Clausey ENGINEERING

PRODUCTION ARCHITECTURE · 45 MINUTES

Building Enterprise-Grade Agentic Knowledge Bases

From messy documents to grounded, cited, measurable answers.

Larry Galan Co-Founder, Clausey

The Problem With “Chat Over Docs”

The default RAG demo is too simple for enterprise use.

- ✕ Parsing loses text, tables, and page numbers
- ✕ Chunking splits clauses in bad places
- ✕ Vector search misses exact terms, dates, and names
- ✕ The LLM infers answers the evidence doesn’t support
- ✕ Citations are decorative, not verifiable
- ✕ Failures are undebuggable; quality drifts silently

What Enterprise Buyers Actually Need

Knowledge bases must behave like reliable systems, not experiments.

TRUST

Correctness & traceability

Right answers, cited to the exact source, every time.

SECURITY

Isolation & auditability

Multi-tenant boundaries, audit logs, deletion and retention controls.

RELIABILITY

Deterministic failure

Retry safety, visible failure states, no silent data loss.

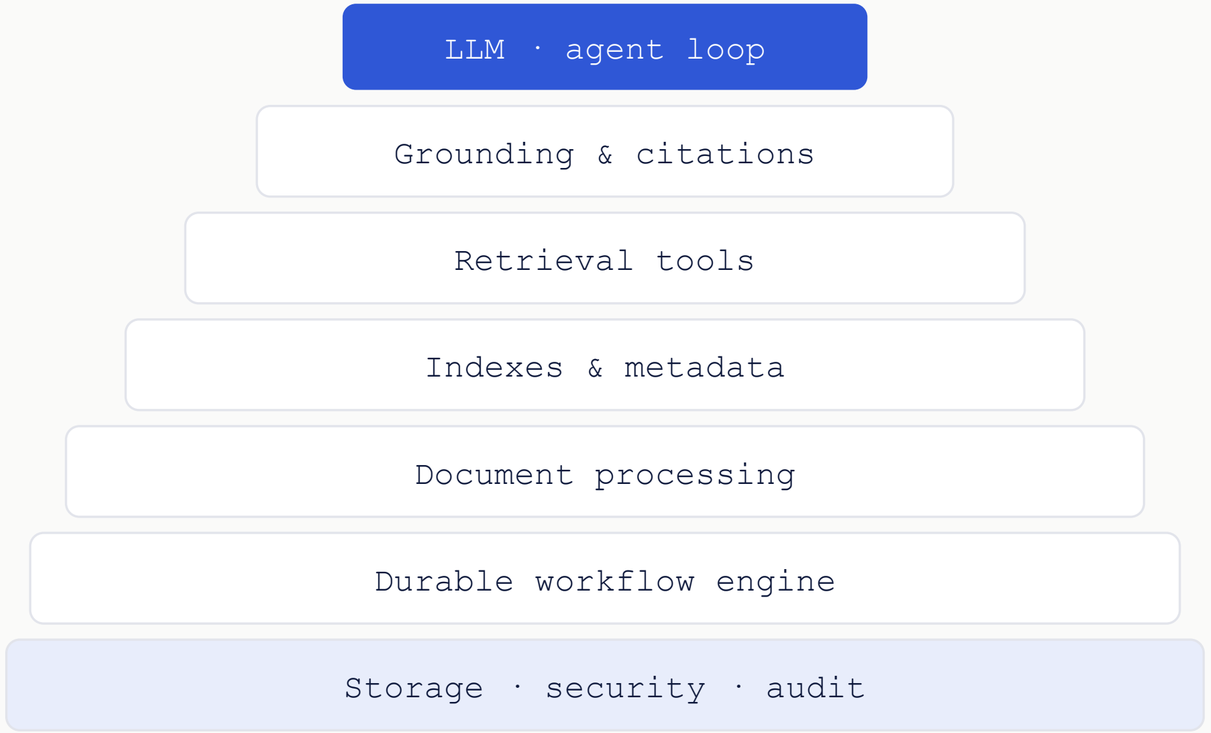
MEASURABILITY

Quality on a scoreboard

Continuous evals, cost visibility, operational monitoring.

Core Thesis

Enterprise agentic RAG is an evidence system with an LLM on top.



> **The LLM is the final reasoning layer. The enterprise value comes from the evidence system underneath it.**

One thin layer reasons. Six layers create, select, verify, and protect the evidence it reasons over.

Reference Architecture

The whole system, once — before we zoom in.

Agentic RAG vs Classic RAG

Agentic RAG is retrieval *strategy selection*, not just retrieval plus generation.

CLASSIC RAG

- One search tool
- Fixed top-K chunks
- One prompt
- One answer

Works when every question looks like the demo.

AGENTIC RAG

- ✓ Multiple retrieval tools
- ✓ Tool selection
- ✓ Context expansion
- ✓ Whole-document reads
- ✓ Table queries
- ✓ Extraction workflows
- ✓ Grounding checks
- ✓ Iterative repair

Chooses the right evidence operation for each question shape.

The Two Hardest Layers

Everything else in the stack exists to support these two.

CHAPTER 1

Ingestion

Creates trustworthy evidence. If it fails, the right evidence never exists.

→

CHAPTER 2

Retrieval

Selects the right evidence. If it fails, the evidence exists but is never seen.

→

CHAPTERS 3–5

The agent

Reasons over that evidence — grounded, cited, and measured.

01

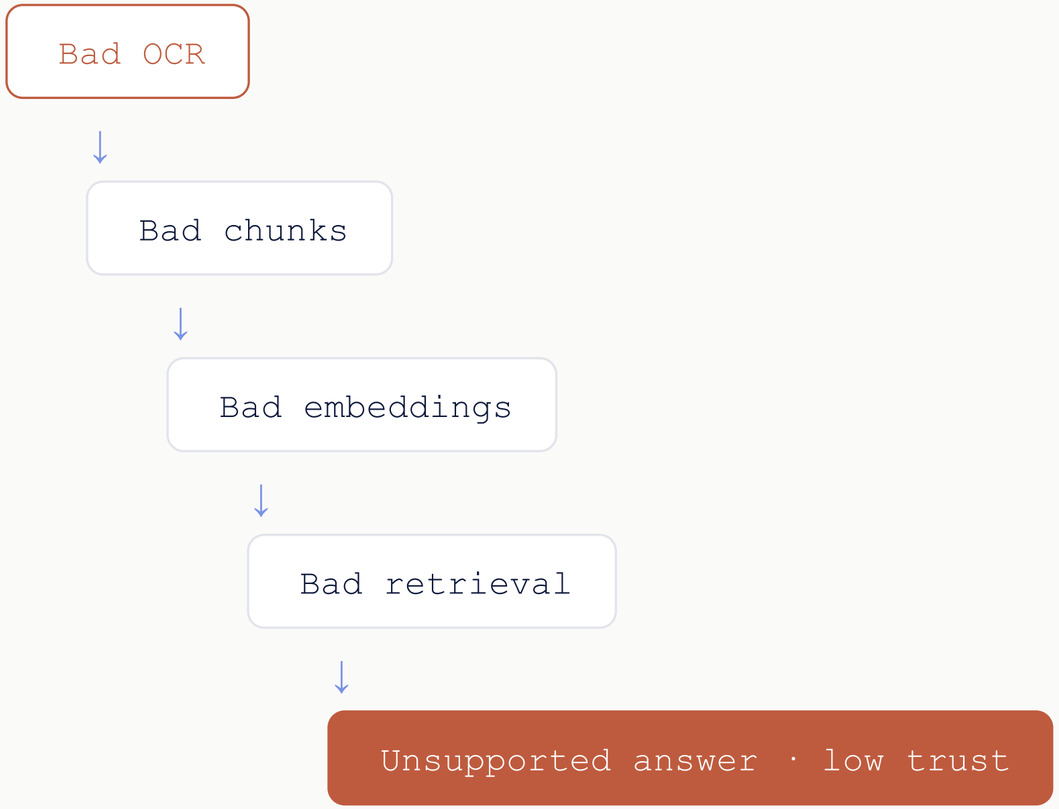
Ingestion

How messy documents become trustworthy, citable evidence.

Durable workflows | OCR & layout | Chunking | Provenance | Failure modes

Why Ingestion Is the Foundation

The LLM cannot recover from missing or mangled source text.



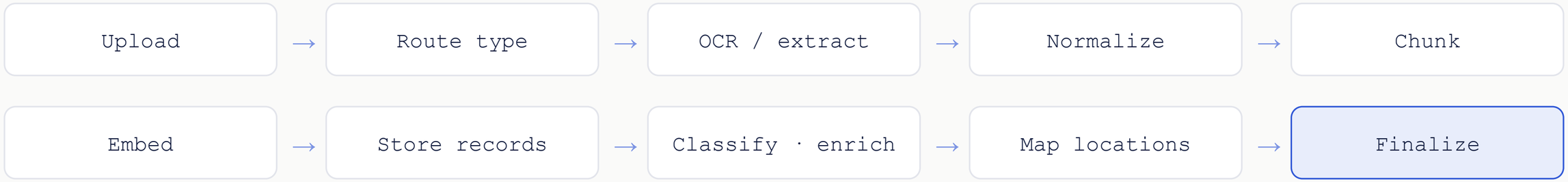
01 Enterprise documents carry tables, scans, headers, footers, signatures, and boilerplate.

02 The system must preserve enough structure to support retrieval and citations later.

03 Ingestion is the data quality layer — you win or lose at document processing.

Upload to Knowledge Object

Ingestion turns a file into a searchable, traceable knowledge object.

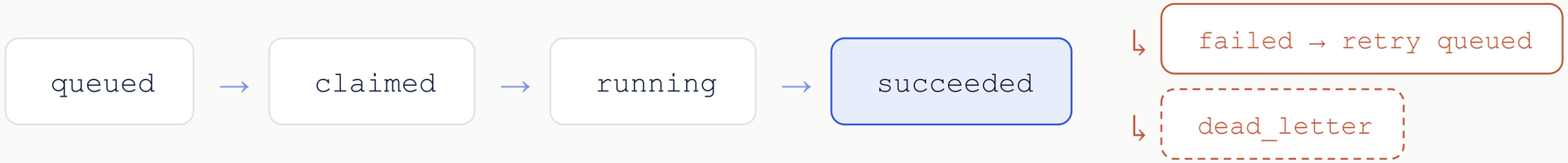


KEY IDEA

The output is not text — it is evidence units with identity, location, metadata, and indexes.

Durable Workflow Ingestion

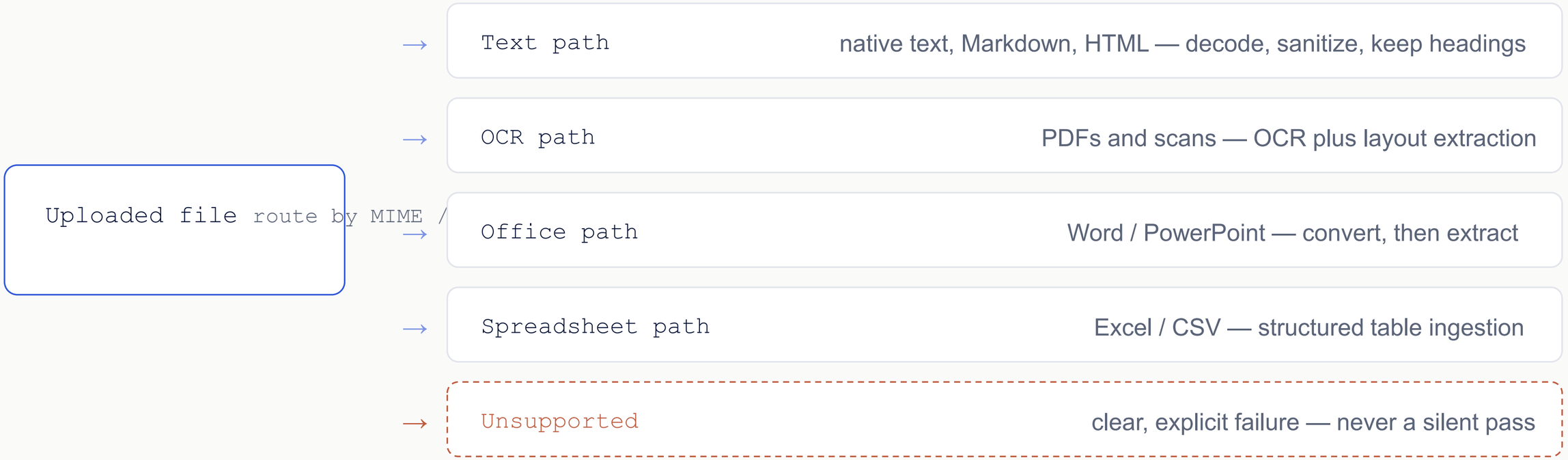
Ingestion should be a workflow, not a fire-and-forget function.



- ✓ Every step has status and can be retried
- ✓ Failures are visible, not silent
- ✓ Large payloads move by reference (claim-check)
- ✓ Workers can crash without losing state
- ✓ Leases and reclaim jobs recover stuck work
- ✓ Operators can inspect the exact failed step

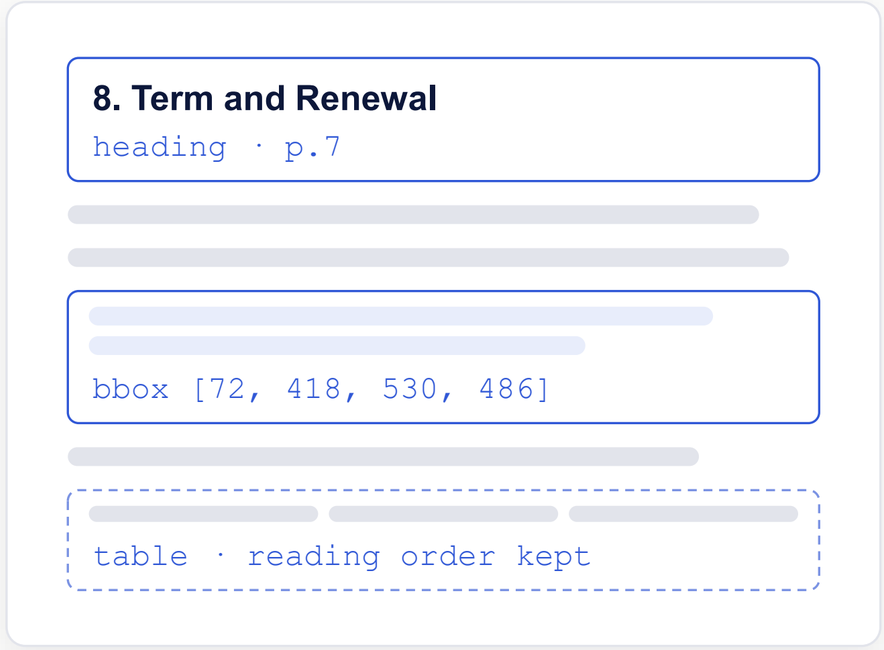
File Type Routing

Different files need different ingestion paths — one generic parser hides failure.



OCR and Layout Extraction

OCR is not just text extraction — it is the foundation for citations.



- ✓ Page boundaries and reading order
- ✓ Headings and table text
- ✓ Coordinates and bounding boxes
- ✓ Confidence and failure signals

Citations need a location. Context expansion needs neighbors. Viewers need highlightable regions. Bad OCR produces confident but false retrieval.

Text Normalization

Remove junk without destroying meaning — clean enough to search, faithful enough to cite.

BEFORE

```
<script>…</script>
## Section 5.2
Renewal Term shall…
```

→

AFTER

```
## Section 5.2
Renewal Term shall…
[page marker kept · p.7]
```

- ✓ Strip unsafe HTML and scripts
- ✓ Clean Markdown artifacts
- ✓ Preserve headings and page markers
- ✓ Detect empty output early
- ✓ Normalize whitespace
- ✓ Never keep OCR garbage as meaning

Smart Chunking

The chunk is the atomic evidence unit of the whole system — not an arbitrary text blob.

NAIVE

Split every N tokens

Clauses split mid-sentence, headings orphaned, tables shredded.

STRUCTURE-AWARE

Agreement

└ Term and Renewal · p.7–8

chunk 42 · parent [38–49] · child [42–44]

- ✓ Respect page boundaries
- ✓ Preserve headings
- ✓ Track section hierarchy
- ✓ Merge tiny fragments
- ✓ Keep chunk index
- ✓ Attach parent / child ranges

Chunk Metadata and Provenance

Every chunk should know exactly where it came from.

```
{ "workspace_id": "workspace-123" , "document_id": "doc-
456" , "chunk_index": 42 , "pages": [7, 8] ,
"section_path": "Agreement › Term and Renewal" ,
"parent_range": [38, 49] , "source_file": "Master
Services Agreement.pdf" , "location": { "page": 7,
"bbox": [72, 418, 530, 486] } }
```

WHAT IT UNLOCKS

- ✓ Filtering and deduping
- ✓ Context expansion and full-document reconstruction
- ✓ Machine-verifiable citations
- ✓ Audit trails
- ✓ Clean deletion and re-ingestion

Embedding and Indexing

Embeddings are one index over the knowledge object — not the entire knowledge base.

Knowledge object

`chunks + metadata + layout`

→

`Vector index`

Semantic retrieval — meaning and paraphrase

`Full-text index`

Lexical retrieval — names, dates, clause labels

`Metadata indexes`

Filtering — scope, type, status, tenant

`Document tables`

Status and security — the deterministic path

Ingestion Failure Modes

Enterprise quality is measured by the ugly cases, not the demos.

FAILURE	IMPACT	MITIGATION
OCR returns empty text	Looks searchable, isn’t	Detect empty extraction, fail early
OCR loses page order	Citations unreliable	Preserve page markers and layout
Chunking splits clauses badly	Partial evidence	Structure-aware chunking
Embedding insert retries partially	Duplicate chunks	Idempotent chunk keys / upserts
Huge payload in workflow state	Timeouts, OOM	Claim-check storage
Layout mapping fails	Answer works, highlight doesn’t	Degrade gracefully, mark confidence
Vendor OCR retains data	Compliance concern	Retention review + deletion story

CHAPTER 2 · SLIDES 18–32

02

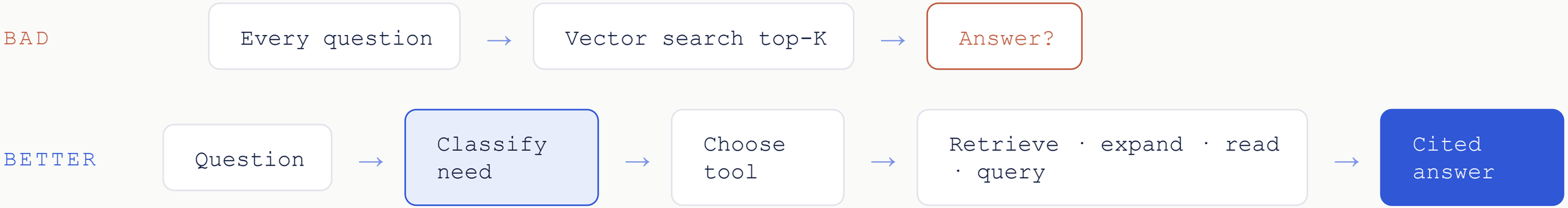
Retrieval

How the right evidence gets selected — strategy, shape, and fusion.

Question types | KB topology | Hybrid search | Reranking | Citations

Retrieval Is Not One Search Call

Different questions require different retrieval strategies.



KEY IDEA **The agent’s first job is deciding what kind of evidence the question requires.**

Question Types Drive Retrieval

Agentic retrieval means choosing the right evidence operation.

USER ASK	RETRIEVAL STRATEGY
“What is the renewal date?”	Search for a fact
“Show me the indemnity clause”	Keyword search + section read
“Summarize this agreement”	Read full document
“Compare these two contracts”	Read both full documents
“Which docs mention force majeure?”	Search across corpus
“How many active students?”	Query tabular data
“Find contracts missing a field”	Metadata / extraction query

Knowledge Base Shape Matters

Retrieval should match the topology of the knowledge base.

▸ contracts/ ▸ vendors/ acme-msa.pdf

Folder-shaped

Policies, contracts, manuals — hierarchy is signal.

list · open · read · search-in-scope

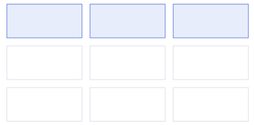


Table-shaped

Rows, columns, records, exports, operational data.

schema · filter · count · group-by · export

Graph-shaped

Entities, relationships, dependencies, ownership.

entities · neighbors · paths · communities

Folder-Shaped Knowledge Bases

If the KB already looks like a filesystem, let the agent use filesystem-like tools: list, open, read, search, navigate.

```
/workspace /contracts /vendors acme-msa.pdf boot-barn-renewal.pdf /policies data-retention.md ai-usage-policy.md /tables students.csv
```

01 List folders — names carry semantic hints

02 Open the likely document — hierarchy narrows the space

03 Read the section or full file

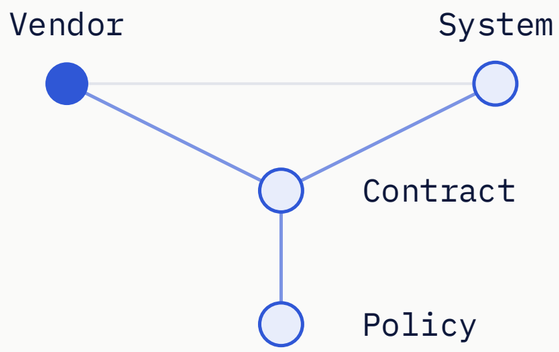
04 Search within scope when navigation stalls

05 Cite the exact file, section, and page

Caveat: pair navigation with hybrid search and filters — names get ambiguous, files get misplaced.

When to Add GraphRAG Tools

Valuable when the answer depends on relationships, not text similarity — a complement to hybrid search, never a replacement.



`find_entities`

`get_entity_neighbors` `find_paths`

`impact_analysis`

WHERE IT HELPS

Vendor / subprocessor maps · policy dependencies · shared counterparties · cross-document reasoning · compliance mapping

WHERE IT HURTS

Extraction is expensive · entity resolution is hard · false edges create false confidence · graphs go stale · graph answers still need document citations

Hybrid Search

Enterprise retrieval combines multiple signals — no single one is trustworthy alone.

Dense semantic	finds concepts
Sparse full-text	finds exact evidence
Fuzzy matching	survives OCR + typos
Metadata filters	enforce scope + security

→

Fused candidates

One ranked evidence set, then reranked

Dense vs Sparse Retrieval

They solve different problems — the best systems use both.

DENSE · VECTOR

“Find meaning like this”

Paraphrase and conceptual search.

“early termination rights” → finds “termination for convenience”

SPARSE · FULL-TEXT

“Find these words”

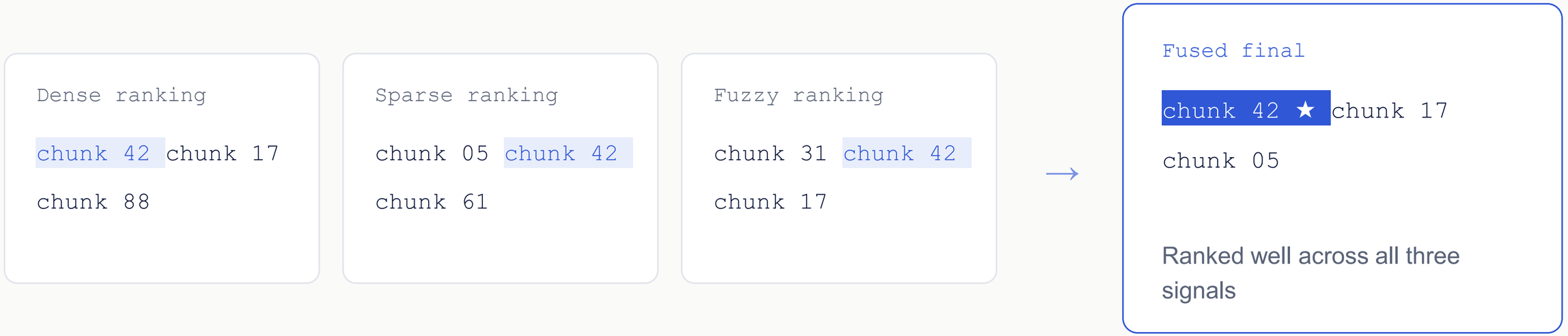
Names, dates, clause labels, identifiers.

“Section 8.4” · “Boot Barn” · “December 31, 2027”

Semantic search finds concepts. Keyword search finds evidence.

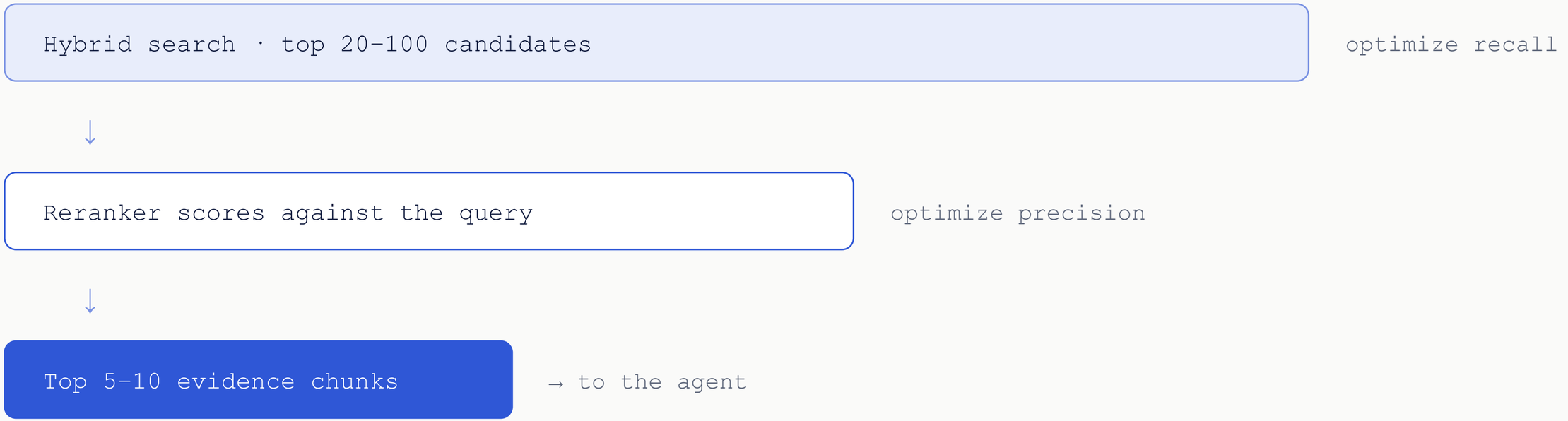
RRF and Score Fusion

Combine rankings instead of betting on one search signal. Reciprocal rank fusion rewards chunks that rank well everywhere.



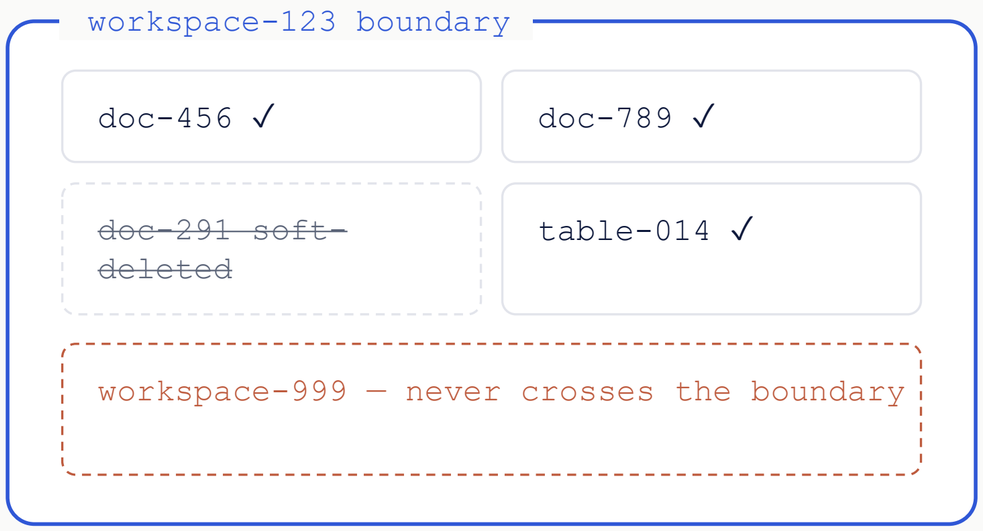
Reranking

Retrieve broadly, then rerank carefully.



Filters and Isolation

Retrieval is security-sensitive — a search bug can be a data leak.



EVERY QUERY ENFORCES

- ✓ Workspace boundary and user permissions
- ✓ Soft-deleted document exclusion
- ✓ Document ID, type, and status scope
- ✓ Tabular vs prose distinction

Search vs Read Full Document

Snippet search is the wrong tool for whole-document tasks.

`search_documents`

- One fact or one clause
- Locating where a topic appears
- Searching across many documents

`read_full_document`

- Summarization and comparison
- Extracting many terms at once
- Tasks where a missed clause changes the answer

LESSON **Repeated search calls cannot reliably reconstruct an entire document.**

Context Expansion

The right answer often lives *near* the retrieved chunk, not only inside it.

- ✓ Legal terms are defined elsewhere
- ✓ Tables continue across pages
- ✓ Exceptions follow general clauses
- ✓ OCR boundaries can be awkward

Structured Data Retrieval

Tables need queries, not prose RAG.

ASK	RETRIEVAL MODE
“How many active students?”	`Count query`
“Break down by visa type”	`Group-by query`
“Show five J-1 students”	`Row query`
“Export all active students”	`Export workflow`

NEEDS Schema detection · filtering · grouping · counts · row previews · computed fields · field-level provenance

Citations as Retrieval Output

Citations should be machine-derived, not model-authored.

BAD

LLM writes the answer and invents the citation text

BETTER

Tool returns structured sources → System harvests candidates → Answer linked to doc / page / chunk

§

Master Services Agreement.pdf · p.7 · chunk 42

“The Renewal Term shall be one (1) year…” · bbox [72, 418, 530, 486] · via search_documents

click to verify →

Retrieval Failure Modes

Failures become diagnosable when each retrieval stage is observable.

FAILURE	SYMPTOM	FIX
Correct doc, wrong chunk	Answer misses key term	Hybrid + rerank + expansion
Correct chunk, no context	Answer misses exception	Expand surrounding chunks
Vector misses exact phrase	“Not found” for present term	Add exact / keyword path
Agent loops on search	Many searches, no answer	Use full-document read
Reranker suppresses answer	Plausible but wrong top hit	Golden evals + reranker tests
Citation weakly supports answer	Trust issue	Citation faithfulness eval
Answer absent but inferred	Hallucination	Refusal + grounding evals

CHAPTER 3 · SLIDES 33–35

03

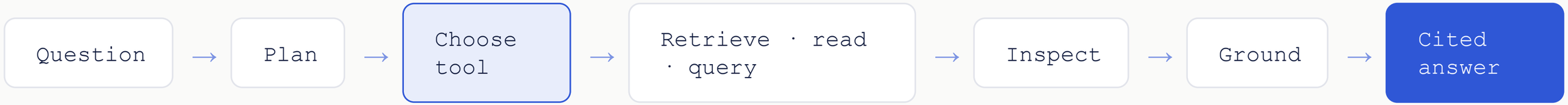
The Agentic Layer

Tool choice, grounding gates, and citations that prove the answer.

Tool loop | Tool catalog | Grounding gate

The Agent Tool Loop

The agent turns a user request into evidence-gathering actions.



↺ repeat until the evidence suffices — bounded by iteration and cost caps

- ✓ Choose search vs read vs query
- ✓ Stop when enough evidence exists
- ✓ Batch independent reads
- ✓ Say “not found” when evidence is missing

Tool Catalog

Enterprise RAG needs a tool catalog, not one search function.

DISCOVERY

`list_documents`
`count_documents`

SEMANTIC RETRIEVAL

`search_documents`

WHOLE-DOC READING

`read_full_document`
`read_section`

CONTEXT REPAIR

`expand_context`

STRUCTURED DATA

`query_tables` `get_schema` · `export`

EXTRACTION

`get_extractions`
`generate_report`

GRAPH (OPT-IN)

`get_entity_neighbors`
`find_paths`

UTILITY

`math` · `planning` `subagents`

Grounding Gate

No workspace factual claims without workspace evidence — enforced structurally, not by prompt.

WHY PROMPT-ONLY GROUNDING FAILS

- ✕ Models can ignore instructions
- ✕ Web results are not workspace truth
- ✕ Models pattern-complete plausible facts
- ✕ Final answers must be structurally constrained

CHAPTER 4 · SLIDES 36–39

04

Evals & Operations

How quality is measured — and kept stable under enterprise constraints.

Golden questions | Eval taxonomy | Ops metrics | Security

LLM and RAG Evals

Enterprise RAG needs a quality scoreboard, not a vibe check.

```
{ "question": "What is the renewal term in the Boot Barn agreement?" , "expected_answer": "One year" , "expected_source": { "document": "Boot Barn Agreement.pdf" , "page": 7 , "text_must_include": "renewal term" } }
```

METRICS TRACKED PER RUN

- retrieval recall
- answer accuracy
- citation faithfulness
- refusal correctness
- tool selection
- latency
- cost

Golden questions run on every change — prompts, models, retrieval — so regressions surface before users do.

Eval Taxonomy

Seven layers, mutually exclusive and collectively exhaustive — evaluate the system, not just the prose.

Data evals	Can it read the data?
Retrieval evals	Can it find the right evidence?
Reasoning evals	Can it answer from the evidence?
Tool evals	Can it choose the right action?
Grounding evals	Can it prove the answer?
Safety evals	Can it stay within policy?
Ops evals	Can it run reliably and economically?

Enterprise Operations

Production quality is also operational quality — every one of these is a dashboard, not a hope.

`cost / answer`
token usage + tool calls

`latency p50 / p95`
and timeout rate

`retry rate`
per workflow step

`dead-letter rate`
work that needs a human

`OCR failure rate`
by file type and vendor

`retrieval no-hit rate`
questions with zero evidence

`citation missing rate`
answers without receipts

`grounding rejections`
answers the gate blocked

`tool call count`
loop efficiency per answer

Security and Compliance

Knowledge bases are high-risk because they centralize sensitive data.

- ✓ Workspace isolation with RBAC / RLS
- ✓ Soft-delete exclusion in every query
- ✓ Audit logs on every access
- ✓ Encryption and secret handling
- ✓ Compliance-ready documentation
- ✓ Service-role boundaries
- ✓ Purge and deletion semantics
- ✓ Vendor retention review (OCR, embeddings)
- ✓ Prompt / tool leakage prevention

KEY IDEA **Retrieval and security are the same surface — a search bug can be a data exposure.**

CHAPTER 5 · SLIDES 40–45

05

Principles & Practice

The decisions that hold up, the defaults we avoid, and how quality stays honest.

Design principles | Anti-patterns | Readiness | Practice

Design Principles That Hold Up

The strongest choices are the ones that reduce invisible failure.

- ✓ Durable workflow ingestion
- ✓ Claim-check storage for large payloads
- ✓ Smart chunk metadata
- ✓ Retrieval tools matched to KB topology
- ✓ Structural grounding gate
- ✓ Cost and iteration caps
- ✓ Hybrid search instead of pure vector
- ✓ Reranking and context expansion
- ✓ Full-document read tool
- ✓ Structured table querying
- ✓ Code-harvested citations
- ✓ Explicit failure states everywhere

Anti-Patterns We Design Against

Each of these is a common default — and each one produces confident, unverifiable answers at scale.

- ✕ Prompt-only hallucination prevention
- ✕ Treating all documents as simple prose
- ✕ Assuming OCR text is enough without layout
- ✕ Retry paths without strict idempotency
- ✕ Ignoring cost until after the system is useful
- ✕ Repeated search calls for compare / summarize tasks
- ✕ Flattening folders and graphs into one vector pool
- ✕ Citations as text generated by the model
- ✕ Relying on demos instead of evals

Enterprise Readiness Checklist

Enterprise-grade means having answers to the predictable hard questions.

01 Can ingestion recover from partial failure?

02 Are writes idempotent?

03 Can you explain why an answer was produced?

04 Can you show the source document and page?

05 Can you measure retrieval quality?

06 Can you prove tenant isolation?

07 Can you delete customer data completely?

08 Can you cap cost and latency?

09 Can you detect regressions before users do?

How We Keep It Honest

Quality is not a launch state — it is a standing practice.

CORRECTNESS

- Golden-question eval suite on every change
- Ingestion-to-answer end-to-end tests
- Citation click-through validation

INTEGRITY

- Deterministic chunk IDs, idempotent writes
- Citations persisted across agent wake and subagents
- Vendor retention and deletion reviewed

VISIBILITY

- Retrieval quality dashboard
- Cost and latency accounting per answer
- Failure states surfaced, never swallowed

Reference Architecture Recap

Seven claims, one system.

Durable ingestion	creates trustworthy evidence
Hybrid retrieval	selects the right evidence
Agent tools	choose the right operation
Grounding gates	block unsupported claims
Citations	prove the answer
Evals	measure the system continuously
Operations	keep it reliable

Clausey

CLOSING THESIS

Enterprise agentic RAG is a system that can ingest reliably, retrieve precisely, reason with tools, prove its answers, and measure itself continuously.

If the evidence layer is weak, the LLM becomes a liability. If the evidence layer is strong, the LLM becomes an enterprise interface.