SPECIMEN · AI AUDIT REPORT

A full AI audit report, on a company that does not exist

This is the deliverable of the AI Audit format, published as it stands. Five candidate uses examined, one started, one deferred, two ruled out, and one that is not a generative AI problem at all. With the arithmetic shown and the technical annexes.

FABRICATED CASE

This company does not exist

FABRICATED CASE

This company does not exist

I don't publish client reports, not even anonymized ones. An audit describes a real company's organization, its slow points and its trade-offs. That is not something to expose because it would flatter my site.

So the case that follows is fabricated end to end. The company, the figures, the interviews, the abandoned project: all invented. What is not invented is the shape of the document, the order of the questions, the way a verdict is argued and the level of detail in the annexes.

Every figure here is a working assumption. Where a figure is a result, its terms are written next to it, so you can redo the sum and argue with the assumption rather than with the total.

THE SUBJECT

The distributor

B2B distribution of industrial supplies: bearings, fasteners, protective equipment, tooling. Customers are manufacturers, maintenance workshops and public bodies.

HEADCOUNT 380 people	FOOTPRINT 4 branches, 1 warehouse
CATALOG 34,000 active part numbers	ORDERS about 4,800 a month
ORDER DESK 14 people	SYSTEMS a 2013 ERP, a separate CRM, a partial PIM
DOCUMENTATION 34,000 supplier PDFs, about 40% scans	HISTORY a chatbot stopped after 4 months in 2025

THE SHORT VERSION

What I recommend, and what I turn down

Five candidate uses were examined. One starts now. A second starts after a piece of work that has nothing to do with AI and has to come first. A third waits for the first one to hold its thresholds. The last two don't happen, and one of them is not a generative AI problem at all.

The rest of the report argues each of those five verdicts, shows the arithmetic behind them, and gives the target architecture for the two projects kept.

REF.	CANDIDATE USE	VERDICT	NEXT STEP	ESTIMATED GAIN
C1	Keying in orders received by email A structured-file integration covers 62% of order lines for less money and without ever getting one wrong. AI takes the tail, about 900 documents a month.	Yes, after the integration	12 d, after the integration	about 630 h a year
C2	Search across the technical documentation Bounded scope, an answer you can verify because it is sourced, and an evaluation set you can build today out of questions support has already been asked.	Start here	4 d of scoping	about 1,320 h a year
C3	Help drafting tender responses This one assumes document search is running. Without it you get unsourced drafting, which is what a tender tolerates least.	Later	to revisit	not costed
C4	Customer chatbot on the site The three causes of the 2025 failure have not moved: no source of truth on stock and prices, no evaluation set, no escalation to a human.	No, not now	none	none
C5	Stock-out forecasting This is statistics on ERP history, and that history has a seven month hole. The work to do here is data work.	Not a generative AI problem	a data project	not costed

EFFORT IS IN WORKING DAYS, NEVER IN EUROS. GAINS ARE IN HOURS A YEAR, WITH THE ARITHMETIC IN SECTION 05: YOUR LOADED HOURLY COST IS SOMETHING YOU KNOW BETTER THAN I DO.

CONTENTS

The report

01 · FABRICATED CASE

The brief, and what I looked at

The scope, the method, and what four days cannot tell you.

The distributor's leadership put it this way: "Everyone talks about AI: where does it actually help here, and where do we start?" As it stands the question has no answer, because it is about a technology and not about a job. So I rewrote it as an **auditable question**, one that four days of work can answer: of the uses your teams raise on their own, which hold up against the data that exists today, at what cost in effort, and in what order should you take them?

Over those four days I ran 11 interviews, from the managing director to the warehouse operator, and 3 observations at the order desk. An interview tells you what people believe they do; an observation shows the moves they no longer think to mention: the re-keying, the back and forth between two screens, the checks that sit in no procedure. I also went through 2 ERP exports covering 12 months and sampled 400 documents at random.

HOW THE DAYS RAN

DAY	WHAT I LOOK AT	WHAT IT PRODUCES
Day 1	Interviews, how orders flow	List of candidate uses
Day 2	Order desk observations, tools at the desk	Real volumes and moves
Day 3	ERP exports, document sample	State of the available data
Day 4	Effort, risks, sequencing	One verdict per track
Readout	Review with the people I met	Annotated report, decisions recorded

The fourth day adds no new material; it sizes the effort and puts the five tracks in order. The readout happens in front of the people I interviewed, and that is the only filter that counts: if a sentence does not hold up in front of the person who does that work every day, it goes. The wider frame is in my method.

What this audit does not say

- This is not a security audit: I look at where the data sits, not at whether it is properly protected.
- This is not a review of the ERP code: I take the system as it runs, without judging how it is written.
- This is not a user test: three observations show moves, they do not measure adoption.
- A sample of 400 documents gives **orders of magnitude**, never rates: read the percentages in this report as markers for discussion.

> CAVEAT
>
> 400 documents drawn at random is not many. A second draw would give different numbers, and I did not make one. When I write that 40% of supplier documents are scans, read "between a third and a half". If a decision of yours turns on a few percentage points, my figures do not settle it: you have to count, and counting takes more than four days.

02 · FABRICATED CASE

Where things stand

Where the order desk's time goes, what state the data is in, and what the 2025 POC taught.

What I describe here comes from 11 interviews, 3 observation sessions at the order desk, 2 ERP exports covering 12 months and 400 documents drawn at random. That is not enough to know everything. It is enough to see where the work is lost. I describe how things actually run, not how the procedures say they run.

The path an order takes

An order arrives by email. Someone opens the message, then the attachment: a PDF, a spreadsheet, sometimes a few lines in the body of the message. They read it, retype the part numbers one by one into the ERP, check the price, send a confirmation back to the customer. That path fills a large share of the day for the 14 people on the order desk.

Of about 4,800 orders a month, **61% arrive by email**, 27% through the web portal and 12% by phone. That split decides what follows: the 27% from the portal already arrive structured and are never retyped, and the 12% by phone are not a job for a language model. The whole subject sits inside the 61%, and nowhere else.

The technical documentation

The distributor keeps 34,000 supplier PDF datasheets. From one supplier to the next, nothing matches: the structure, the vocabulary, the units. About 40% of these datasheets are scans, page images with no text inside.

The consequence is concrete. A keyword search cannot see those 40%. A salesperson looking for a diameter or a standard in a scanned datasheet finds nothing, while the information sits right there on screen. So they open them one by one. This is the point the interviews raise most often.

The state of the data

Two facts weigh more than the rest. The ERP history carries a **7 month hole in 2022**, at the time of the migration: the movements from that period were never carried over. That hole rules out one precise thing: statistical sales forecasting. A seasonality calculation needs complete years to compare a month with itself, and 2022 is no longer a complete year.

The PIM, the repository that describes the products, is partial and filled in by hand. It rules out something else: serving as a source of truth. No assistant can state a price, a stock level or a specification on that basis, because nothing says whether the record was ever updated. I have written elsewhere about what makes a dataset usable: Is your data ready for AI?.

The 2025 POC

A customer chatbot was ordered from an agency in 2025, then stopped after 4 months. I hold nothing against that agency: the mistake is a common one, and it is structural. Three causes stack up, and none of them comes down to the quality of the work delivered.

— No source of truth was exposed for stock and prices: the chatbot answered with nothing to check against.

- No evaluation set existed, so nobody could say whether it answered correctly, or see any progress from one version to the next.
- No escalation path to a human was planned, so a stuck customer stayed stuck.

> This POC did not fail because of the model. It failed because it was asked to answer on data it had never been given, with no way of knowing when it was wrong.

These three gaps can be closed, and they get closed outside the chatbot. That is why, further on, I propose starting with something else.

03 · FABRICATED CASE

What I start, and in what order

One project to start now, a second to start after an integration that has nothing to do with AI.

The order below is not a merit ranking, it is an execution order. C1 makes more noise inside the company: order entry occupies the order desk every day, and it is the first subject raised in the interviews. C2 still goes first, for a simple reason. It is the only one of the five cases whose quality can be checked before it reaches anyone's hands, and on ground where a mistake never reaches a customer. Starting there means learning to measure on a scope with no consequences, before applying the same discipline to work that touches real orders.

C2: searching the technical documentation

This pain is named in 9 of the 11 interviews, unprompted. A technician looks for a dimension, a tightening torque, an equivalence between two part numbers. They open three datasheets, they call a colleague, they end up calling the supplier. Nobody can say how long it takes. Everybody can say it happens several times a day.

The assistant does one thing and one thing only: it finds the useful passages in the 34,000 supplier datasheets, and it answers by citing the datasheet and the page. It writes no sales pitch, it guesses no price, it does not replace the supplier. When it finds nothing, it says so. That is the pattern described in the document assistant.

Three reasons make it the first project, and the verdict is **yes, start here.** The scope is bounded: one corpus, one question, one answer. The answer is checkable because it is sourced: whoever reads it opens the cited datasheet and settles it in ten seconds. And the evaluation set, meaning the list of questions whose right answer is known, can be built right now. Support has already received those questions. They sit in the tickets and the mailboxes, they only need gathering.

What would make it fail is known in advance. About 40% of the datasheets are scans, images with no usable text. If the image reading step is rushed, the assistant will give wrong answers on that share of the corpus, and never flag that it does not know. Second risk, a scope that widens along the way to take in prices, stock and commercial terms, three subjects whose source of truth sits elsewhere. One point still needs checking before launch: what the supplier contracts allow on internal indexing.

C1: entering orders received by email

The verdict is **yes, but not as planned.** The initial request was an AI that reads every order arriving by email. The ERP export says otherwise: 62% of order lines come from 23 regular customers. Those 23 customers can send a structured file, a format the ERP reads on its own. That leaves 38% of the lines, about 900 documents a month, in formats no integration will ever cover: scans, improvised spreadsheets, orders typed into the body of the message. That, and only that, is where AI has a role.

On the integrable share, a plain file integration beats AI on the three criteria that matter. It costs less to set up and to run. It is never wrong, because it infers nothing: either the file conforms, or it is rejected. And it consumes nothing in use, not one token. Putting a language model where a file format is enough means paying for a read on every document, for work that a mapping between fields settles once and for all.

THE TWO ROUTES FOR C1, AND WHAT EACH ONE COVERS

ROUTE	WHAT IT COVERS	RUNNING COST	GETS IT WRONG
Structured file	62% of lines, 23 customers	None once set up	No, the file passes or it is rejected
AI extraction	38% of lines, 900 documents a month	Billed per document processed	Yes, a rate to measure and to watch

The measurement rule is set before the first trial, not after. The model's success rate is not what gets measured: that number means nothing to the order desk, which counts orders, not fields. What gets measured is the share of orders integrated with no human rework, on a sample reviewed line by line. That single number decides whether to continue, adjust or stop. The document reading chain is covered in detail in this guide.

So the order is: file integration for the 23 customers first, AI extraction second, on the tail alone. Reversing the two means having a machine read documents that should never have been documents.

> Part of the work in an AI project is not AI, and it is often the part that produces the gain.

04 · FABRICATED CASE

What I defer, and what I rule out

One case that depends on another, one failure not to repeat, and one need that calls for no language model.

Three of the five tracks do not start now. Two are deferred, one is ruled out in its current form. None of them is held back because the subject is uninteresting, but because a condition is missing, and that condition can be named.

C3: Tender response support

Verdict: **later**, and that is not a refusal. Answering a tender means producing a document where every technical claim commits the distributor. A language model is very good at writing a plausible paragraph. Without a reliable retrieval base behind it, it produces unsourced prose, exactly the opposite of what a tender demands. The risk is not lost time, it is stating a false product characteristic and being held to it.

What unblocks it is precise: C2 running in production, with sourced answers and an evaluation set kept up to date. The day a writer asks for a characteristic and gets back the supplier sheet that carries it, writing support becomes a thin layer on a sound base. Before that day, C3 is not a project, it is a bet.

C4: Customer chatbot on the website

Verdict: **no, not now**. A no about timing, not about principle. The 2025 POC stopped after four months for three reasons, and those three reasons have not moved. No source of truth is exposed for stock and prices. No evaluation set exists, so nobody can say whether the tool answered correctly. No procedure hands over to a human when the machine leaves its scope. Restarting without settling those three points means repeating the failure with a newer model.

The three conditions for reopening are checked, not declared.

- Stock and prices are machine-readable, in real time, through an interface supplied by the ERP vendor, and that interface returns the same value as a salesperson's screen.
- An evaluation set of at least two hundred real customer questions exists, with the right answer written beside each one, and the share of correct answers is measured before anything goes live.
- An escalation rule is written and tested: after two failures, or on any negotiated price question, the conversation goes to the order desk with its history.

Transparency comes on top: the European AI regulation requires the user to know they are talking to a machine. Of the five cases, this public chatbot is the only one concerned. One sentence on screen is enough, it is not a project. The framework is set out in the EU AI Act for businesses.

C5: Stockout forecasting

Verdict: **this is not generative AI.** Forecasting a stockout means reading outbound history, supplier lead times and seasonality, then applying statistics. A language model adds nothing here, and it introduces a source of error where the subject calls for steadiness. The real obstacle is elsewhere: the ERP history has a seven-month hole in 2022, at the migration. A forecast fitted on an incomplete year gets it wrong with confidence.

Instead: deal with that hole, set a simple calculation method on the references that matter, and measure the gap between forecast and actual for a quarter before automating anything. It sells less well than an AI project. It is more useful, and that data work then serves every other case, including the ones we start. The outline is in is your data ready for AI.

> A case ruled out with a written condition is worth more than a case launched without knowing what would make it fail.

WHAT WOULD CHANGE MY MIND

Three conditions, and you can check them without me. For C3, C2 has been running in production for a quarter and its rate of correct, sourced answers is measured. For C4, the three points above are ticked and the order desk has approved the escalation rule. For C5, the 2022 hole is dealt with and twelve continuous months of history are available. If one of these conditions falls sooner than expected, that case moves ahead of the others.

05 · FABRICATED CASE

What it returns, what it costs

Hours a year, never euros: the arithmetic is laid out, the hourly cost is yours.

This report converts no gain into euros. I do not know your loaded hourly cost, on the order desk or in support. You know it, and so does your finance department. An invented amount in an audit report discredits everything around it, including the parts that hold. So I give hours per year, with the terms of the calculation set next to the result. The multiplication is yours: it takes ten seconds and it produces a number you can defend.

Two workstreams already carry a gain you can quantify: C2 on document search, C1 on the tail of orders arriving by email.

C2 GAIN, DOCUMENT SEARCH, IN HOURS PER YEAR

Document searches per day, all departments	120
Share the assistant handles without rework, assumption	60%
Searches in scope per day	72
Average time today, target time	7 min, then 2 min
Time saved per search	5 min
Gain per day	6 hours
Working days per year	220
Annual gain	about 1,320 hours

This number is worth exactly what the 60% assumption is worth, and that share has not been measured: the prototype evaluation set will confirm it or refute it. It is not a promise, and it says nothing about the implementation effort.

C1 GAIN ON THE TAIL, ORDER EXTRACTION, IN HOURS PER YEAR

Order documents from customers not eligible for integration, per month	900
Share handled without rework, assumption	70%
Documents in scope per month	630
Keying time today, check after extraction	6 min, then 1 min
Time saved per document	5 min
Gain per month	52.5 hours
Months per year	12
Annual gain	about 630 hours

This number assumes the 23 regular customers send a structured file first: without that step the tail is not isolated and the calculation collapses. It does not cover the 62% of lines taken over by integration, which is not an AI workstream.

These hours do not turn into cut jobs, and I will not sell you the opposite. 630 hours a year spread over four branches and 14 people on the order desk comes to a few hours a week per site. What they become is more useful: orders keyed the same day instead of the next morning, a customer called back before noon, and the retyping nobody ever liked doing, gone. If your goal is to cut headcount, say so now, because that changes how the project is run, and I will not present it to the teams under another name.

> An hour saved becomes a response time before it becomes a budget line.

Running cost is computed from volumes, not from a flat fee. C2 sends about 13.2 million input tokens and 0.88 million output tokens a month, the token being the unit of text the model bills. C1 sends about 4.86 million input and 0.54 million output. The formula is stable: cost = (input tokens in millions) x P_input + (output tokens in millions) x P_output. P_input and P_output are the published prices per million tokens of the model you pick, on the day you decide. I do not write them here: they move several times a year, and a rate copied into a report is wrong before it is read. Take the vendor's price of the day, put it in the formula, and add hosting and supervision, which are not tokens. The detail workstream by workstream, including the first indexing of the product records, is in appendix A4, and the levers that bring the bill down are covered here.

Implementation effort follows the same rule: the plan states it in days, never in euros. I publish no rate card, and the pricing page says why: a price posted blind would be wrong for your case. You get the number on the first call, before any commitment.

> ASSUMPTION
>
> This whole case is fabricated. The distributor does not exist and none of these figures was measured at a client. The volumes and the 60% and 70% shares are working assumptions, set down to show the shape of a calculation. What matters is not the result, it is the structure: a volume, a share handled without rework, a time before, a time after. At your company the terms change, the structure holds.

06 · FABRICATED CASE

Risk, compliance and adoption

What the regulation actually asks for here, and the risk everyone underrates.

Four risks can stop one of these projects after it starts: regulation, supplier contracts, adoption, and dependency on one provider. The last three weigh more here than the first.

What regulation asks for here

The European regulation on AI, known as the "AI Act", sets a **transparency** obligation: a user must know they are talking to a machine. It applies to case C4, the assistant open to customers on your site. If you restart it one day, it has to say it is a machine, and the handover to a human has to stay visible.

The two projects I recommend starting fall outside that scope. Internal document search, C2, and order extraction, C1, are **minimal risk**: the regulation puts no specific obligation on them. The point to watch is elsewhere. Orders carry personal data, a buyer's name and contact details for example, and the GDPR applies to them as it already applies to your ERP. This is not a new subject, just the same one extended to one more processing operation. For the general framework, see the EU AI Act for businesses.

> CAVEAT
>
> I am not a lawyer. The above names the points to look into, it is not a legal opinion. Validate them with your counsel.

Supplier catalogs

The 34,000 datasheets come from your suppliers. Indexing them so your teams can search them internally is not the same as redistributing them to your customers. The first reading is defensible, the second is not without an agreement. I will not settle the question for you: it has to be read contract by contract, and your 11 main suppliers do not have the same usage terms. The answer belongs to your legal department, or to your outside counsel. Ask before the prototype, not after.

Adoption

This is the most underestimated risk on the list. An assistant nobody opens after three weeks has cost what it took to build and returned nothing. The POC stopped in 2025 belongs to that category, and getting from POC to production is largely decided there. What answers it is nothing spectacular, and it is decided before the first day of code.

- The assistant opens inside the tool already used every day, not in one more tab to remember.
- Three people from the order desk and two from support try the prototype and say what is wrong, before any wide rollout.
- Every answer shows its sources, because an answer you cannot check ends up unread.
- Weekly active users are tracked from day one, and so is answer quality.
- A stop condition is written in advance: if usage does not take off in two months, we stop and we say why.

Dependency

Open formats, data exportable at any time, and both the report and the code produced during the engagement belong to you.

RISK REGISTER

RISK	WHAT IT PRODUCES	WHAT ANSWERS IT
Undetected wrong answer	Decision made on bad data	Sources shown, evaluation set before rollout
Rights on the catalogs	Internal use challenged by a supplier	Contract by contract review before the prototype
Personal data in orders	Processing outside the GDPR framework	Register kept current, restricted access, hosting settled
Teams drop it	Effort spent, no gain	Pilot users, usage measured, stop condition
Provider dependency	Exit cost out of control	Open standards, code and documentation delivered

> On this engagement, the regulatory risk is the best documented and the least likely. The risk of abandonment is exactly the reverse.

07 · FABRICATED CASE

The roadmap

Three horizons, effort in days, one start condition per project and a set of stop criteria.

This roadmap commits to two things: an order between the workstreams, and a starting condition for each. It does not commit to a result. Nobody can promise today that the document assistant will hold the target threshold, because the evaluation set that would let you check it does not exist yet. What I can set out is the order in which the uncertainties get cleared, from the cheapest to the most expensive.

The order follows a simple rule: no step is funded until the previous one has produced its proof. Effort is counted in implementation days. It excludes your teams' time, hosting, and model usage, which is worked out separately.

THE THREE HORIZONS, WITH WHAT ALLOWS EACH WORKSTREAM TO START.

HORIZON	WORKSTREAM	EFFORT	STARTING CONDITION
0 to 3 months	Scoping and evaluation set for C2	4 days	A named document owner, with time freed up
0 to 3 months	C2 prototype	12 days	Evaluation set frozen and accepted by the business
0 to 3 months	Structured files for the 23 customers	Outside the AI scope	Quote and date obtained from the ERP vendor
3 to 6 months	C2 into production	20 days and up	Thresholds met by the prototype on the evaluation set
3 to 6 months	C1 extraction prototype	12 days	Integration of the 23 customers delivered or scheduled
6 to 12 months	C1 into production	20 days and up	Thresholds met by the extraction prototype
6 to 12 months	C3, answering tenders	Not costed at this stage	C2 holds its thresholds in live production

The workstreams in the first three months run in parallel, but they do not draw on the same people. Integrating the files of the 23 customers is an ERP matter, not an AI matter: it belongs to the vendor. On its own it covers 62% of order lines, with no model involved. If it stalls, the extraction prototype loses part of its purpose, since the AI would end up handling volume that should never reach it.

Stopping criteria

A workstream is better stopped early than late. Here are the points to check, and what should stop the work rather than extend it.

— At the end of the 4 scoping days, if the questions already sent to support are not enough to build an evaluation set the business accepts, the prototype is not commissioned.

- At the end of the C2 prototype, if the assistant stays below the threshold set at scoping, the move into production, 20 days and up, is not commissioned.
- At the closing review, if support staff stop opening the prototype unless reminded, the need was not where we thought.
- In production, if the share of answers reworked by hand rises two months running, we freeze new features and go back to the evaluation set before adding anything.
- Before the C1 prototype, if the ERP vendor has given neither quote nor date for the structured files, the AI part does not start.
- At each monthly review, if the cost recomputed at current rates goes over the envelope set at scoping, we switch models or we stop; we do not let it drift.

> A workstream with no written stopping condition has no serious starting condition.

> OUT OF SCOPE
>
> C5, stockout forecasting, will never enter this roadmap as a language model. It is statistics on the ERP history, and that history has a 7 month hole in 2022. The workstream exists, it belongs to data quality, and it is handled elsewhere. C4, the customer chatbot, comes back only on the three conditions set above. It needs a source of truth exposed for stock and prices, an evaluation set that says whether it answers correctly, and an escalation path to a human. Until those three are dealt with, restarting it means repeating the 2025 failure.

Splitting the work into scoping, then prototype, then production is not specific to this case: it is how I work, described in my method. Each step ends in a decision, and that decision can be to stop.

TECHNICAL ANNEXES

What follows is written for a technical team. A director doesn't need to read it to decide. Their supplier, though, should be able to argue with it line by line.

A1 · TECHNICAL ANNEXES

Target architecture for document search

Ingestion, chunking, hybrid search, and the hard part, which is a metadata field and not a model.

The architecture described here serves track C2, search across the 34,000 supplier PDF datasheets. It has four layers: ingestion, chunking, retrieval, generation. Each has its own failure mode and its own measurement point. The thread through this appendix is simple: on this corpus, what breaks an answer happens before the model. The general framing is in Enterprise RAG, where to start.

Ingestion pipeline

The first step is an automatic sort. Every document goes through a text-layer test: if direct extraction returns ordered text that is dense enough, the PDF is native and takes the short path. Otherwise it is treated as a scan. About 40% of the datasheets take that path and go through OCR, page by page, block positions preserved. The sort itself becomes metadata: a fragment that came out of OCR does not carry the same confidence. The trade-offs at this step are covered in vision and OCR extraction.

The next layer rebuilds structure: heading hierarchy, column reading order, and above all specification tables. This is the sensitive point. A badly extracted table shifts by a row or a column, the value read stays plausible, so nobody challenges it. On this kind of corpus, **that is the leading cause of a wrong answer**, well ahead of anything the model invents. Table extraction is tested on its own, as a component.

> METHOD
>
> The 400 documents sampled at random during the audit form the basis of the ingestion test set. I annotate the tables of a sample of datasheets by hand, half native, half scanned, and measure the share of cells correctly attached to their header. That figure is produced before the prototype.

Chunking

Chunking follows the sections of the datasheet, not a fixed token size. A fixed size cuts through the middle of a table and separates a value from its column header. The fragment stays readable, no longer means anything, and still comes back in search. Every fragment therefore carries its section heading and its table headers, with an overlap for sections that run onto the next page.

Mandatory metadata

Six fields are mandatory on every fragment. At indexing time the pipeline rejects any fragment missing one, rather than indexing an object that can be neither filtered nor cited.

- Part number: the filter key used most often, since a question almost always starts from one.
- Supplier: it narrows the search and tracks the terms of use of the catalog, to be checked contract by contract.
- Revision code: it tells apart two versions of the same datasheet in the corpus.
- Revision date: it breaks the tie between two codes when the supplier changes its numbering logic.
- Language: the corpus is multilingual, and a fragment in a foreign language is no use at the order desk.

— Page: without it the citation cannot be checked, and an answer that cannot be checked is worth nothing here.

AN INDEXED FRAGMENT AND ITS METADATA · JSON


```
{
  "fragment_id": "fp-018342-rD-p07-t02",
  "document_id": "fp-018342",
  "part_number": "REF-40218-C",
  "supplier": "F07",
  "revision_code": "D",
  "revision_date": "2024-11-18",
  "current_revision": true,
  "language": "en",
  "page": 7,
  "section": "Mechanical characteristics",
  "block_type": "table",
  "extraction_source": "ocr",
  "extraction_confidence": 0.94,
  "column_headers": [
    "characteristic",
    "value",
    "unit",
    "tolerance"
  ],
  "text": "Mechanical characteristics. Basic dynamic load rating: 29.6 kN, tolerance 2%. Maximum operating
temperature: 110 °C, tolerance 5 °C.",
  "tokens": 312
}
```


Retrieval

Retrieval is hybrid: a lexical index and a vector index queried in parallel, then a rerank of the merged candidates. Lexical alone misses rephrasing. Ask for an operating temperature when the datasheet says thermal range and you get nothing back. Vector alone misses alphanumeric part numbers: two that differ only by a suffix have almost identical vectors, yet they do not fit the same equipment. Lexical treats those strings as exact identifiers. Each covers the other, and that is why retrieval is hybrid.

Generation and refusal

The answer cites the datasheet and the page, as structured output, not free text. An answer without a usable citation is rejected before display. The relevance threshold is explicit: if no fragment clears it after reranking, the assistant says it found nothing and points to support. It does not compose an answer out of the least bad fragment. An assistant that says it does not know is an assistant you can use, and its error rate becomes measurable.

The hard part

Two revisions of the same datasheet coexist in the corpus, and nothing stops the old one from matching the question better than the new one. That is the main risk in this architecture, and it needs naming: **this is not a model problem.** It is a metadata field and a filtering rule. The rule: index every revision, serve only the most recent one for a given part number by default, and return earlier revisions on explicit request, for equipment already installed.

> An out-of-date datasheet served with an exact citation is more dangerous than no answer at all.

The evaluation set and its thresholds

Sixty questions from real tickets, four measures, one threshold each, and the CI that replays them.

The set holds 60 questions. Not one was written in a workshop. They come from support tickets and from questions the branches have already asked, in their original wording. For each one, the right answer already exists: a supplier datasheet, an archived exchange, the memory of whoever answered. We find it, freeze it, and tie it to a datasheet and its revision.

That origin changes what the measurement means. A question invented in a workshop is clean: right vocabulary, complete reference, context supplied. It tests the system on a distribution that does not exist in the company. A real ticket carries the mistyped part number, the in-house abbreviation, the question asked in two lines. Another effect: nobody argues with a test case that carries a ticket number.

The four measures

Four independent measures, computed on the same run. Of the 60 questions, 45 have an answer in the corpus and 15 do not: those traps serve the third measure.

- **Answer found:** the answer contains every required element listed in the case, value, unit and condition of use.
- **Exact source:** the datasheet cited is the one that carries the answer, in the revision in force on the day of the test. A neighboring datasheet that says the same thing counts as a failure.
- **Refusal when needed:** on a question with no answer in the corpus, the system says it does not know and offers to escalate.
- **Numeric fidelity:** no numeric value absent from the cited passages appears in the answer. The check is automatic.

THE FOUR MEASURES AND THEIR DECISION THRESHOLDS

MEASURE	WHAT IS COUNTED	PASSING THRESHOLD
Answer found	Answerable cases carrying every required element	90% of the 45 cases
Exact source	Answers citing the right datasheet at the right revision	95% of answers produced
Refusal when needed	Traps where the system refuses and escalates	100% of the 15 traps
Numeric fidelity	Answers containing a number absent from the sources	0, otherwise no production

These are decision thresholds, not performance promises. They say what the result is for: below the threshold, nothing goes to production. They are set before the first run, otherwise they end up matching the result and decide nothing.

When the set is replayed

- On every prompt change, including a rewording that looks harmless.
- On every model change or model version change.
- On every index rebuild, especially after a new chunking.
- On every change to a search parameter: passages retrieved, similarity threshold, weighting.
- In continuous integration on every merge, and before any release to production.

Automating the replay is not a convenience. The model vendor ships new versions on its own calendar, and a switch shifts behavior without anything changing on your side. Without a set replayed automatically, that drift goes unnoticed until the first unhappy customer, and by then you have no date, no cause, no comparison.

Maintaining the set

An evaluation set ages without warning. Datasheets get revised, references drop out, suppliers change format. An answer that was right at scoping becomes wrong two quarters later, and the set still counts it as right.

Three rules are enough. Every datasheet cited by a case is watched: if its revision changes, the cases attached to it go into quarantine and are revalidated before the next run. Each month, a few cases come from recent tickets and as many stale ones drop out, so the set keeps its size. Each quarter, a review checks that these questions still look like the ones people ask.

The set has a named owner on the business side: the support person who answers these questions today, the only one who decides what a good answer is. The service provider supplies the tooling, not the truth. With no name on that line, the set dies within months.

> An evaluation set that is never replayed is a photograph of the past, not a measurement.

ONE TEST CASE FROM THE SET · YAML

```yaml
- id: DOC-2026-014
  source_ticket: SUP-48213
  type: answerable
  question: "What tightening torque for the M8 class 8.8 screws in the tooling catalog?"
  expected_answer:
    required_elements:
      - "25"
      - "N.m"
      - "dry thread"
  expected_source:
    datasheet: FT-VIS-M8-88
    revision: "R4"
    page: 3
  expected_refusal: false
  allowed_numbers: [25, 8, 8.8]
  owner: support-order-desk
  last_revalidation: 2026-08-19
```

This is an acceptance set. Monitoring in production is covered in measuring an AI agent in production.

Order extraction: schema and validation loop

A constrained output, a match against the item master, and one rule: no part number is ever guessed.

Extraction does not produce free text. A schema constrains the model output, and validation runs on arrival: every field has a type, a set of allowed values, and a presence rule. The model does not write, it fills in. A non-conforming output is retried once, then the document goes to review. That also fixes the model's scope: it returns what it reads on the document, it decides nothing.

CONSTRAINED OUTPUT SCHEMA, ORDER EXTRACTION · JSON

```
{
  "name": "extracted_order",
  "strict": true,
  "schema": {
    "type": "object",
    "additionalProperties": false,
    "required": ["document_id", "pages", "customer", "lines", "extraction_confidence"],
    "properties": {
      "document_id": { "type": "string" },
      "pages": { "type": "integer", "minimum": 1 },
      "customer": {
        "type": "object",
        "additionalProperties": false,
        "required": ["company_name_read", "customer_reference_read", "confidence"],
        "properties": {
          "company_name_read": { "type": ["string", "null"] },
          "customer_reference_read": { "type": ["string", "null"] },
          "confidence": { "type": "number", "minimum": 0, "maximum": 1 }
        }
      },
      "lines": {
        "type": "array",
        "minItems": 1,
        "items": {
          "type": "object",
          "additionalProperties": false,
          "required": ["page", "source_text", "part_number_read", "description_read", "quantity", "unit",
"confidence"],
          "properties": {
            "page": { "type": "integer", "minimum": 1 },
            "source_text": { "type": "string" },
            "part_number_read": { "type": ["string", "null"] },
            "description_read": { "type": ["string", "null"] },
            "quantity": { "type": ["number", "null"], "exclusiveMinimum": 0 },
            "unit": { "type": ["string", "null"], "enum": ["piece", "box", "meter", "kg", "lot", null] },
            "confidence": { "type": "number", "minimum": 0, "maximum": 1 }
          }
        }
      },
      "extraction_confidence": { "type": "number", "minimum": 0, "maximum": 1 }
    }
  }
}
```

Matching against the reference data

The model chooses no catalogue reference. It returns a string it read, a description, a quantity, a page. Matching comes next, in code, line by line, against the reference data holding 34,000 active references: exact match on the supplier reference, exact match on the customer reference when the reference data knows it, text

similarity on the description. Every candidate comes out with a match score.

If no item clears the threshold, the line goes to human review, with no default candidate offered. **No reference is guessed**, even when a single item is close. An invented reference that reaches an order costs more than ten lines sent to review: a line in review costs one check at the order desk, a wrong reference costs a shipment, a return, a credit note, and a customer who doubts the rest of their order.

> METHOD
>
> The threshold is set on the evaluation set, not by feel. You raise it as long as a wrong reference gets through. You lower it only after a review showing that the rejected lines could all have been matched.

Confidence score and routing

Two scores travel together. The line score combines the model's extraction confidence and the match score. The order score is the **minimum** of its line scores, not their average: an average dilutes the doubtful line among the good ones, and that line is the one you are after.

ROUTING AN EXTRACTED ORDER

SITUATION	DESTINATION
Every line above the threshold	Straight into the ERP
At least one line below the threshold	Human review, whole order
Output not conforming to the schema	One retry, then review
Customer absent from the reference data	Human review, no automatic creation

The whole order goes to review, not just the doubtful line. An order integrated halfway forces the order desk to reopen a partial document and check what already went through: that costs more than handling it in one go.

The measure that counts

The production measure is the **share of orders integrated with no human rework**, taken from the real flow of the month, not the model's score on a benchmark. The gap between the two always runs the same way.

A benchmark scores isolated fields, on chosen documents, often legible and alike, and it tolerates approximation: a description that is almost right counts as right. Production scores whole orders, on the documents that actually arrive, scans included, and an order is reworked as soon as one of its lines is wrong. Errors do not average out, they add up along the order.

WHY A GOOD PER-LINE SCORE GIVES A LOWER ORDER RATE

Share of correct lines, assumption	95%
Lines per order, assumption	8
Calculation	0.95 to the power of 8
Orders with no error at all	about 66%

Illustrative figure, computed on the two assumptions set here and on nothing else. It shows why a per-line score never reads as an order rate, and why the 70% assumption used for C1 stays an order of magnitude, not an announced performance. Measurement protocol in [measuring an AI agent in production](/en/resources/measuring-an-ai-agent-in-production).

The feedback loop

Every human correction is recorded with four elements: the source document, the model output, the correction kept and the reason, taken from a short list, reference not found, quantity misread, line missed, column misassigned. Those four items become a case in the evaluation set, added with no filtering. You do not keep the interesting cases, you keep the ones that happened.

After a few months, that set no longer looks like a generic corpus. It describes this company's awkward cases and no one else's: the supplier whose scans come out skewed, the customer who puts quantities in the right-hand column, the in-house description that matches no catalogue reference. It is the only asset in the project that does not expire when the model changes.

> A model is replaced fast. An evaluation set is built slowly, and it is the one you own.

A4 · TECHNICAL ANNEXES

The cost calculation in detail, and its sensitivity

The formula, its variables, and the five things that move it.

This appendix sets out the token volumes for the two cases selected, the formula that turns them into a monthly bill, and the precise points where that bill moves. The volumes come from the load assumptions in the report. They are not measured on any existing installation.

C2 volumes, document search

The load assumption is about 2,200 searches a month. Each search sends about 6,000 input tokens, the retrieved context attached to the question. It produces about 400 output tokens.

MONTHLY TOKEN VOLUME, CASE C2

Searches a month	2,200
Context sent per search	6,000 input tokens
Answer produced per search	400 output tokens
Monthly volume	13.2 million input tokens, 0.88 million output

Monthly cost = 13.2 x P_input + 0.88 x P_output, where P_input and P_output are the prices per million tokens for the model chosen on the day of the decision.

Input weighs fifteen times as much as output. Any optimization that leaves the context sent untouched therefore acts on the smaller share of the bill.

C1 volumes, extraction on the tail

The assumption covers 900 documents a month, 3 pages on average. A page read as an image counts about 1,800 input tokens. The structured output counts about 600 tokens per document.

MONTHLY TOKEN VOLUME, CASE C1

Documents a month	900
Pages per document	3
Image reading	1,800 input tokens per page
Structured output	600 output tokens per document
Monthly volume	4.86 million input tokens, 0.54 million output

Monthly cost = 4.86 x P_input + 0.54 x P_output, with the same price variables. A manual redo consumes nothing, a second automatic pass counts the whole input again.

Here the ratio of input to output is nine. Image reading holds the dominant term, which makes the scanned share of orders a cost variable, not only a quality one.

The initial indexing

Both amounts above are monthly. The initial indexing is a one-off cost: the 34,000 records go once through image reading for the scanned share, then through a vector computation on every fragment. Two orders of magnitude separate these operations, image reading costing far more than vectors.

That one-off cost is in fact paid several times, and that is the point to remember: every rework of the chunking forces the vectors to be recomputed, and a fix to the OCR chain forces the pages to be read again. In the first few weeks, it happens more often than you expect.

> METHOD
>
> I do not estimate it, I measure it. A thousand records go through during the prototype, keeping the same share of scans as the corpus, and the measurement is extrapolated to the 34,000. The figure covers one full pass: that is what you multiply by the number of reworks you allow yourself.

How sensitive the bill is

WHAT SHIFTS THE MONTHLY AMOUNT

WHAT CHANGES	EFFECT ON THE BILL
Volume doubles	Bill x 2, both terms rise together.
The context sent doubles	Input term x 2, output unchanged.
The model changes tier	Direct factor on P_input and P_output.
The scanned share of orders rises	More pages in image reading, input goes up.
The automatic redo rate rises	A second pass, input and output counted again.

Levers and trade-offs

Four levers act on these amounts. None is free, so I give each one with what it costs in return.

- Caching the stable part of the context cuts the input term on what repeats. In return, that part has to be frozen and its invalidation handled at every update of the corpus.
- Running a smaller model on the first sorting pass lowers the price per million. In return, you need a second pass on the doubtful cases and an evaluation set maintained across two models.
- Processing documents in batches overnight opens up a batch rate, if the provider chosen offers one. In return, orders are no longer handled as they come in.
- Improving reranking to send less context attacks the dominant term of C2. In return, it is one more component to evaluate, and cutting too hard loses the useful passage.

> CAVEAT
>
> A model price is not durable data. It moves, up and down, and tiers get renamed. What I deliver here is the formula and the volumes, not an amount. Check the current price with the provider before any decision.

The mechanics of these levers are detailed in cut your LLM bill. Before any decision, I recalculate with the prices in force that day and the volumes observed on the prototype, not the ones set out here.

What I did not keep, and why

Fine-tuning, a dedicated vector database, multi-agent, automatic writes into the ERP.

An audit is judged as much on what it rules out as on what it keeps. Here are the four options I looked at and then set aside, each with the condition that would put it back on the table.

Fine-tuning

Fine-tuning retrains an existing model on the company's own corpus, so it picks up that vocabulary and answer format. It **does not put a moving catalog inside a model.** The 34,000 references change, supplier sheets get replaced, and a fine-tuned model freezes the catalog as it stood on training day. The problem here is freshness and source citation, not style.

What would make it relevant: a tightly constrained output format, repeated tens of thousands of times, that instructions alone fail to hold, or trade terms the model reads badly despite a well-built context. Measure the gap on the evaluation set, then decide. The full reasoning is in RAG or fine-tuning.

A dedicated vector database

A dedicated vector database is a server specialized in similarity search, with its own indexes, memory and upkeep. The distributor already runs a relational database in production, and its vector extension is enough for a corpus this size. A dedicated component adds a backup, monitoring, version upgrades and a skill to hold, for a latency gain nobody sees here.

The question comes back above a few million indexed passages, or if search becomes a product with its own load peaks. Below a million it stays theoretical. The day it does, the switch is cheap: chunks and vectors get recomputed, the application logic does not move.

A multi-agent architecture

A multi-agent architecture splits the work between several models that hand it along, each with its own role and tools. The C2 task is bounded: one question, one search, one sourced answer. A single agent with a good search tool covers it. Every agent added brings one more call, more latency, more tokens and one more point of failure, for a gain that does not show on a task this narrow.

What would change that: a task that chains unlike decisions, read a specification, find the references that answer it, check availability, then write a sourced reply, each step with its own rework criterion. That is the C3 profile, not C2. We come back to it once C2 is running and we have per-step measurements.

An agent that writes into the ERP

Automatic write-back moves the C1 extraction from proposal to action: the order enters the ERP with no human validating it. **I am ruling that out for now.** One wrong line written without a check produces a wrong delivery, a wrong invoice and a customer dispute, and that cost exceeds the minute of checking it saves.

This door opens on a measurement, not on trust. It takes a stable rework rate, tracked over several months, by document type and customer, and a switch-over rule written in advance: below a threshold we set together, the system writes without validation, above it, the document goes back to human review. Without that series, full automation is a bet.

METHOD

Each of these four refusals carries a condition for reopening it. If the condition is met, the decision gets replayed, without starting the debate over.

> An option ruled out with no condition for return is not a trade-off, it is a preference.

QUESTIONS

About this report, and what it is worth

Is this report a real engagement?

No. The company, the figures and the interviews are fabricated, and the page says so before the report's first line. I don't publish client reports, not even anonymized ones. What is real is the shape of the document, the order of the questions and the level of detail in the annexes.

Why is no gain priced in euros?

Because I don't know your loaded hourly cost and you do. The report gives hours a year and shows the arithmetic; the multiplication is yours. An invented euro figure in an audit report discredits every number around it.

Is a four-day audit enough?

To settle five candidate uses and lay out a roadmap, yes, provided the scope is set on the first call. Four days is not enough to audit the security of an information system, nor to test an interface with its users, and the report says so itself in its first section.

What if the audit concludes you should do nothing?

I tell you, and we stop there. On this case two of the five are ruled out and a third deferred. A no after a few days beats a disappointment after six months.

Do I own the report?

Yes, entirely, and it commits you to nothing. You build with me, in house, or with someone else. Everything is written to be picked up by another team: target architecture, cost assumptions, stop criteria.

Your case won't look like this one

The distributor is a textbook case: four branches, an aging ERP, documentation in PDF. Yours has its own sore points, and the sorting starts from scratch.

What won't change is the method. I look at the real processes, put every candidate up against the data that exists, the gain you can measure and the risk, and I tell you what doesn't pass. The report is yours, whatever happens next.