



Retrieval-augmented generation

Connecting AI to governed internal knowledge without pretending the model knows everything.

WORKS WITH THE ASSISTANTS TEAMS ALREADY USE



EDITION 03 / EMBER & VIOLET

EXECUTIVE WORKING EDITION - 08 OCT 2026

03

AT A GLANCE

Connecting AI to governed internal knowledge without pretending the model knows everything.

Retrieval-augmented generation (RAG) lets an assistant answer from your own documents instead of from memory alone. The assistant finds relevant passages at the moment of the question and writes an answer based on them, ideally with citations. It can make policies, procedures and past work far easier to use, but only if the underlying content is trustworthy and the system is tested. This playbook covers when RAG is worth building, how to scope sources, how to evaluate retrieval and answers separately, and how to keep it accurate over time.

WHO IT IS FOR

Knowledge, operations, service and technology leaders deciding whether to put an assistant on top of company documents, and the people who will own the content and the system.

TIME TO USE

60 minutes to read; one workshop to apply

What you will have at the end

- A clear yes or no on whether RAG fits your use case
- A source inventory with named owners and freshness rules
- A test set with separate scores for retrieval and answers
- Access controls that match your existing document permissions
- A 90-day plan from a narrow pilot to a decision to scale

Contents

01	What RAG means for the business	03
02	Where RAG fits and where it does not	04
03	How RAG works, in plain language	06
04	Operating model, roles and content ownership	08
05	Risks, controls and human review	09
06	Measuring quality and value	11
07	Implementation guide: a narrow pilot first	13
08	Common pitfalls, an illustrative scenario and vendor questions	14
09	A first quarter that produces evidence	16
10	Terms used in this publication	17

EXECUTIVE BRIEF

What RAG means for the business

01



A model answers better when it is shown the right passage at the moment of the question, and says so when it cannot find one.

Language models are trained on public text up to a cutoff. They do not know your current policies, price lists, contract terms, engineering standards or the decision your team made last week. If asked, they may still produce a plausible answer, because fluency is what they are built for. RAG addresses this by separating knowledge from the model. The knowledge stays in your documents. At question time, the system searches those documents, selects the most relevant passages and gives them to the model with an instruction to answer from them and to cite them.

The practical benefits are specific. Employees get answers in seconds instead of searching shared drives. Answers point to a source the reader can open and verify. Content updates flow through when documents change, without retraining anything. Access can follow existing permissions, so people see answers only from documents they could already read. The costs are equally specific: someone must curate the sources, the system must be tested continuously, and the quality of the answers can never exceed the quality of the documents.

- RAG is a search problem first and a writing problem second: if retrieval fails, the best model cannot recover
- Most disappointing RAG projects fail on content quality, not on model choice
- A good system admits when the documents do not contain an answer
- Value comes from a defined set of questions and a defined set of sources, not from indexing everything

PLAIN TEST

If your best colleague would answer a question by opening the same two or three documents each time, RAG probably fits. If the answer lives in people's heads or in no document at all, it will not.

WORKING QUESTION

Which ten questions do people ask repeatedly whose answers already exist in a document somewhere?

WHERE IT FITS

Where RAG fits and where it does not

02

Choose RAG when answers must be grounded in your content. Choose something simpler when people only need to find a document, or something else when the data is structured.

RAG sits between three alternatives. Enterprise search returns documents and leaves reading to the user. A general assistant with no connection drafts from its training only. Fine-tuning changes the model's behavior or style using examples, but it is a poor way to teach changing facts. For questions on text content that changes and must be traceable, RAG is usually the right pattern. For questions on numbers in tables, such as sales by region, the better pattern is a query against the database, which may be combined with RAG for the explanatory text.

OPTION	BEST WHEN	WATCH FOR
Enterprise search	People just need to find the right document	No synthesis; reading effort stays with user
RAG assistant	Answers must be grounded in your documents with citations	Stale, duplicated or contradictory sources
Database query with AI interface	Questions are about structured figures	Wrong joins; definitions of metrics
Fine-tuning	Consistent style or format is the goal	Poor for facts that change
General assistant, no connection	Drafting from supplied text	No knowledge of your internal facts

USE CASE	FIT	REASON
HR and IT policy questions	Strong	Stable documents, clear owner, many repeat questions
Sales proposal support from past proposals	Strong with review	Reuse value high; confidentiality and currency need controls
Service agents seeking troubleshooting steps	Strong	Time-critical lookup; measurable resolution impact
Legal interpretation of contracts	Moderate	Useful for locating clauses; professional judgment required
Questions with answers not written anywhere	Weak	Nothing to retrieve
Real-time operational data	Weak	Needs live system integration, not document search

COMMON MISTAKE

Indexing every shared drive on day one. Poorly maintained folders produce confident answers from outdated drafts. Start with one question set and one owner.

WORKING QUESTION

Is your use case a document question, a data question, or both, and have you separated them?

HOW IT WORKS

How RAG works, in plain language

03

Four stages: prepare content, find passages, write the answer, check the result. Each stage has its own failure modes and its own controls.

01 Prepare

Collect documents, extract clean text, attach metadata such as owner, date, version and access rights, and split the text into passages sometimes called chunks.

02 Index

Convert passages into numerical representations called embeddings, and store them with the text in a searchable index. Many systems also keep a keyword index.

03 Retrieve

At question time, find the most relevant passages using meaning-based search, keyword search or both, filtered by what the user may see.

04 Generate

Give the model the question and the retrieved passages, instruct it to answer only from them, cite them and say when the answer is not there.

05 Check

Apply validation, such as confirming that quoted text appears in the source, and log the question, passages and answer for review.

Retrieval quality depends on choices that look technical but have business consequences. Chunk size matters: passages that are too small lose context, and passages that are too large dilute the signal. Metadata matters: if the system cannot tell the 2022 policy from the 2025 policy, it will blend them. Combining keyword and meaning-based search usually helps with codes, part numbers and exact terms that embeddings handle poorly. A reranking step, which reorders candidates by relevance, often improves results. None of these needs to be solved in theory, but all need to be tested on your documents.

COMPONENT	WHAT IT DOES	TYPICAL FAILURE
Document parsing	Turns PDFs, slides and tables into clean text	Tables flattened; headers lost; scanned pages missed
Chunking	Splits text into passages	Answer split across two passages
Embedding and index	Makes passages searchable by meaning	Poor match on codes and names
Retrieval and reranking	Selects best passages	Relevant passage not in the top results
Generation	Writes grounded answer	Adds claims beyond the passages
Access filter	Hides passages the user may not see	Permissions out of date or ignored

KEY DISTINCTION

Wrong answers come from two places: the right passage was not retrieved, or the right passage was retrieved and misused. Measure these separately, or you will fix the wrong thing.

WORKING QUESTION

For a question you care about, can you point to the exact passage that contains the answer, and would a search find it?

OPERATING MODEL

Operating model, roles and content ownership

04

RAG is a content product. It needs source owners, a system owner and a routine for fixing what is wrong.

The weakest point in most deployments is not software but stewardship. Documents have authors who leave, versions that are never retired, and drafts saved next to finals. Before indexing anything, create a source register: for each source, record the owner, the purpose, the review date, the access rule and the process for retiring outdated versions. If no one will own a source, do not index it.

ROLE	OWNS	TYPICAL TASKS
Business owner	Use case and outcomes	Defines questions, accepts quality, reports value
Source owner (per source)	Accuracy and currency of documents	Reviews, retires and updates content on schedule
Knowledge manager	Structure, metadata and taxonomy	Defines naming, versions, tagging standards
System owner	Pipeline, index and releases	Monitors, tunes retrieval, manages incidents
Security and privacy lead	Access and data handling	Maps permissions, reviews vendors, approves sources
Reviewers	Answer quality in sampled review	Score answers, log defects, suggest fixes

Source register fields

- Source name, location and format
- Named owner and a backup
- Content type: policy, procedure, contract, proposal, guidance
- Last reviewed date and review interval
- Access rule: who may read, and how that is enforced in the index

- Sensitivity class and any restrictions on use in AI systems
- Retirement rule: how superseded versions are removed from the index

Close the feedback loop with users. Add a simple way to flag an answer as wrong, outdated or unhelpful, and route flags to the right source owner. Many apparent AI errors are documentation errors. When a source owner fixes a policy because the assistant exposed a contradiction, the organization gains twice: a better assistant and better documentation.

WORKING QUESTION

If you indexed your first source today, who would be accountable for the accuracy of what it says in six months?

RISKS AND CONTROLS

Risks, controls and human review

05

The new risks are leakage through retrieval, confident answers from bad sources and over-trust. Controls start with access and end with review.

Frameworks such as the NIST AI Risk Management Framework and ISO/IEC 42001 provide structure for governing AI systems, and data protection law, sector rules and the EU AI Act may apply depending on the use and the data. This playbook does not give legal advice. Confirm current requirements with qualified advisers and turn them into controls on sources, access, logging and review.

RISK	EXAMPLE	CONTROL
Access leakage	Assistant surfaces a salary document to someone without rights	Enforce document permissions at retrieval; test with restricted users
Stale answers	Superseded policy cited as current	Version metadata; retirement rule; show the date in citations

RISK	EXAMPLE	CONTROL
Unsupported claims	Model adds details not in the passages	Require citations; verify quotes; instruct to decline when unsure
Contradictory sources	Two documents give different rules	Source register; resolve conflicts at source; show both when unresolved
Prompt injection in documents	Text in a file tries to instruct the assistant	Treat content as data; restrict actions; review ingested sources
Over-trust	Users stop checking citations	Training; visible source links; periodic audits

Controls to confirm before launch

- Every source has an owner, a review date and a sensitivity class
- Permissions in the index match those in the source system and are tested
- Answers show citations that open the original document at the right place
- The assistant is instructed to say it does not know when evidence is missing
- Questions, passages and answers are logged with appropriate retention
- A way to report and fix bad answers is live and staffed
- High-stakes use, such as legal or safety guidance, includes expert review

EXAMPLE (ILLUSTRATIVE)

A manufacturer indexes safety procedures. Answers always show the document title, version and date. Any answer touching lockout steps also displays: Confirm against the current controlled procedure before acting. The safety lead audits a sample each month.

WORKING QUESTION

Have you tried to make the assistant reveal something a restricted user should never see, and what happened?

MEASURING VALUE

Measuring quality and value

06

Test retrieval and answers separately, on real questions, and compare with how people find answers today.

Start with a baseline. Ask ten colleagues to answer twenty representative questions using current methods, and record time taken and correctness. Then build a test set: fifty to a hundred real questions with, for each, the document and passage that contain the answer and a short reference answer. Include awkward cases: questions with no answer in the documents, questions using internal jargon, questions where two documents conflict and questions about superseded policies.

METRIC	QUESTION IT ANSWERS	HOW TO MEASURE
Retrieval hit rate	Was the right passage in the top results?	Check test questions against known passages
Answer correctness	Is the answer right and complete?	Reviewer scores against reference answer
Groundedness	Is every claim supported by the cited passages?	Reviewer checks claims against citations
Correct refusal rate	Does it decline when no answer exists?	Include unanswerable questions in test set
Citation accuracy	Do links lead to the supporting text?	Sample check of links and quotes
Time to answer	Is it faster than current methods?	Timed comparison with baseline
Deflection or resolution	Do fewer questions reach experts or service queues?	Ticket or query counts before and after

How to run an evaluation

01 Write the questions

Draw from real queries, support tickets and expert interviews, not from invented examples.

02 Mark the evidence

Record where the answer lives. This lets you test retrieval independently.

03 Score retrieval first

If the right passage is not found, fix chunking, metadata or search before touching prompts.

04 Score answers

Use two reviewers on a sample and resolve disagreements to calibrate scoring.

05 Set a pass mark

Agree thresholds before the pilot, higher for riskier content, and re-test after every change.

ILLUSTRATIVE FIGURES ONLY

If staff spend 12 minutes finding a policy answer and the assistant delivers one in 2 minutes with a check, the saving is 10 minutes per question. Multiply by your measured volume, and subtract time spent correcting wrong answers.

Monitor continuously after launch. Track the share of answers flagged as unhelpful, the questions that return no answer, and the sources cited most often. Unanswered questions are a content roadmap: they show what people need and your documents lack.

WORKING QUESTION

What would your retrieval hit rate need to be before you let anyone rely on an answer without opening the source?

IMPLEMENTATION GUIDE

Implementation guide: a narrow pilot first

07



Pick one audience, one question set and a small number of owned sources. Prove accuracy, then expand.

01 Define the use case

Name the audience, the twenty to fifty questions that matter and the decision each answer supports.

**02 Inventory and clean sources**

Select three to ten documents or collections, assign owners, retire duplicates and fix obvious contradictions.

03 Design access

Decide how permissions carry into the index and who may query what.

04 Build or configure

Set up parsing, chunking, metadata, hybrid search and citation display. Keep the first version simple.

**05 Evaluate and tune**

Run the test set, inspect failures, adjust, and re-run. Keep a log of changes and results.

06 Pilot with real users

Release to a small group with a feedback button, weekly review and clear limits on use.

07 Decide

Compare with baseline and the pass mark. Scale, change or stop.

Source readiness

- Owner named and willing to maintain the content
- Current version clearly identified; old versions removed or tagged
- Text can be extracted cleanly; scanned files have been processed with quality checks
- Tables and images that carry meaning have been handled deliberately



- Sensitive content classified and permissions understood
- Document dates and titles are present and accurate



Plan for content work, which is usually the larger share of effort. Typical tasks include rewriting ambiguous policies, splitting long mixed documents, adding effective dates, and creating short answer pages for common questions. This work improves human search too, so it is not wasted if the assistant is later changed. Also plan for people: show users how to phrase questions, how to read citations and when to escalate to an expert.

WORKING QUESTION

Which three sources would you index first, and which would you deliberately leave out?



PITFALLS AND VENDORS

Common pitfalls, an illustrative scenario and vendor questions

08



Most RAG disappointments are content, scoping and measurement problems. Vendors can help, but they cannot supply governance you have not built.

- Indexing everything, then discovering the assistant quotes drafts and expired policies
- Judging success on demonstration questions rather than a representative test set
- Treating the model as the cause of every error when retrieval is the real problem
- Ignoring permissions until the first security review
- Launching without a feedback route or a content owner
- Expecting RAG to answer questions that no document answers



ILLUSTRATIVE SCENARIO (HYPOTHETICAL)

A services firm launches an assistant over its whole intranet. Users report contradictory answers about travel rules. Investigation shows three versions of the policy in the index, one from several years ago. The firm narrows scope to five owned sources, adds effective dates, writes a test set from real questions and re-launches. Complaints drop and the content owners now review the unanswered-question report monthly.

VENDOR QUESTION	WHAT TO LOOK FOR
How are document permissions enforced at query time?	Index-level filtering tied to your identity system, with a way to test it
Can we see retrieved passages and scores for each answer?	Full traces to diagnose retrieval versus generation errors
What control do we have over chunking, metadata and search?	Configurable pipeline, hybrid search, reranking options
Where is our content stored and processed, and is it used for training?	Documented regions, retention settings and written data terms
How do you handle updates and deletions?	Prompt re-indexing; deleted content removed from the index
Can we export our index and configuration?	Portability to avoid lock-in

Build or buy decision

- A packaged product covers your sources and permission model
- You need custom pipelines for unusual formats or strict controls
- Your team can run evaluation and content operations after launch
- Cost is modeled at pilot and production volume, including indexing and queries
- Exit plan exists if quality or terms change

 WORKING QUESTION

If the vendor's answers were wrong tomorrow, could you tell whether retrieval or generation was at fault?

30 / 60 / 90

 A first quarter that produces evidence

09

Prove accuracy on a narrow, owned set of sources, then widen with evidence.

 First 30 days: ground it

- Define the audience and collect fifty real questions
- Build the source register with owners, dates and access rules
- Record baseline time and correctness for current methods
- Clean or retire conflicting and outdated documents in scope

 Days 31 to 60: test it

- Configure the pipeline with metadata, hybrid search and citations
- Run the test set; score retrieval and answers separately
- Test permissions with restricted users
- Fix top failure causes and re-run the evaluation

 Days 61 to 90: decide

- Pilot with real users, feedback button and weekly review
- Compare results with the baseline and pass mark
- Publish an unanswered-question report to source owners
- Decide to scale, change or stop; plan the next sources

GLOSSARY

Terms used in this publication

10

TERM	MEANING
Retrieval-augmented generation	Answering from passages found in your documents at question time.
Embedding	A numerical representation of text used to find passages with similar meaning.
Chunk	A passage of a document stored and retrieved as a unit.
Hybrid search	Combining keyword search with meaning-based search.
Reranking	Reordering retrieved passages by relevance before they reach the model.
Grounding	Basing an answer on supplied evidence rather than model memory.
Citation	A link from an answer to the passage that supports it.
Metadata	Descriptive fields such as owner, date, version and access rights.
Retrieval hit rate	The share of questions where the right passage appears in the results.
Hallucination	A fluent statement not supported by any source.
Source register	A list of indexed sources with owners, dates and access rules.
Access filtering	Limiting retrieval to documents the user is allowed to read.

A note on use

This publication is a public orientation resource, not legal, technical, financial or professional advice. Adapt it to your people, information, workflows and obligations, and check current tool terms, regulations and risks before acting.



TAKE THE NEXT STEP

Take the free Halden assessment at halden.company/assessments, or talk to us at halden.company/contact about turning this into a plan for your organization.

