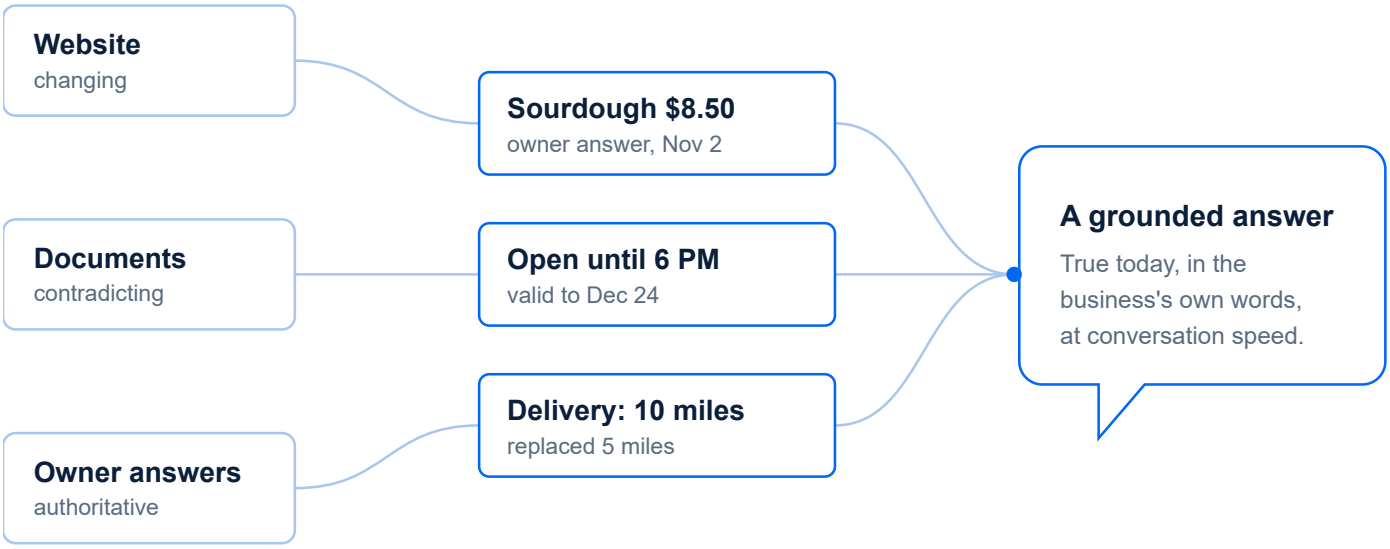


WHITE PAPER

Grounding AI Chat in a Small Business's Own Knowledge

A fact-record architecture and an evidence-based retrieval stack for assistants that must say what the business actually says.

Ron Alterio, Founder

November 2026

EXECUTIVE SUMMARY

Accuracy starts before retrieval

Retrieval-augmented generation is usually treated as a retrieval problem. For an assistant answering in a small business's name, that is only half of it. A business's information changes, contradicts itself and expires, and a system can retrieve perfectly and still answer wrongly if the knowledge underneath has lost track of what is true today.

This paper presents the architecture behind Xorventa's chat windows. Knowledge is stored as dated, sourced, single-detail fact records with explicit precedence, not as text rewritten by a model. Retrieval is built from components chosen by paired measurement, under a real constraint: conversation speed on CPU.

27 of 27

Scenario questions answered correctly by fact records, against 21 of 27 for model-rewritten knowledge.

75% → 95%

Chunk hit on our own knowledge base after the embedding change (14 gained, 2 lost, p = .004).

28 gained, 6 lost

The largest retrieval gain measured, from a small cross-encoder (p = .0002), used where no one is waiting.

32 ms

Median to embed a message on one CPU thread, keeping live-chat retrieval at millisecond cost.

Written for: engineers and technical leaders building or evaluating customer-facing RAG systems.

1. The problem

When a general assistant makes a mistake, the user discounts it. When a business's assistant makes one, it has quoted a wrong price, wrong hours or a service the business does not offer, in that business's name. The bar is not plausible answers. It is the business's actual position, today.

1.1 Knowledge fails before retrieval begins

A small business's knowledge is not a corpus, it is a history. A real business accumulates a website, flyers, price sheets and the owner's direct answers, and they drift apart. Five failure modes follow, and none is a retrieval failure:

- **Precedence.** The owner said $8.50 and a flyer says $10. Which one speaks for the business?
- **Time.** "Open until 6 PM through December 24" is true today and false on December 27, without anyone editing anything.
- **Partial change.** "Delivery within 5 miles for $5" changes to 10 miles. The fee must survive, and that requires knowing they were two facts.
- **Invention.** Summarizing "open Monday to Friday" can produce "closed weekends", a statement the business never made.
- **Regression.** A routine re-crawl of an unchanged website must not undo a correction the owner made last week.

1.2 Retrieval is unforgiving on small corpora

Large corpora have redundancy: if one passage is missed, another may do. In a small business's knowledge base a question often has exactly one right passage, so a single rank position can decide whether the answer exists for the model. Conversation adds to this: "the second one" and "how about yearly?" contain nothing to search for on their own.

1.3 The operating constraints are real

A multi-tenant service for small businesses cannot assume a GPU per customer. Retrieval runs on CPU, in-process, within a conversational latency budget shared by text and voice. Component choices that look free on a benchmark table are not free here.

2. Why the conventional approaches fall short

Common approach	Why it falls short for this problem
Have a model rewrite the sources into clean knowledge	The output is well written and has no memory. It holds no dates, no provenance and no record of which source prevailed, so a contradiction is resolved implicitly, differently from run to run, and invisibly to the owner. We measured this: rewriting chose a flyer's price over the owner's direct answer in every run.
Pick the embedding model from a public benchmark, or an average	Differences between small models are domain-specific. One model moved a single knowledge base by 20 points and three others by less than two. An average reports neither.
Add a larger re-ranker	Bigger was slower and no better. A cross-encoder six times slower returned a lower hit rate than a small one on the same embeddings.
Turn on hybrid search by default	On short, clean business text, rank fusion lost cases that vector search alone answered. Keyword search earns its place on exact identifiers and larger catalogues, not automatically.
List the phrases that count as follow-ups	Any list misses the reply nobody anticipated, and that reply is then answered as if the conversation had not happened.

3. The solution

The architecture separates the two problems. A **knowledge layer** decides what is true today and keeps the evidence. A **retrieval layer** gets the right part of it to the model at conversation speed. A confidence signal between them turns every gap into a question for the owner.

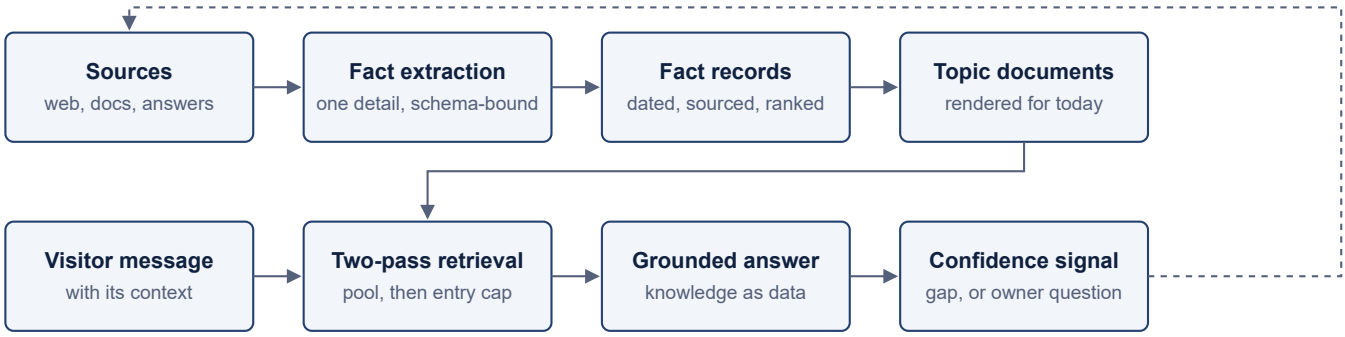


Figure 1. The knowledge layer (top) feeds the retrieval layer (bottom). The dashed path is the learning loop: a gap becomes an owner's question, and the answer becomes a fact.

3.1 The fact record

Every input is treated as the same kind of thing: an owner's answer, an uploaded document, a website page and a business profile form each become fact records.

Field	Purpose
Topic	A short name, reused across sources, so facts on one subject are comparable and render together.
Statement	One self-contained detail. "Delivery within 10 miles for $5" is two facts, so that either can change without disturbing the other.
Learned date	When the system learned it, which orders replacement.
Valid from, valid to	Only when the source says the fact is temporary. A temporary fact never replaces a permanent one, which applies again afterwards.
Source	What it came from, and the source's kind, which determines precedence.
Replaces	The earlier facts this one makes no longer true. A fact that only adds detail replaces nothing.

Extraction is constrained, not trusted. Each piece of text goes through a single fast-model call whose JSON schema is enforced by the API during generation, so the output can only be records. The instructions forbid inference outright: a day, price or policy the text does not mention is unknown, never "closed", "free" or "included". Source text is passed as untrusted data between markers. Numbers, names and terms are copied exactly.

3.2 The rules that decide what is true

Rule	Reasoning
Only what is new is processed. Sources are hashed when read.	An unchanged re-crawl is not information, and so cannot undo a correction. It also keeps extraction cost proportional to change, not to library size.
Newer replaces older on the same subject.	Ordinary currency: a November price update supersedes the old price.
An owner's direct answer is never replaced by a document or the website.	The owner is the authority on their own business. The conflicting value is recorded as a disagreement rather than being applied or discarded.
Retiring a source retires only the facts that came only from it.	Removing a document must not remove something another source also says.
Nothing is deleted. Retired facts keep their reason.	Every change is explainable. An owner can read what changed and why, and reverse a wrong replacement.

3.3 Rendering knowledge for today

The assistant does not read the fact table. It reads one document per topic, rendered from live facts for the date of the question. A fact past its end date is simply not rendered, so temporary hours expire on schedule with no edit and no job to run. Adding a fact re-renders that topic alone, never the whole library. This is the structural difference from rewriting: a change has a defined, small blast radius.

Each topic document carries its own instruction: this is the business's current information on the topic, and anything not listed was not stated. **Inference is deliberately split by stage.** It is forbidden when storing, because a stored inference would become something the business said. It is allowed when answering, only with near certainty, because "No, we're open Monday to Friday" is more useful to a visitor than a refusal. Below that bar, the assistant says it does not have the information.

3.4 Retrieval

- **Embedding.** snowflake-arctic-embed-s, 384 dimensions, with an HNSW index in Postgres (pgvector). The model is asymmetric: questions carry an instruction prefix and documents do not. It runs in process on CPU, baked into the deployed image at a pinned revision and loaded offline. A different revision is a different vector space, so a golden-vector check guards against silent drift.
- **Chunking.** 900 characters with 150 overlap, each piece embedded with its entry's title, which keeps a short piece anchored to its subject.
- **Reading the turn in context.** Every turn with a conversation behind it retrieves twice: once on the message alone, and once on the message blended with recent turns. The results are combined, so nothing has to recognize a follow-up. When the message's own best match reaches .61 it is treated as a question in its own right, and its documents keep reserved places. Measured on our knowledge base, follow-ups such as "the cheaper one" score .49–.59 alone and real questions .62–.74, so the threshold sits between them. Without this reserve, a blended pass on the previous topic can evict every document a new question found.
- **Repair.** When the best match falls below .57, a further pass corrects likely misspellings against the business's own vocabulary and retrieves again.
- **Diversity.** A candidate pool of four times the final count is ranked with at most three pieces from one entry, so that one long entry cannot crowd the others out.
- **Injection defense.** Retrieved knowledge reaches the model between delimiters with a per-turn random nonce, and is treated as data, never as instructions.

3.5 Confidence as a product signal

Top similarity is kept and put to work, not discarded after ranking. Because a model's similarity scale is its own, thresholds are defined by behaviour rather than by number. When the embedding model changes, each threshold moves to the same percentile of the new distribution, so that its meaning survives:

```
t_new = Q_new( F_old( t_old ) )
```

Decision	Threshold	Flags unanswerable	Flags answerable	Decisions unchanged by model change
Record a knowledge gap	.675	70%	18.3%	89.4%
Treat the message as a new question	.61	50%	3.7%	91.2%
Ask the owner	.59	31%	1.3%	93.5%
Repair the query	.57	19%	0.4%	96.5%
Allow a web look-up, where enabled	.55	11%	0%	97.6%

Measured on 240 answerable and 100 unanswerable questions. The thresholds rise in order of cost: recording a gap is free, so it is sensitive; a web look-up is slow, so it fires only on a clear miss.

This closes the loop in Figure 1. A question the knowledge does not cover becomes a question for the owner, and the owner's answer becomes a fact at the highest precedence. The system improves along exactly the lines its visitors ask about.

3.6 Re-ranking where it can be afforded

A cross-encoder is the most effective retrieval component we have measured and the most expensive on CPU. So the decision is about when, not whether: Xorventa applies ms-marco-MiniLM-L6, at a candidate pool of two times and 256 pieces, to email and direct-message drafts. No one is waiting on those, and it fails open if it is unavailable.

4. The evidence

4.1 How we test

Question	Candidates	Method
How should knowledge be stored?	Model-rewritten entries; fact records	One business's history played through both (Appendix A): nine events and nine questions, three runs, one model at temperature 0, graded by required and forbidden patterns.
Which embedding model?	all-MiniLM-L6-v2, bge-small-v1.5, e5-small-v2, snowflake-arctic-embed-s	240 questions over four knowledge bases, each generated from a sampled passage that serves as its label; production ranking; chunk hit, entry hit and MRR; 25 hand-labelled questions.
Is re-ranking worth it?	ms-marco-MiniLM-L6, bge-reranker-base; pools of 2× to 4×; lengths 256 and 512	The same questions, timed on one CPU thread.
Does keyword search help?	Vector alone; rank fusion at four weights, as introducer and as re-ranker	Hand-labelled questions, on about 60 short entries.
Does it work end to end?	The production path	217 questions through the real chat window, checked for required and forbidden content, grounding and latency.

Paired comparisons use an exact two-sided McNemar test on the questions where two setups disagree, and intervals are 95% Wilson. Test data is Xorventa's own knowledge and demonstration content.

4.2 Knowledge storage

Approach	Correct, three runs	Stale or wrong answers	Owner's corrected price
Dated fact records	**27 of 27**	**0**	**Correct, 3 of 3 runs**
Model-rewritten entries	21 of 27	4	Lost to a flyer, 3 of 3 runs

One business scenario. Rewritten knowledge also gave current hours incorrectly in one of three runs while a temporary schedule applied, and in two of three runs it could not state the hours once that schedule had ended. Which questions it missed varied between runs on identical input.

> **The failure that repeated is the one the design exists to prevent: the owner's direct answer lost to a later document. Fact records returned the owner's price every time and recorded the disagreement.**

4.3 Embedding models

Model	Chunk hit	Entry hit	MRR	Hand-labelled /25
all-MiniLM-L6-v2	83.3%	91.7%	.793	22
bge-small-v1.5	84.6%	91.3%	.806	23
e5-small-v2	86.3%	92.9%	.813	23
snowflake-arctic-embed-s	**87.9%**	**92.9%**	**.835**	**23**

Knowledge base	MiniLM chunk hit	arctic chunk hit	Gained	Lost	p
Xorventa's product knowledge	**75.0%**	**95.0%**	**14**	**2**	**.004**
Help library	70.0%	71.7%	7	6	1.00
Demonstration business A	93.3%	91.7%	4	5	1.00
Demonstration business B	95.0%	93.3%	3	4	1.00

The aggregate result is not significant (27 gained, 16 lost, p = .13), and arctic is not statistically separated from e5 or bge-small. The decisive result is domain-specific, on the knowledge base closest to real use. This is why we test per corpus and read paired results rather than averages. Score separation also improved slightly, with AUC rising from .835 to .852.

4.4 Re-ranking

Setup, one CPU thread	Chunk hit (95% CI)	Gained / lost vs MiniLM	p	Median ms	p95 ms
MiniLM	83.3% (78.1–87.5)	—	—	12	24
arctic	87.9% (83.2–91.5)	27 / 16	.13	32	59
arctic + ms-marco, pool 2×, length 256	**92.5% (88.5–95.2)**	**28 / 6**	**.0002**	**1,284**	**2,158**
arctic + ms-marco, pool 3×, length 256	92.5% (88.5–95.2)	27 / 5	.0001	1,954	3,722
arctic + ms-marco, pool 4×, length 512	92.9% (89.0–95.5)	28 / 5	.0001	3,076	4,985
MiniLM + ms-marco, pool 2×, length 256	87.9% (83.2–91.5)	14 / 3	.013	913	2,146

Pool 2× at length 256 keeps almost all the gain at 42% of the latency of the largest setup. The larger bge-reranker-base took 7.5–8 s per query with a lower chunk hit than ms-marco on the same embeddings: 87.9% against 90.0% on MiniLM, and 88.8% against 92.9% on bge-small.

4.5 Keyword search and end-to-end results

Rank fusion scored 23 of 25 at every weight and in both roles, against 25 of 25 for vector search alone. On the production path, the most recent full live run passed 144 of 150 questions, and the 25 hand-labelled questions all passed.

5. How this performs in a chat window

Every design choice above lands in one number the visitor feels: how long the answer takes. These are the costs that matter.

Step	Time	Where it applies
Embed a message, one CPU thread	32 ms median, 59 ms p95	Every turn, and once more with a conversation behind it
Re-rank candidates	1.3 s median, 2.2 s p95	Unattended email and DM drafts only
Web look-up	About 10 s	Only windows built for it, such as Xorventa Instruct, on a clear miss

The result is a clear division of labour. Live chat gets its accuracy from choices that cost milliseconds: the embedding model, chunking, the two-pass read and the entry cap. Seconds-scale components run where nobody is waiting. And the knowledge layer, which accounts for the largest accuracy difference we measured, adds nothing to a turn at all, because its work happens when information arrives, not when a visitor asks.

The same retrieval serves text and voice, so a voice call answers from the same facts as the chat window. By default a window answers only from the business's approved knowledge.

6. Guidance for teams building similar systems

If you are deciding	Our recommendation	What would change it
How to store changing knowledge	Store dated, single-detail facts with explicit source precedence. Separate what is stored from what is inferred.	A static corpus with one authoritative source may not need it.
Which small embedding model	Run a paired comparison on your own content, per corpus, with questions from real passages.	Small models were close overall, so your domain decides.
Whether to re-rank	Evaluate it before changing embeddings, and shrink the pool and length before rejecting it on latency.	A GPU changes the trade-off and may allow it in live chat.
Whether to add keyword search	Measure it on your corpus rather than assuming it helps.	Part numbers, exact codes and large catalogues.
How to change embedding models	Map every threshold by percentile, pin the revision, re-embed atomically, and tune as a separate step.	Thresholds you already know are wrong should be re-tuned.
How to evaluate	Repeat runs, keep full outputs, and read failures. Non-deterministic pipelines vary on identical input.	An independent holdout set is better still.

7. Conclusion

For an assistant that speaks for a business, accuracy starts before retrieval. Storing knowledge as dated, sourced facts with explicit precedence removes a whole class of errors that better retrieval cannot reach, and it makes every change explainable to the person who owns the business. On top of that foundation, retrieval is engineered component by component, with each choice justified by paired measurement and placed according to what it costs at conversation speed.

These results come from Xorventa's own and demonstration knowledge bases, and the fact-record comparison from one detailed scenario. They should be read as measured evidence for this class of system: small knowledge bases, changing sources and CPU inference. The methods in this paper transfer directly, and we encourage teams to run them on their own content.

Appendix A: the test scenario

To reproduce the comparison, play these events in order through your system, then ask the questions. Today is November 2, 2026, except where noted. Answer with one model at temperature 0, using only the knowledge given.

#	Event	Content
1	Website	Sunny Bakery, fresh bread every day. Open Monday to Friday, 7 AM to 3 PM. Sourdough loaf is $8. Delivery within 5 miles for $5. Gift cards available in store.
2	Owner answer	Do you make wedding cakes? Yes, wedding cakes start at $250 and we need 4 weeks notice.
3	Owner answer	What are your holiday hours? We're open until 6pm Monday to Friday until Dec 24.
4	Document	Sourdough is now $9 a loaf. Our delivery area has grown to 10 miles.
5	Owner answer	Is delivery free? Delivery is free on orders over $50, otherwise $5.
6	Website	Re-crawl, unchanged.
7	Website	Changed: "We no longer sell gift cards."
8	Owner answer	How much is the sourdough? Sourdough is $8.50.
9	Document	Holiday flyer: Sourdough loaves $10.

Question	Correct answer	Wrong if it says
How much is a sourdough loaf?	$8.50	$8, $9 or $10
Do you make wedding cakes?	Yes, from $250	
How much notice do you need for a wedding cake?	4 weeks	
How late are you open on a weekday this month?	6 PM	3 PM as the current hours
Are you open on Saturdays?	No, or that it isn't known	Yes
Do you sell gift cards?	No	Gift cards are available
How far do you deliver?	10 miles	5 miles
Is delivery free?	Free over $50	
(Asked December 27) How late are you open on a weekday?	3 PM	6 PM

Appendix B: the two calculations

Threshold mapping, where `old` and `new` are each model's top similarity for the same pooled answerable and unanswerable questions:

```
new_threshold = numpy.quantile(new, (old < old_threshold).mean())
```

Paired comparison, where `b` and `c` are the questions only one of the two setups answered:

```
p = min(1, 2 * sum(comb(b + c, i) for i in range(min(b, c) + 1)) / 2 ** (b + c))
```

References

The models, index and statistical methods this work builds on. None of them is ours; the measurements on our own content are.

- Snowflake, snowflake-arctic-embed-s
- sentence-transformers, all-MiniLM-L6-v2
- BAAI, bge-small-en-v1.5
- intfloat, e5-small-v2
- cross-encoder, ms-marco-MiniLM-L6-v2
- BAAI, bge-reranker-base
- pgvector, vector similarity search for Postgres
- Malkov and Yashunin, Efficient and robust approximate nearest neighbor search using HNSW graphs
- Cormack, Clarke and Buettcher, Reciprocal rank fusion outperforms Condorcet and individual rank learning methods (SIGIR 2009)
- McNemar, Note on the sampling error of the difference between correlated proportions or percentages (Psychometrika, 1947)
- Wilson, Probable inference, the law of succession, and statistical inference (JASA, 1927)

THE PRACTICAL CONCLUSION

Decide what is true before you retrieve it, and measure every component at the speed your users feel.

Dated facts instead of rewritten knowledge

Explicit precedence, with the owner's answer first

Paired measurement, per corpus, repeated

Conversation speed as a design constraint

Explore Xorventa at xorventa.ai