

Every Ada conversation feeds a structured pipeline that finds the next knowledge gap to close: KB problems and product problems, sorted into the right inbox.

01 · PULL

Pull every conversation.

By product — PMS, POS, RMS, EMS. The full corpus, every month. No sampling, no shortcuts.

02 · ANALYZE

Run the prompt pack.

Hot topics. Knowledge gaps. Handoff trends. CSAT root cause by language. Same templates, every month, every product.

03 · SHIP

Two artifacts.

A long analysis doc — escalations, clusters, sentiment. And a short product brief that lands on a PM's desk each month.

Same pipeline finds KB gaps (our problem) *and* product gaps (engineering's problem). The line between “content issue” and “product issue” is the most useful thing the pipeline draws.

AI inside the authoring flow.

Drafts release notes and in-app messages directly from Jira — humans review, AI accelerates. The loop is two passes.

01 Pull context.

Ticket summary, description, reporter, approver — straight from Jira, no copy-paste.

02 Draft v1.

A ready-to-review draft lands in Confluence — tagged, timestamped, marked for human review.

03 SME marks it up.

Real reviewer, real corrections, real SME notes — inline, in the document, where edits already live.

04 Second pass.

Console re-applies resolved edits. Unresolved ones get flagged with [QUERY: ...] — never silently dropped.



The discovery agent.

Launched April 2026. Plain-language search across Mews University, in every supported language; no taxonomy memorization required.

TODAY

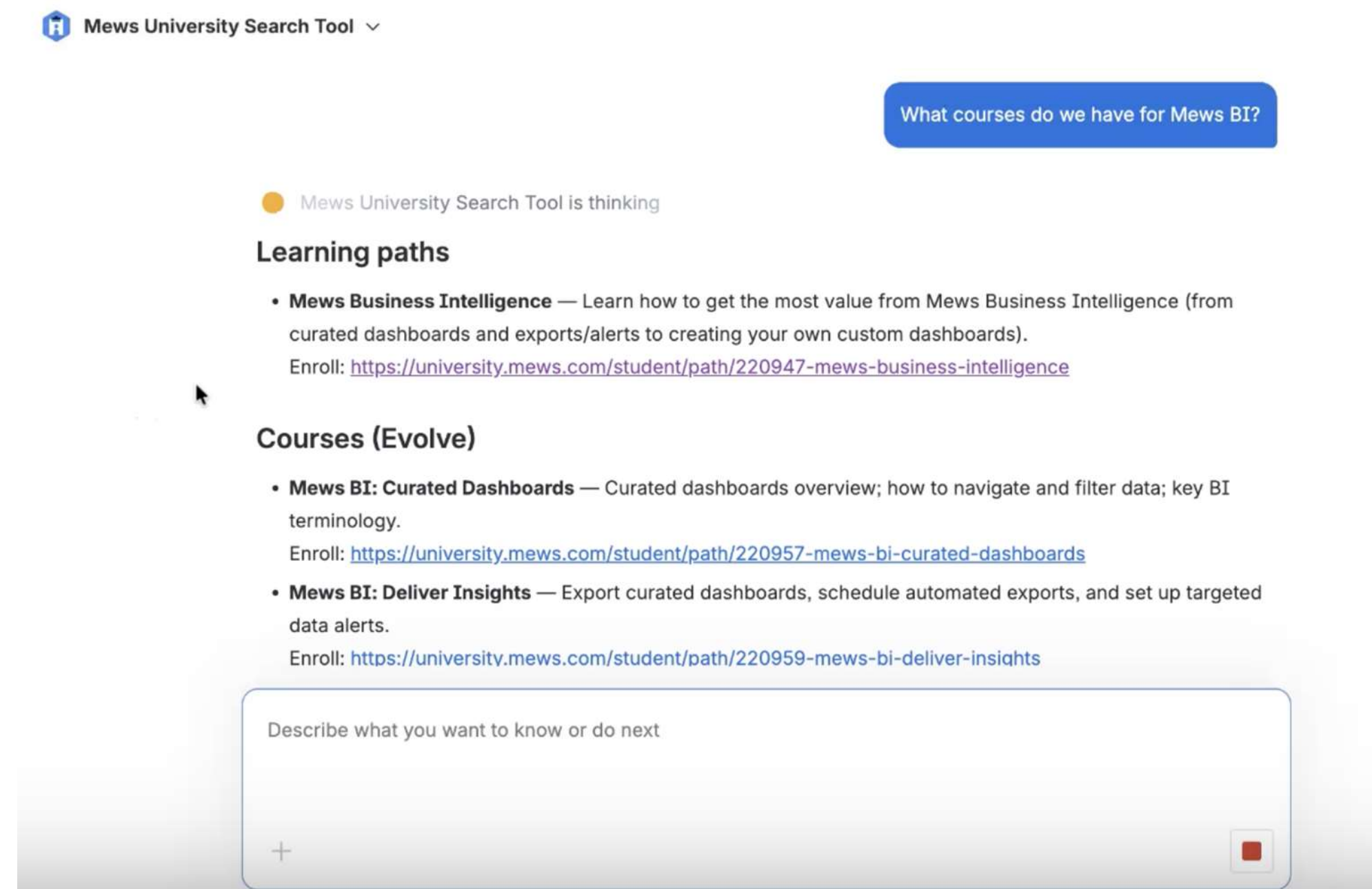
Live for everyone.

Mewsters query Mews University in plain language. No course-code lookups, no taxonomy gymnastics.

NEXT

Surfaced inside Ada.

A customer hits a gap. Ada doesn't just answer — it recommends the course. Containment becomes upskilling.



Four principles, in priority order.

01

Knowledge is a product.

Treat the KB the way you treat a feature — owned, versioned, instrumented, refreshed. Publication isn't a finish line.

02

AI sits at both ends of the loop.

One AI delivers the knowledge. Another finds the gaps. A third drafts the next note. Different jobs, one flywheel.

03

Trained customers ask for help less.

Upstream investment shows up downstream. The containment number is the leading indicator that your knowledge layer is doing its actual job — not a marketing claim.

04

Ship the gap-finder before the audit.

An AI that hallucinates is loud. An AI that knows what it doesn't know is useful. Measure the gap first, then fill it.

Where this goes from here.

01 · INSIGHT AUTOMATION

Ada × Glean.

The Claude pipeline runs across both knowledge sources. The loop gets wider without getting slower.

02 · MCP WITH WRITES

From diagnostics to action.

From read-only diagnostics to safely-scoped writes — Claude opens the Jira ticket the analysis just identified.

03 · MUST INSIDE ADA

Containment becomes upskilling.

A customer hits a gap. Ada doesn't just answer — it recommends a course. Knowledge that follows the user, not the other way around.

04 · THE HORIZON

AI talking to AI.

We won't be designing content for human readers much longer. We'll be designing it for the machines that read on their behalf.

Writers become content engineers.

The job moves up the stack. The thing we ship stops being the article — and starts being the system that drafts it, delivers it, and learns from how it lands. The byline shrinks. The leverage grows.

01 · Design

The systems.

The retrieval pipeline, the prompt pack, the loop that closes itself. Architecture, not just articles.

02 · Tune

The prompts.

Voice, structure, edge cases. The taste lives in the prompt now — and in the eval that catches it when it slips.

03 · Measure

The output.

Containment, sentiment, gap reports. Ship the gap-finder, read the result, write the next brief.

The KB used to be where answers
lived. Now it's where every AI in
your stack learns. Treat it
accordingly.

Grab the paintbrush.

Questions?

