

RAG for Marketing

Governed knowledge, retrieval and answer quality



HOW TO USE THIS RESOURCE

RAG for Marketing

Marketing operations, knowledge owners and technical partners

Your practical outputs

- Source authority and corpus register
- Retrieval and answer test cards
- Access and prompt-injection test plan
- Maintenance economics and release packet

Read, choose, apply

Start with the overview and decision rules. Choose a concrete constraint in your business. Work through the relevant templates using your own evidence, record open assumptions and agree a review with the accountable people. The complete online edition provides linked sources and contextual navigation.

Editorial context

Explicitly fictional cases and example figures are working aids, not client results or universal benchmarks. Frameworks structure decisions; they are not scientifically validated scores. Platform capabilities may change. Sources and review context are disclosed.

Read online: [RAG for Marketing](#)

CONTENTS

1. Define the knowledge service before the index	4
2. Choose the first question population	4
3. Create a source authority hierarchy	5
4. Enforce access before evidence is assembled	6
5. Treat retrieved instructions as untrusted content	6
6. Evaluate answers and retrieval as separate layers	7
7. Prepare documents without destroying meaning	8
8. Choose passage boundaries through evidence tests	9
9. Diagnose the first broken link	9
10. Price knowledge maintenance, not only generation	10
11. Test retrieval separately from generation	11
12. Build a claim-grounded answer contract	11
13. Operate freshness, deletion and ownership	12
14. Release a bounded knowledge service	13
Turn this resource into a team brief	15
Sources & further documentation	16

1. Define the knowledge service before the index

A marketing RAG service retrieves selected material and uses it as context for generating an answer. The [original paper by Lewis and colleagues](#) introduced a research approach combining parametric and non-parametric memory. A contemporary marketing implementation still needs its own source governance, permission design, evaluation, and operating decisions. Research results from another dataset do not predict the quality of your knowledge base.

Start with a question population, not a folder upload. Which people ask what questions, for which decisions, using which approved evidence? An internal product-marketing assistant, a public website assistant, and a sales enablement assistant have different permissions and consequences. Do not use one index simply because the documents overlap.

The original framework here is CLEAR: Curate authoritative evidence, Limit access and scope, Examine retrieval, Assess answers, Renew knowledge. Each step has a separate owner and acceptance artifact. Retrieval relevance is not truth. A document may match the query while being wrong, obsolete, restricted, or outside product scope.

Service definition worksheet

Write the intended user, question classes, permitted source classes, output destination, expected abstentions, and explicit exclusions. State whether the system only answers or can also choose tools and perform actions. Keep action authority out of the initial knowledge service unless a separate business case and control design require it. Describe the manual alternative when evidence is absent. The service succeeds when it helps a user make a bounded decision with inspectable evidence, not when it always produces a confident paragraph.

2. Choose the first question population

Use the FICTIONAL B2B SaaS company NorthstarDesk as a worked case. Its marketing team needs internal answers about approved product capabilities when preparing campaign briefs. It does not need the assistant to answer contractual questions, disclose roadmap commitments, or publish customer-facing claims. Every example and numerical assumption in this playbook is fictional and is not a client result or benchmark.

Collect real question shapes from approved internal work without importing information the pilot is not permitted to process. Cluster them into capability, use-case, positioning, integration, limitation, and missing-evidence questions. Preserve wording variation because users rarely phrase a question like a document heading. Include German and English questions if both languages belong in the intended service.

Question family	Needed evidence	Expected disposition
Current capability	Approved product reference	Answer within scope

Question family	Needed evidence	Expected disposition
Future roadmap	Planning document only	Decline public commitment
Customer outcome	No approved study	Explain evidence gap
Product limitation	Approved limitation note	Answer with qualification

Question-card template

Record question, user role, intended decision, product, language, date sensitivity, acceptable sources, and expected answer or abstention. Ask whether a wrong answer would be caught before use. If the only detection point is public publication, the service needs stronger review or a narrower boundary. Keep out-of-scope questions in the test set to check correct refusal, but exclude them from claims about supported coverage. The population definition becomes the denominator for later reporting: “good on our selected internal capability questions” is a useful statement; “accurate for marketing” is not.

3. Create a source authority hierarchy

A knowledge base needs a rule for conflicting evidence before retrieval begins. Define which source is authoritative for product capabilities, positioning, pricing, security claims, customer outcomes, and roadmap material. The newest file is not automatically authoritative. A newer presentation may be an unapproved draft, while an older product reference remains the approved statement.

Build a source register with document ID, owner, approval status, product scope, audience permission, effective date, expiry or review trigger, language, and superseded-by link. Separate approved evidence from research material and rejected content. Do not rely on filenames such as “final-final” to communicate authority.

Conflict-resolution workflow

When two sources disagree, preserve both identifiers and the conflicting passages. Determine whether the difference concerns version, scope, terminology, or an unresolved business decision. Route to the source owner. Until resolved, the answer should qualify the conflict or abstain according to the service contract. The generator should not synthesize a compromise that neither source supports.

For NorthstarDesk, an approved capability reference governs internal campaign-brief answers. A sales presentation may explain common language but cannot expand the capability claim. A roadmap slide remains outside the public-claim boundary. Record these rules in metadata and in the reviewer rubric. Verify that retrieval applies them rather than assuming the model will infer organizational authority from document style. The [NIST Generative AI Profile](#) is a broader

voluntary risk reference; the concrete hierarchy here is a local operating decision that needs named ownership and maintenance.

4. Enforce access before evidence is assembled

A knowledge service must answer only from material the user is permitted to access for the intended purpose. Index membership does not imply universal access. Separate public, internal, restricted, and customer-specific collections where the permission model requires it. Keep identity and source authorization in the retrieval path rather than asking the generator to remember who should see what.

Test with representative roles, including a restricted user and a user whose access has been removed. Verify retrieval results, cached answers, logs, exports, and links. A source passage may be protected while its summary remains visible through an earlier conversation or shared output. Map those secondary paths before declaring the access boundary complete.

Access-test matrix

Record user role, source class, expected access, actual returned evidence, generated answer, output destination, and result. Test both allowed and denied cases. A system that denies everything is not useful; a system that permits everything is not governed. Include changes in group membership and source sensitivity.

Keep administrative permissions visible: who can upload documents, alter metadata, change filters, export logs, or grant tools? These capabilities can change the service boundary without changing a prompt. This is operating guidance, not legal advice or a claim of compliance. Resolve privacy, contractual and sector-specific requirements with qualified owners. If the access model cannot enforce a required restriction, exclude the material or use a narrower service. Do not let a promising answer demonstration become a reason to accept an unverified permission design.

5. Treat retrieved instructions as untrusted content

A retrieved document may contain instructions that try to change the assistant's behaviour, expose information, or trigger a tool. [OWASP describes prompt injection](#) as a risk class that retrieval does not eliminate. A document can be factually useful and hostile at the same time. Source admission and content trust are therefore related but distinct questions.

Keep the knowledge assistant read-only at first. Restrict tools and destinations independently of the retrieved text. Do not place credentials in prompts or allow a source passage to authorize external actions. If dynamic agents are introduced, assess tool choice, intermediate state, and action approval separately from answer quality. The [Anthropic architecture discussion](#) provides a

useful conceptual distinction between predetermined workflows and systems choosing their own steps.

Safe adversarial exercise

Create harmless test documents containing instructions to ignore review, reveal a restricted placeholder, or send a message to a dummy destination. Use no live secrets and no actual external recipients. Observe whether the system treats those instructions as data and whether technical restrictions block actions regardless of model output. Preserve the trace needed to diagnose the failure.

Do not claim complete prevention because a selected attack set passes. Attack forms, documents, models and integrations change. Record what was tested, what was not, and which controls limit consequence if the model follows a bad instruction. Defence should reduce exposure and improve detection while preserving an explicit stop path. A stronger prompt may help, but cannot substitute for missing access and action boundaries.

6. Evaluate answers and retrieval as separate layers

Build a layered evaluation set. The source layer checks whether the approved evidence exists and is current. The retrieval layer checks whether it is returned with enough context. The answer layer checks support, scope, useful uncertainty, and task completion. The permission layer checks whether the user and destination are allowed. A single end-to-end score cannot explain which layer needs repair.

Use [OpenAI's evaluation documentation](#) as a technical reference where appropriate. The rubric here should be specific to your knowledge service. Include exact identifiers, paraphrases, multilingual questions, conflicts, absent evidence, restricted sources, and hostile content. Freeze unseen cases before tuning the preferred configuration.

Layer	Passing evidence	Failure meaning
Source	Approved current passage exists	Corpus gap or authority problem
Retrieval	Required context returned	Search or filtering issue
Answer	Claims supported and scoped	Generation or instruction issue
Permission	Correct user and destination	Boundary failure

Interpretation rule

Report numerator, denominator, case selection, configuration and source snapshot. A sample score is not population performance. Zero critical failures in a small test set does not prove absence of critical risk. Inspect weak segments and keep severe cases separate from averages. Automated judges can assist triage, but require calibration and error inspection. The release owner must decide from quality, consequence, cost, and reversibility together. An answer that is stylistically excellent but discloses restricted information fails regardless of every other score.

7. Prepare documents without destroying meaning

Ingestion quality determines what retrieval can see. Extract text, headings, tables, footnotes, product labels, and effective dates in a way that preserves their relationship. A table cell detached from its column header can become misleading evidence. A limitation moved away from a capability statement can turn a qualified claim into an apparent promise.

Inspect representative files manually after parsing. Include PDFs, slide exports, web pages, and structured documents if those formats belong in scope. Check whether repeated navigation, headers, legal footers, hidden text, or OCR errors dominate the extracted content. Do not assume successful upload means useful ingestion.

Ingestion acceptance checklist

- Document identity and source link survive extraction.
- Headings and section relationships remain inspectable.
- Tables retain units, column labels and qualifications.
- Product and version context travels with relevant passages.
- Restricted or rejected material is excluded before indexing.
- Deletion and supersession propagate to the serving collection.

Store parsing configuration and source version together. When extraction changes, test retrieval and answer behaviour again, not just file counts. For hosted tools, inspect what evidence is actually returned and what metadata can be filtered. [OpenAI file-search documentation](#) describes a retrieval facility, but teams still need to validate their own source preparation. A smaller, clean corpus can be more useful than a large collection whose authority and structure are unclear. Measure coverage gaps explicitly rather than hiding them under the number of uploaded pages.

8. Choose passage boundaries through evidence tests

Chunking is an evidence-design choice, not simply a token-count setting. A passage must contain enough context to support a claim while remaining specific enough to retrieve. Very small chunks may lose qualifications; very large chunks may mix products, dates, or permissions. There is no universal best size for all marketing corpora.

Start with document structure: headings, paragraphs, tables, and coherent claim-plus-limitation units. Preserve parent document and section identifiers. Attach product, audience, language, and version metadata. If overlap is used, check that repeated passages do not dominate retrieval or exaggerate apparent evidence diversity. Five near-duplicate chunks from one document are not five independent sources.

Chunk experiment card

Record the question set, source snapshot, parser version, boundary method, overlap, retrieval configuration, and evaluation results. Compare retrieval of expected evidence, irrelevant content, context completeness, and resulting answer support. Keep the generation configuration fixed while testing chunk changes where practical. Otherwise you cannot tell which change helped.

Use difficult examples: a capability whose limitation sits in the next paragraph, a pricing table with a footnote, and a product name shared across versions. Inspect the actual passages. If the correct fact is retrieved without its limiting condition, the retrieval result is not sufficiently useful even though it contains a matching phrase. Choose a configuration that serves the defined questions and review process. Revisit it when document structure or query population changes, and retain the old version for regression comparison.

9. Diagnose the first broken link

When the answer is wrong, inspect the chain in order. Was the question within scope? Did approved evidence exist? Was it ingested correctly? Did retrieval return it with context? Did generation use it faithfully? Did review catch the issue? This sequence avoids treating every defect as a prompting problem.

Preserve a restricted diagnostic packet with query, user role, source snapshot, configuration, returned passages, answer, expected disposition, and reviewer rationale. Mask unnecessary sensitive content. Compare the failed case with a nearby passing case. The contrast can reveal a product-code mismatch, language issue, metadata filter, or missing qualification.

Symptom	First check	Repair owner
Plausible invented feature	Evidence existed and was returned?	Source/retrieval/generation owner

Symptom	First check	Repair owner
Old product answer	Supersession and filters applied?	Source steward and maintainer
Citation does not support claim	Passage entails wording?	Answer designer and reviewer
Restricted information appears	Authorization before retrieval?	Security and technical owner

Closure rule

A repair needs a causal explanation or an explicit uncertainty statement, a reproduced test, a targeted fix, and regression evidence. Do not close a defect because one regeneration looks good. If the failure concerns access or forbidden action, stop the affected route and involve the incident owner. If the corpus is incomplete, update it through source admission rather than importing a convenient unapproved document. Keep unresolved defects visible to the release owner and narrow permissions when the cause remains unclear.

10. Price knowledge maintenance, not only generation

The economics of RAG include source stewardship, ingestion checks, index operations, retrieval usage, generation, review, evaluation, access administration, and incident handling. A cheap generated answer may require expensive knowledge maintenance. Compare the service with the real alternative, such as a maintained reference page or a human source expert, rather than assuming the alternative is unlimited manual search.

FICTIONAL NorthstarDesk planning case: assume 300 accepted answers per month. Manual lookup and explanation take 12 minutes each; assisted lookup and review take 7 minutes. At an assumed €60 hourly rate, the capacity difference is €1,500. Assume €250 tooling and index cost, €600 source maintenance, and €300 evaluation and administration for the same month. The remaining recurring capacity value is €350 before an assumed €4,200 setup investment. These are illustrative assumptions, not prices, benchmarks or client outcomes.

Sensitivity worksheet

Vary accepted volume, review minutes, source-update frequency, and maintenance effort independently. If review rises to ten minutes per answer, the capacity difference becomes €600 and the recurring net value becomes negative under these assumptions. If only a narrow subset is answerable, use that subset's volume rather than all incoming questions.

Measure costs per accepted, evidence-supported answer and record rejected attempts. Track whether saved time is actually used. Do not attribute pipeline or revenue changes to the knowledge assistant without a separate causal design. A clean, governed FAQ may be the better

investment when the question population is small and stable. RAG is one delivery pattern, not a requirement for knowledge maturity.

11. Test retrieval separately from generation

A generated answer cannot reveal whether retrieval worked unless the evidence is inspected. Build a retrieval-only test that asks whether the expected approved passage was available in the returned set. Separately inspect irrelevant passages, conflicting versions, permission violations, and missing context. Then test whether generation uses that evidence correctly.

Keyword and semantic approaches can behave differently on product codes, acronyms, paraphrases, and multilingual questions. Hosted file search may combine retrieval methods; inspect the actual documented behaviour and your results rather than assuming a label determines quality. Choose query and filter strategies from observed failure patterns.

Observation	Retrieval diagnosis	Next experiment
Correct source absent	Coverage or indexing problem	Check admission and ingestion
Correct source indexed but not returned	Ranking/query issue	Compare queries and filters
Old version returned	Authority metadata issue	Test supersession filtering
Correct passage returned, wrong answer	Generation or task issue	Inspect answer constraints

Retrieval scorecard

For each query record expected source, returned source IDs, rank or order when available, evidence completeness, permission status, and reviewer explanation. Aggregate by question family and language. A single retrieval percentage can conceal systematic failure on product codes or German phrasing. The sample describes selected queries, not all future requests. Preserve the returned passages for reproducibility under appropriate access controls. If retrieval cannot find a required source reliably, do not ask the generator to compensate from memory. Improve the corpus or retrieval, narrow coverage, or return an explicit evidence gap.

12. Build a claim-grounded answer contract

Define the answer format around the user's decision. A useful internal answer contains the direct response, scope and qualifications, claim-level source references, and unresolved questions. It should distinguish what the source states from the assistant's interpretation or suggested next step. Do not require a full answer when the evidence is insufficient.

For NorthstarDesk, “Can we claim this integration is available?” requires the current approved capability source and product scope. A relevant roadmap passage is not enough. The correct answer may explain that availability is not established and route the question to the product owner. This is useful assistance, even though it does not satisfy the user's hoped-for claim.

Answer-contract worksheet

Specify required sections, permissible uncertainty language, expected citation detail, refusal conditions, escalation owner, and forbidden inferences. Include examples of an answerable question, a partially answerable question, a contradiction, and an out-of-scope request. Ask the reviewer to identify which sentence supports the requested decision.

Inspect whether the answer adds unsupported causal or quantitative language while paraphrasing. “Supports onboarding workflows” does not entail “reduces onboarding time by a specific amount.” A citation beside that sentence does not repair the inference. Test citation entailment, source authority, scope, and freshness separately. A response can cite a real source and still fail every material requirement. Keep final publication approval outside the answer contract unless that action has its own permission and review design. The answer service prepares a decision; it does not acquire authority from sounding confident.

13. Operate freshness, deletion and ownership

A knowledge base becomes stale through ordinary business change. Product releases, pricing changes, retired claims, reorganized ownership, and revoked permissions can all invalidate useful evidence. Use event-driven updates alongside scheduled review. A source review date is a reminder to inspect, not proof that the content is still correct.

Define an update path: source owner approves a change, ingestion produces a new version, targeted tests inspect affected questions, the serving collection switches, and the previous version remains available for authorized investigation. A deletion path should identify the source, index copies, cached or derived outputs where applicable, and any downstream tracing limits. Do not promise complete deletion from every system unless the actual architecture supports it.

Ownership handover card

Record source domain, authoritative collection, update triggers, overdue items, affected questions, access rules, technical maintainer, test owner, and incident contact. The receiving steward performs one update and one retirement in a safe environment. They then verify that a question no longer uses the retired evidence.

Keep a source-health view with overdue reviews, ownerless documents, conflicting versions, failed ingestions, and unresolved coverage gaps. Pair counts with consequence: one stale pricing source may matter more than many harmless archived notes. The [NIST AI RMF](#) is a broad reference for continuing risk management; the concrete freshness process must fit your own

service. Retire sources that no longer have an accountable owner or a permitted purpose. More documents do not automatically create more knowledge quality.

14. Release a bounded knowledge service

Release the service in stages: curated offline tests, internal shadow answers, supervised use by a defined group, and only then a considered expansion. Keep the source snapshot, retrieval configuration, generation configuration, permission model, evaluation set, and release decision linked. A model upgrade or new corpus can change behaviour without a visible change in the user interface.

Define STOP rules before launch. Stop the affected route for unauthorized disclosure, forbidden action, unresolved source authority on consequential claims, missing operational ownership, or inability to return to a manual answer path. Quality and economic misses may call for redesign rather than permanent retirement, but they do not justify silent expansion.

Release packet

Include question population, excluded questions, approved sources, access tests, retrieval and answer results by segment, critical failures, unresolved defects, cost range, review capacity, update process, rollback procedure, and named receiving owner. Ask the receiver to process an answerable question, an evidence-gap question, a conflicting-source question, and an access-denied question. The handover is complete only when they can explain the disposition and find the evidence.

Expand one dimension at a time: corpus, user group, language, audience exposure, or action authority. Public-facing answers require their own assessment even when the internal assistant works well. A retrieval service need not become an autonomous agent to be useful. If dynamic tool selection is introduced, evaluate the new intermediate decisions and consequences separately. The strongest outcome is a knowledge service whose scope, evidence and limits remain understandable after the original builder leaves.

Worked retrieval-versus-answer diagnosis

Consider a FICTIONAL NorthstarDesk question: “Can we position the Atlas connector as available for every account?” The approved product reference limits availability to a specified product edition. An older sales presentation uses broader wording. The retrieval trace returns both documents. This is not enough to call the answer grounded. The reviewer must inspect which source is authoritative, whether the limiting passage was returned, and how the answer used it.

First, the source steward confirms the current reference and marks the older presentation as superseded for capability claims. Second, the technical maintainer checks whether the serving index contains the correct metadata and whether the filter actually excludes superseded capability evidence. Third, the answer reviewer compares the generated claim with the current passage. If the output says “every account,” it fails even when the correct document appears in its citations. A citation proves location, not entailment.

Run three diagnostic variants. In the first, remove the current source from a safe test snapshot: the desired behaviour is an evidence-gap response. In the second, provide the correct passage with its limitation: the answer should retain that limitation. In the third, use a restricted test role: the system must respect access even when the document would answer the question. These variants isolate corpus coverage, answer faithfulness, and permission enforcement.

Save the case as a regression family rather than a single memorized question. Add paraphrases, product-edition terms and language variants without claiming they represent the entire population. Record investigation and source-repair effort in the maintenance ledger. Release the repaired route only after the relevant owner accepts the updated source rule, retrieval behaviour and answer disposition. This walkthrough is a fictional diagnostic design, not an observed failure rate or a claim that the same repair fits every RAG implementation.

Primary references and limits

Sources checked 24 October 2026. Technical documentation changes; verify implementation details before release. Frameworks, worksheets and fictional assumptions are original operating proposals, not vendor guarantees.

- [NIST AI Risk Management Framework](#)
- [NIST Generative AI Profile](#)
- [OpenAI: Working with evals](#)
- [OpenAI: Agents](#)
- [Anthropic: Building effective agents](#)
- [OWASP: Prompt injection](#)
- [Lewis et al.: Retrieval-Augmented Generation](#)
- [OpenAI: File search](#)

YOUR TRANSFER CANVAS

Turn this resource into a team brief

Use this page for a joint review. Agree one practical output, observable evidence and an accountable owner. Add the specific methods and templates from the relevant chapters.

Working field	Your agreement
Priority output	Source authority and corpus register
Your starting point	Finding / source / date: _____
Smallest useful test	_____ _____
Evidence & counter-evidence	_____ _____
Ownership & review	Owner / reviewer / date: _____
Stop or change when ...	_____ _____
Decision & next action	_____ _____

Keep unresolved assumptions explicitly open. A completed worksheet is not a substitute for evidence or accountable judgement.

Sources & further documentation

Primary sources checked 24 October 2026. Sources support the linked technical or factual statements; the working models and fictional cases are original editorial decision aids. Recheck changing platform details before implementation.

1. [NIST AI Risk Management Framework](#)
2. [NIST Generative AI Profile](#)
3. [OpenAI: Working with evals](#)
4. [OpenAI: Agents](#)
5. [Anthropic: Building effective agents](#)
6. [OWASP: Prompt injection](#)
7. [Lewis et al.: Retrieval-Augmented Generation](#)
8. [OpenAI: File search](#)

Next step

If you want help adapting the working plan to your product, evidence and team capacity, request a non-binding strategy conversation.

patrickwings.com