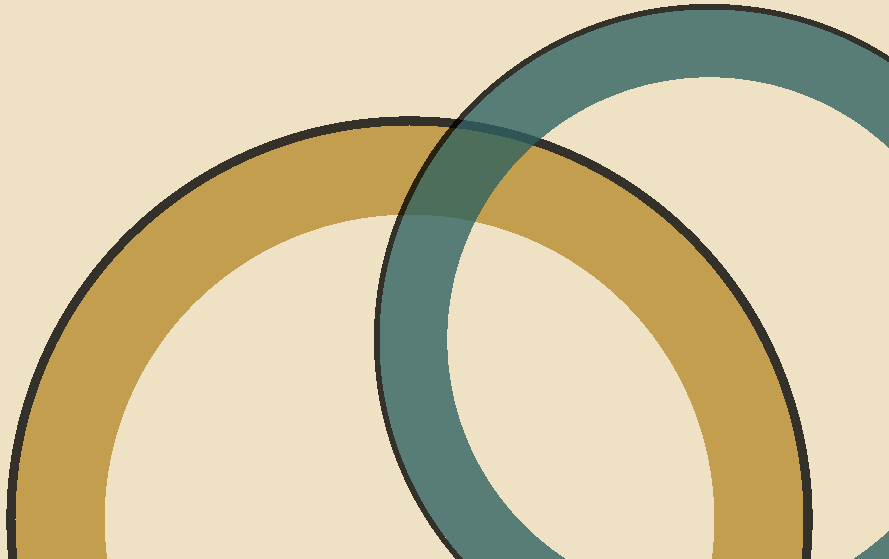
Engineering AI Agents · Week 5

Retrieval, memory, and workflow patterns

A library behind the desk · an office around it · arrows your code owns



Slides CC BY 4.0 · from the teaching kit of *AI Agents, Engineered* by Enrique Gutiérrez · aiagentsengineered.com/teach/

Predict before the next slide

Ask the most capable model you can reach: “What is our refund policy for annual plans cancelled mid-term?”

What comes back?

Chapter 8, “Retrieval-Augmented Generation”

By the end of today you can

Four objectives

1. Draw the **RAG pipeline** and name the lever for a retrieval failure
2. Choose **retrieval, the long window or fine-tuning** for a knowledge need
3. Sort memories into **episodic, semantic, procedural**; state the quarantine's **two doors**
4. Choose **chain, route, parallelize or loop**, and name the pattern's failure

Part 1 · Chapter 8 (full book)

Retrieval and knowledge

Retrieval-augmented generation: the oldest move in knowledge work, made mechanical

Look it up

Fresh

Add a document today; answerable within minutes

Private

Usable while staying private

Less invention

Recall becomes “read this one”

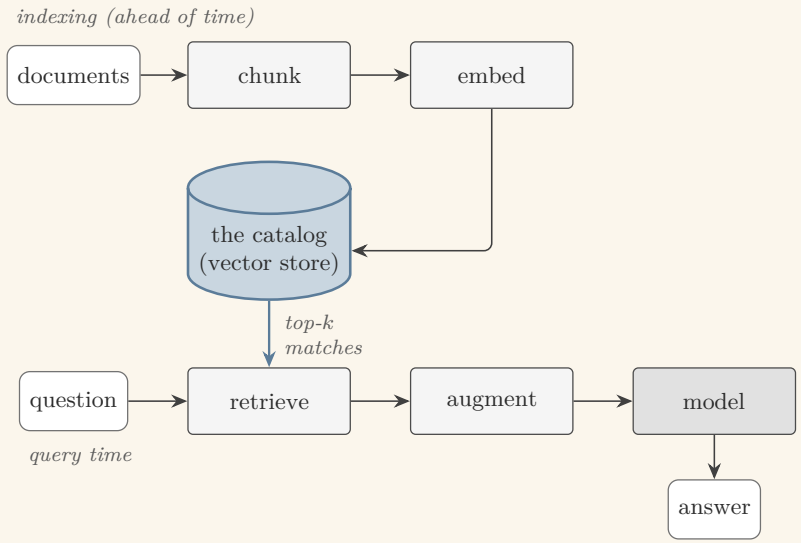
Provenance

Your code chose the passages, so it can point to them

> *“An answer that points at its sources hands you a signal you can verify; an answer from the weights hands you prose.”*
>
> Chapter 8, “Retrieval-Augmented Generation”

Chunk, embed, store; retrieve, augment, generate

- **Chunk**: passages of a few hundred words
- **Embed**: a position on a map of meaning
- **Top-k**: the few nearest chunks
- Same embedding model for both sides



The matches augment the prompt, so the model answers from what was retrieved.

‹ 4 / 4 ›

Figure from Chapter 8 · k of “three, five, ten”: the chapter’s illustration

Predict before the next slide

A customer asks: “What does error QX-1147 mean?” The retriever returns three thoughtful passages about error handling in general.

The page that defines QX-1147 exists, and its title is the code. Why is it not in the top ten?

Chapter 8, “Beyond Naive Retrieval” · the code is invented; the failure is real

When the plain pipeline disappoints, it disappoints in nameable shapes

Beyond naive retrieval

Symptom	Lever
Identifiers missed: codes, clause numbers, names	**Hybrid search**: keyword + meaning, fused
The caveat two lines above was cut off	**Chunking**: overlap, cut at headings, match small and read the parent
“It crashes when I export”	**Query rewriting**: resolve, rephrase, search several
Near the question on the map, yet irrelevant	**Reranker**: wide net of ~50, keep 3–4
Another product’s manual surfaces	**Metadata filter**: prune before scoring

Chapter 8 · “More retrieved context is not better context.” · ~50 and 3-4: the chapter’s example

One search, then one answer, is a workflow;
the agent can search again

Agentic retrieval

“Did last March’s outage affect any customer whose contract guarantees four-nines (99.99 percent) uptime?”

`incident report` → `deployment records` → `contracts`

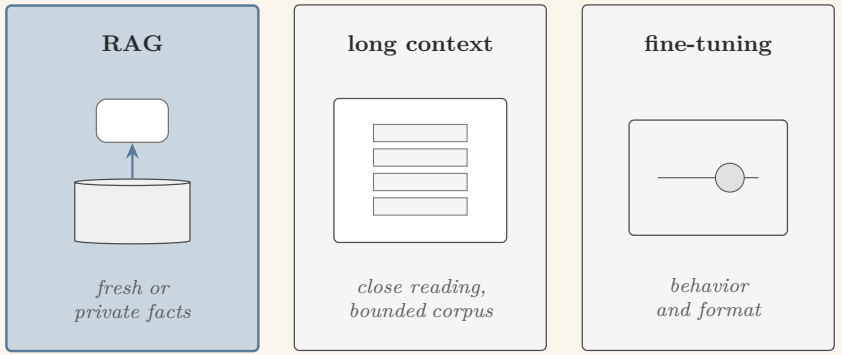
Multi-hop: the second query cannot be written until the first result is read

- The agent decides **whether**, **what**, **where**, and **whether it is enough**
- Retrieve, evaluate, refine
- Every judgment is a model call: cap the hops
- A front door: easy lookups static, hard shapes to the loop

Chapter 8, “Agentic Retrieval” · the bearing: “the simplest retrieval that works”

“We need RAG.” “Let’s fine-tune on our docs.”
Diagnose before you price

Three ways to give a model knowledge



‹ 4 / 4 ›

Figure from Chapter 8

> *“Retrieval for what the model should know; fine-tuning for how it should respond; the long window for close reading of a bounded set.”*
>
> Chapter 8, the rule

Check · one minute on paper · which road, and which condition decided it?

Choose the road

A

An agent reviews one 40-page supplier contract and must notice where an annex quietly amends a clause

B

A support agent answers from a help center that changes every week, and each answer must link its source

C

A triage step labels 200,000 tickets a day into a fixed house format that long prompts keep getting wrong

Chapter 8, “RAG versus Long Context versus Fine-Tuning” · sizes illustrative

Part 2 · Chapter 9 (full book)

Memory: working state across long runs

Predict before the next slide

Yesterday your coding agent spent an hour discovering that the tests only pass with one environment flag set. It fixed the build.

This morning, a fresh session, same agent, same project: “run the tests.” What happens next?

Chapter 9, “The Memory Taxonomy”

Memory is “an achievement rather than a feature”

Two lifetimes, three kinds

Kind	Office furniture	Holds	Write discipline
Short-term	the desk	the live window	dies with the run
Episodic	the logbook	what happened	append-only
Semantic	the filing cabinet	what is known	curated: reconcile, expire
Procedural	the procedures manual	how things are done	reviewed like code

“A wrong fact costs you an answer. A wrong rule costs you the agent.”

Chapter 9 · the three kinds after the CoALA framework (Sumers et al., 2023)

Check · pairs · two minutes

Sort five memories

1. “On Tuesday the user rejected the first draft because it used the passive voice”
2. “The user prefers day-month-year dates”
3. “Run the linter before committing; the build fails otherwise”
4. “The run at 14:02 called the deploy tool twice and was rolled back”
5. “The production database is the one ending in -green”

Episodic, semantic or procedural? Then: which one would you **never** let the agent write alone?

Chapter 9, “The Memory Taxonomy” · examples in the chapter’s style

Hot, warm, cold, and the constitution

- **Hot**: always loaded, “ruthlessly small”
- **Warm**: loaded per task
- **Cold**: retrieved on demand
- The **constitution**: the standing project-instructions file, hot
- Each line: would removing it cause mistakes?

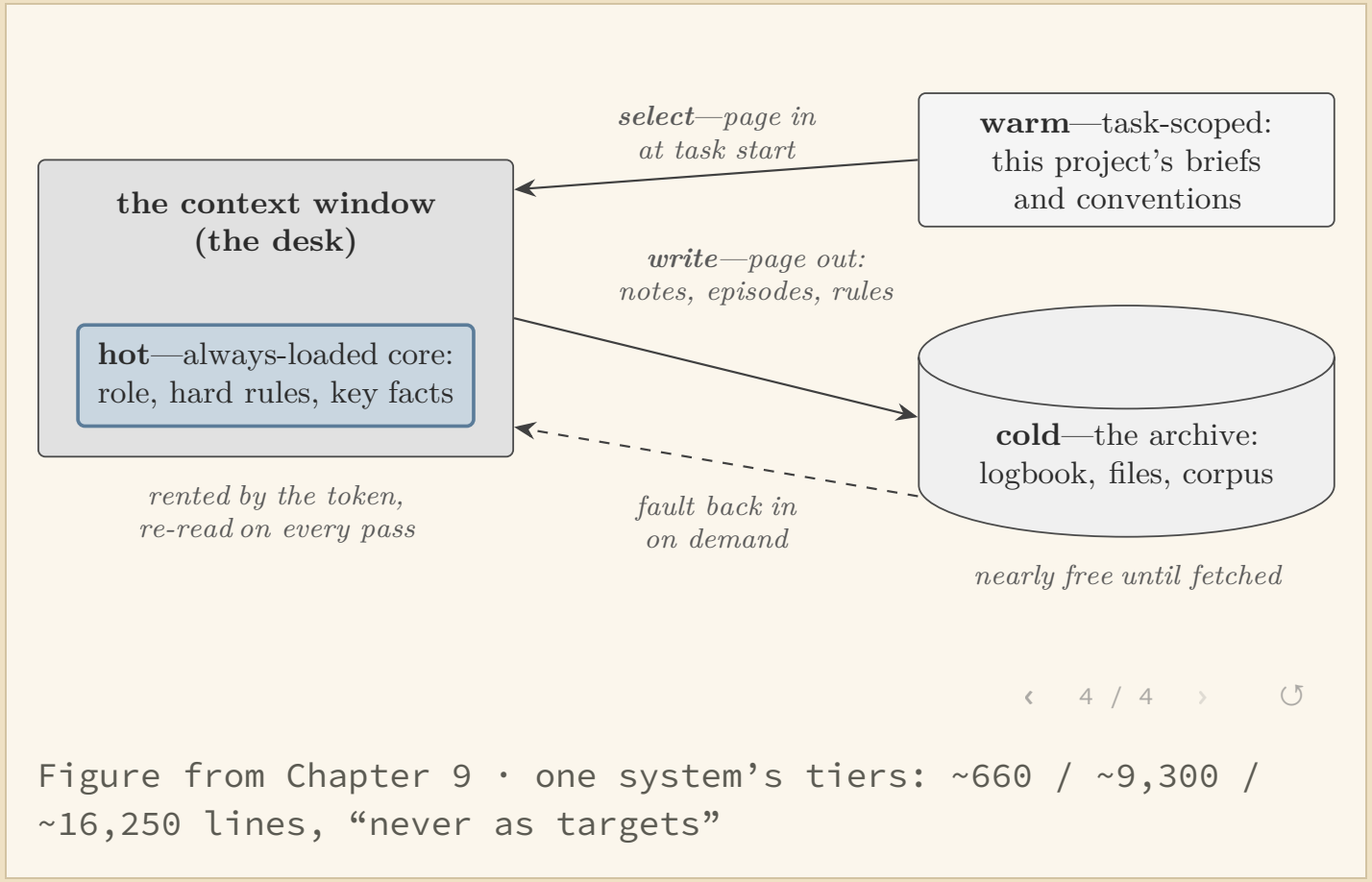


Figure from Chapter 9 · one system’s tiers: ~660 / ~9,300 / ~16,250 lines, “never as targets”

The lifecycle of one remembered thing: write, manage, read

Memory fails in ways a stateless agent cannot

- **Write**: in the hot path (immediate, slower turn) or in the background (no latency, lags)
- **Manage**: consolidate, deduplicate, expire; the stage most designs underbudget
- **Read**: relevance + recency + usefulness, filtered by owner first

Stale

Recalled with the store's authority

Contradiction

"Prefers email" and "prefers SMS"

Poisoning

An attacker's sentence becomes a belief

Store everything

Junk in retrieval, every liability multiplied

Chapter 9, "Memory Architectures" · lifecycle after one survey of agent memory

Hour three of one long run: the treatment,
cheapest and safest first

Keeping a long run healthy

`1 · trim tool results` → `2 · offload to files` → `3 · just-in-time retrieval` →
`4 · compaction` → `5 · pruning` → `6 · start fresh`

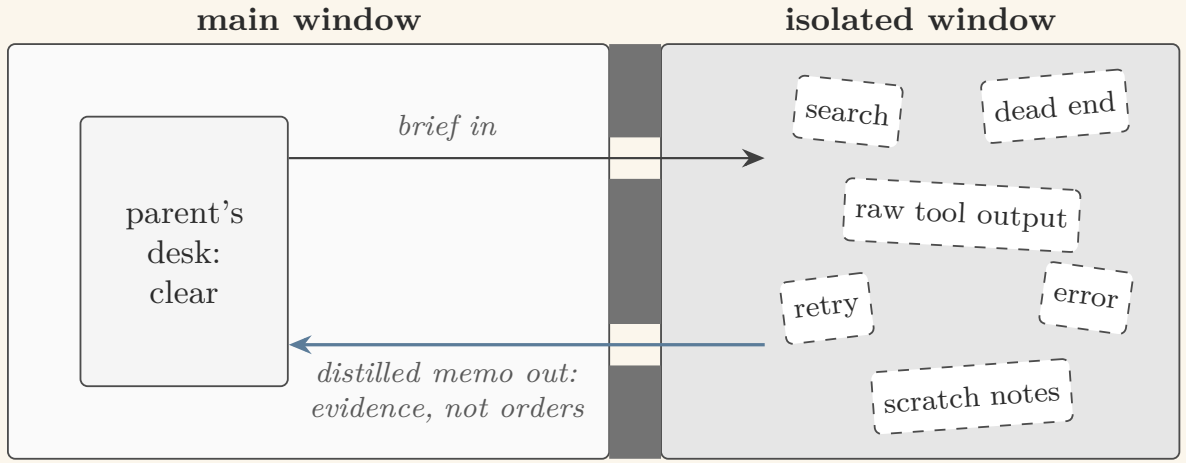
- Compaction keeps decisions, open problems, the plan, the artifacts; drops the journey
- Tune the summary for recall first, then precision
- Prefer summaries written to disk while the run is still healthy

“A healthy long run is a chain of short runs joined by good notes.”

Chapter 9, “Keeping a Long Run Healthy”

Quarantine and its two doors

> *“Nothing enters an isolated context except what its brief deliberately admits, and nothing comes back out except a distilled result sized for the desk that will read it—carried as evidence, never as orders.”*
>
> Chapter 9, the rule



nothing else crosses the wall: the mess stays sealed in

Only a distilled memo comes out, carried as evidence, never as orders.

‹ 5 / 5 ›

Figure from Chapter 9

Part 3 · Chapter 10 (full book)

Workflows and composition patterns

One unit: the augmented LLM, a model call with retrieval, tools and memory

Three overloads, three shapes

A single call is overloaded by…	Shape
Sequence: several jobs in order crowd one desk	Prompt chaining
Category: kinds of input that want different handling	Routing
Stakes or scope: too much for one draw, or independent concerns	Parallelization

“The arrows belong to your code, and the shaded boxes belong to the model.”

Chapter 10, “The Augmented LLM as the Base Unit” · “Exhaust the box before you reach for the wiring.”

Prompt chaining: each call processes the output of the previous one

The gates are the pattern

- A gate is a **check in code** at a seam: retry, fall back, or summon a person
- Ungated, a wrong label at step one is elaborated by every step after it
- Gates are narrow: form, never substance
- A forced chain shows as gates you cannot write

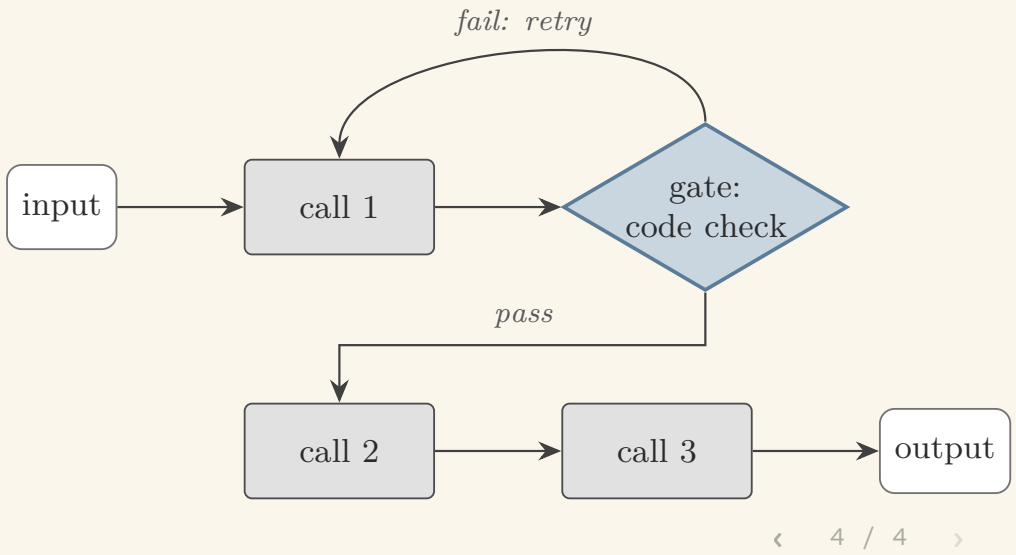


Figure from Chapter 10

Chapter 10 · “A chain without gates is just a slower way to be wrong; the gates are what you are actually building.”

The router is a classifier

- Ends the mega-prompt's tug-of-war
- Its error rate is a **floor** under every branch: measure it on its own
- Cheap tier for the common case: the first cost lever
- The cascade: cheap first, escalate on low confidence; the escalated share waits twice

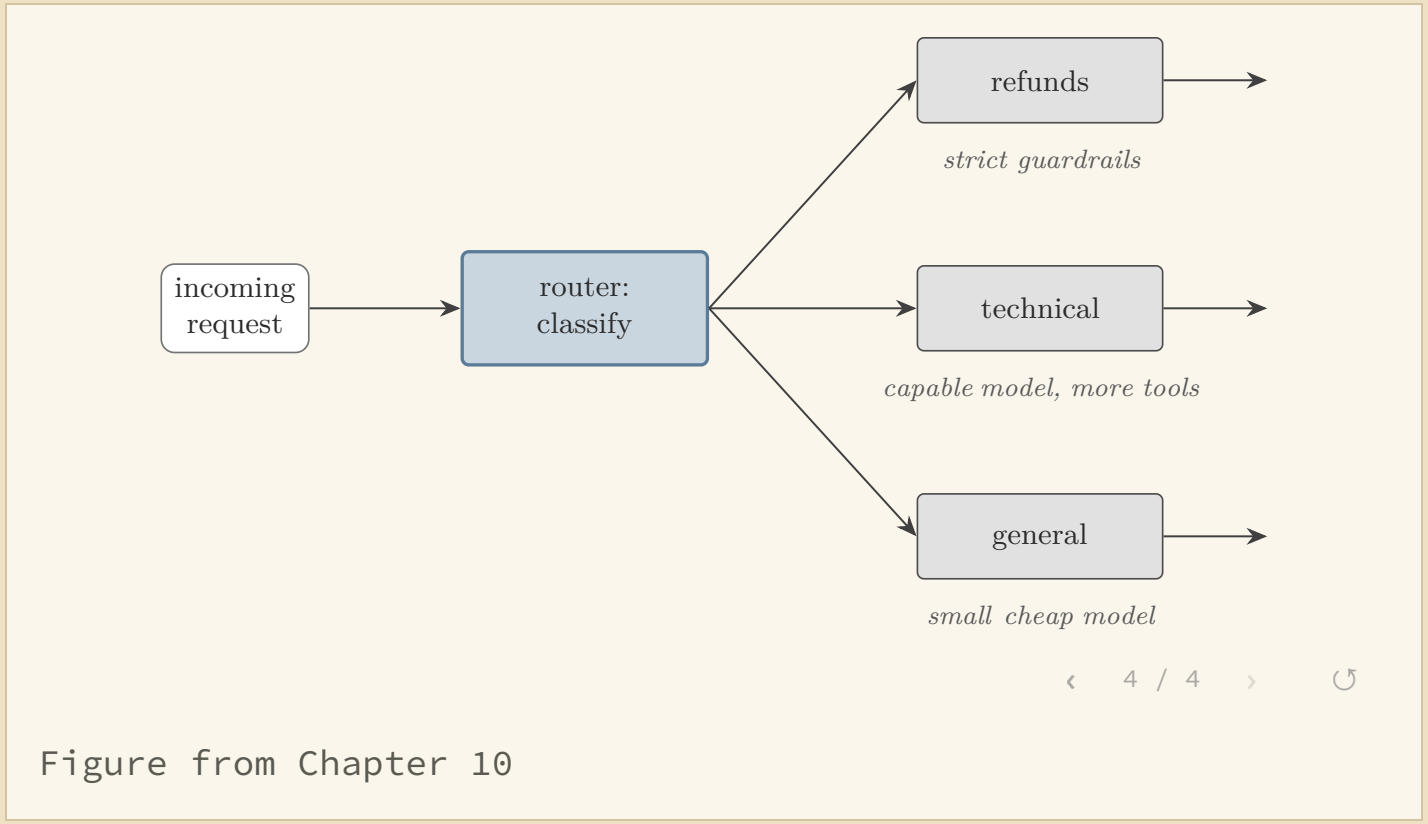


Figure from Chapter 10

Chapter 10 · learned routing cut costs "by over 2 times in certain cases" (one research system, dated)

Sectioning and voting

Sectioning

Split into independent subtasks: screen for abuse while another call drafts the reply

Fine print: independent. A hidden shared decision and the pieces will not compose

Voting

Run the same task several times: several review prompts, flag if any objects

Fine print: five votes, five times the tokens; answers must be comparable by equality

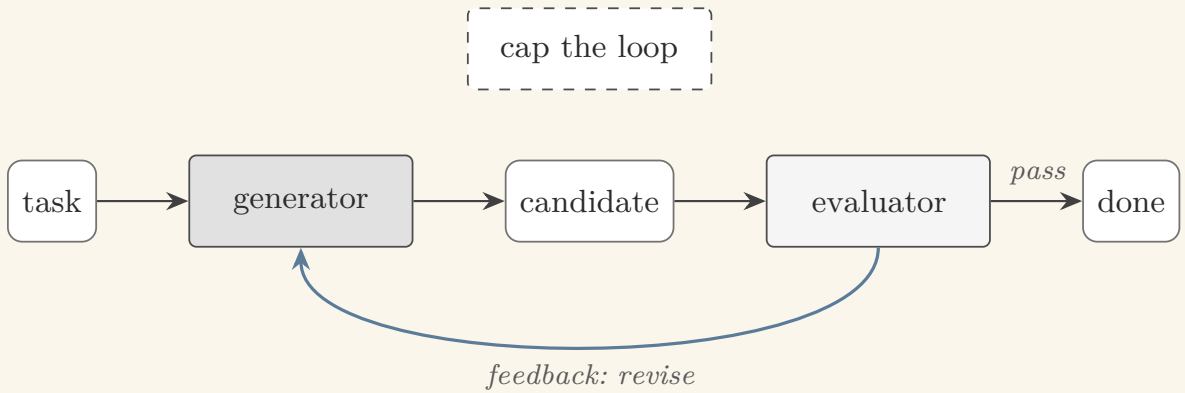
All three shapes share one limit: **you** wrote the subtask list, before the input arrived.

Chapter 10 · “If you cannot write the subtask list before seeing the input, no shape in this chapter will write it for you.”

The one shape whose defining arrow points backward

The evaluator–optimizer loop

- Generator and evaluator are **separate**
- Fits when feedback improves the work **and** something can produce it
- Seat a **verifier**; else a rubric critic
- Gains front-load; cap rounds, no progress, cost; keep the best



A hard cap is the mandatory second exit, since a verdict alone guarantees no exit.

‹ 5 / 5 ›

Figure from Chapter 10 · cap of three to five rounds: the common figure, illustrative

Chapter 10 · “the loop optimizes exactly what the evaluator measures, and nothing else”

None announces itself; all show up in cheap signals your code can watch

Four ways the loop fails

Rubber stamp

Everything passes on round one

Oscillation

Round two's fix brings back round one's flaw

Over-edit

Late rounds polish away substance

Wrong hill

A weak critic, climbed confidently

The Rule of Five, a field recipe: four or five self-reviews at varying altitude, until the model says it can do no better. The weakest evaluator on the ladder.

"Five reviews and then the test suite is a discipline. Five reviews instead of the test suite is a séance."

Chapter 10, "The Evaluator-Optimizer Loop" · recipe reported by Yegge, attributed to Emanuel

In class · teams of three · 20 minutes · whiteboard

A support pipeline: chain, then router

1. Draw it as a **chain**: shade the model calls; at least two **gates** as conditions code can check, each with its failure action (7 min)
2. Open the compounding error calculator: whole-chain success without gates, and with one verified retry per step (2 min)
3. Redraw the front as a **router**: refunds, technical, general. What differs per branch? Where is the router measured? (5 min)
4. Mark the **one** step that needs a loop: who evaluates, and the cap (2 min)



Chapter 1's ticket pipeline, as Chapter 10 reads it · 90% per step: illustrative · glossary hunt: homework, with Lab 4

Lab 4 and reading

Lab 4 · extend Lab 3

- A **retriever**: keyword + meaning, fused; a select tool
- A **constitution**: loaded every call, small, not agent-writable
- **Compaction**: bounded, and keeps what matters
- **Quarantine**: a worker, a brief, a memo

Reading for week 6

- Ch. 11, Multi-Agent Systems `full book`
- Ch. 12, Oversight and Autonomy `full book`
- Ch. 13, Writing the Outer Loop `full book`
- Ch. 14, Choosing Your Approach `full book`

This week’s reading: Ch. 8 · Ch. 9 · Ch. 10 (full book) · glossary: App. B (free) · no paid API needed

Five things to keep

- Retrieval converts weights into receipts; **hybrid** because meaning blurs identifiers
- Retrieval for what to **know**, fine-tuning for how to **respond**, the long window for a bounded set
- Memory is engineered: a wrong rule costs you the agent, so **procedural writes get review**
- A long run is **short runs joined by good notes**; quarantine has two doors
- Arrows in code: chain with **gates**, route, parallelize, loop with a **cap**

Engineering AI Agents · Week 5

Keep the arrows in your code for every step you can draw in advance

Slides from the teaching kit of *AI Agents, Engineered* by Enrique Gutiérrez, CC BY 4.0 · aiagentsengineered.com/teach/

Reuse, adapt and translate with attribution · creativecommons.org/licenses/by/4.0/

