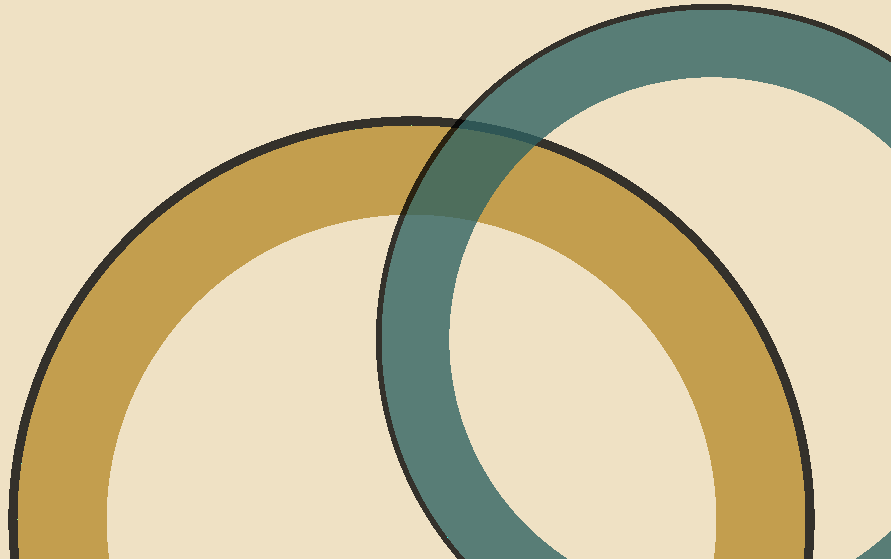
Engineering AI Agents · Week 4

Skills, protocols, and context engineering

Expertise on a shelf · sockets in the wall · deciding what lands on the desk



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

A week of watching a well-equipped agent

- Formats the quarterly report in a style nobody at your company would recognize
- Skips the release checklist every engineer on the team knows by heart
- Fumbles the one internal API the business runs on

“Every tool call succeeds, and the work is still wrong.”

Chapter 6, opening

By the end of today you can

Four objectives

1. Say which layer a gap belongs to: **tool**, **protocol** or **skill**
2. Compute the **M×N → M+N** collapse and state three costs of protocols
3. Name the five **context failure modes** from their symptoms
4. Apply **write, select, compress, isolate** to a mid-run desk

Part 1 · Chapter 6 (full book)

Skills, protocols, and interoperability

A named folder of know-how

main file: name + description, then the instructions

references: a style guide, a schema, a lookup table

templates: skeletons for the output

scripts: the steps better done deterministically

> *“Building a skill for an agent is like putting together an onboarding guide for a new hire.”*
>
> Quoted in Chapter 6, “What a Skill Is”

Files: versioned, reviewed, shared, fixed once for every agent that loads them

A year of your team's know-how: a hundred procedures, each a few pages, about a couple of thousand tokens.

Paste them all into the standing prompt. What does the agent carry on *every* call, to answer one customer email?

Chapter 6, "Progressive Disclosure" · sizes illustrative

Progressive disclosure · reveal in layers,
each pulled in only when needed

A table of contents, then one chapter

- **L1, the menu**: name + description of every skill; ~100 tokens a line
- **L2, the body**: one skill, loaded when judged relevant
- **L3, the bundle**: only the slice this branch needs
- Scripts are **executed, never read**

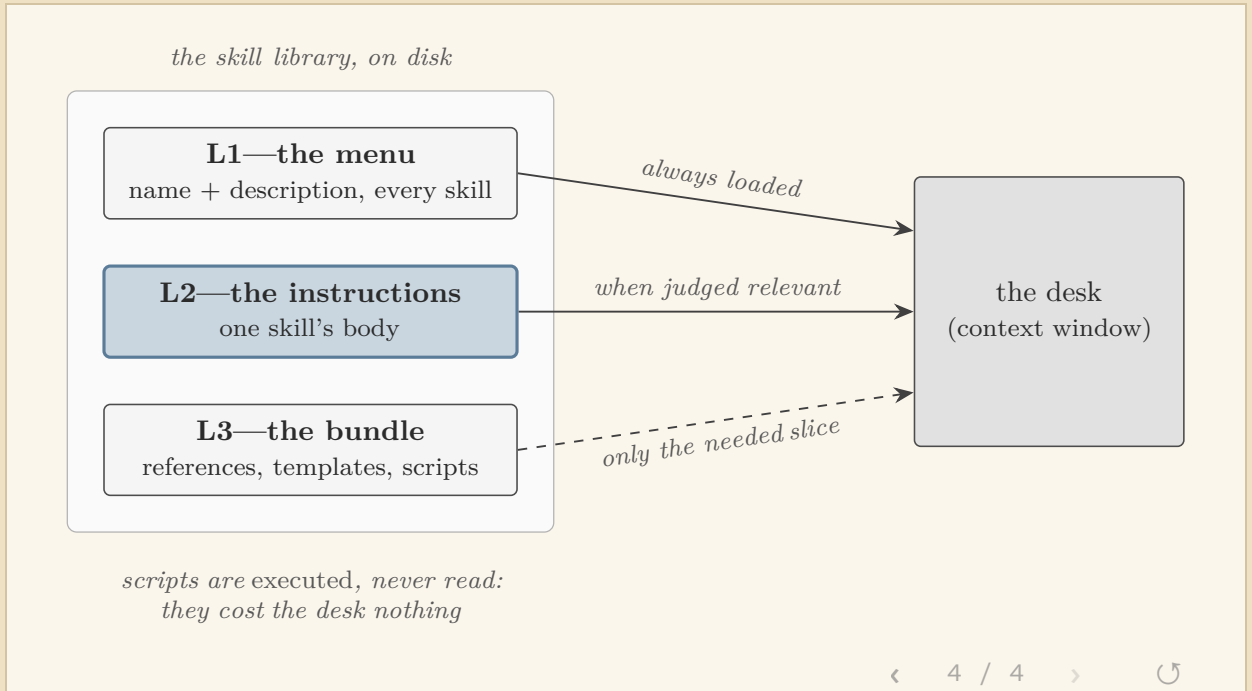


Figure from Chapter 6 · ~100 tokens per line: one guide's illustration

At the moment of choosing, the menu line is all the agent has

The description decides whether it loads

The shape to imitate (vendor guidance quoted)

Extract text and tables from PDF files, fill forms, merge documents. Use when working with PDF files or when the user mentions PDFs, forms, or document extraction

Never fires

Helps with documents

- **What**, then **when**; third person; the words a user would type
- Body: under a few hundred lines; push weight out into references

Chapter 6 · “It is the highest-leverage sentence in the folder.”

The difference between the diagram and production

Three caveats

Triggering is probabilistic

Test on your host, both ways: tasks that should fire it, tasks that should not

The host must be capable

Navigate files, follow references, run code; smaller models wander

The menu is not free

Fifty skills: several thousand tokens before any fires

Quick run (3 min): open the skill token budget with 10 skills, then change it to 50. Write down the menu, one task's total, and the reduction, for both.

Chapter 6 · “Curate the library like you curate the toolset: install what earns its line.” · fifty-skill figure: one implementation’s

Skills versus tools versus protocols

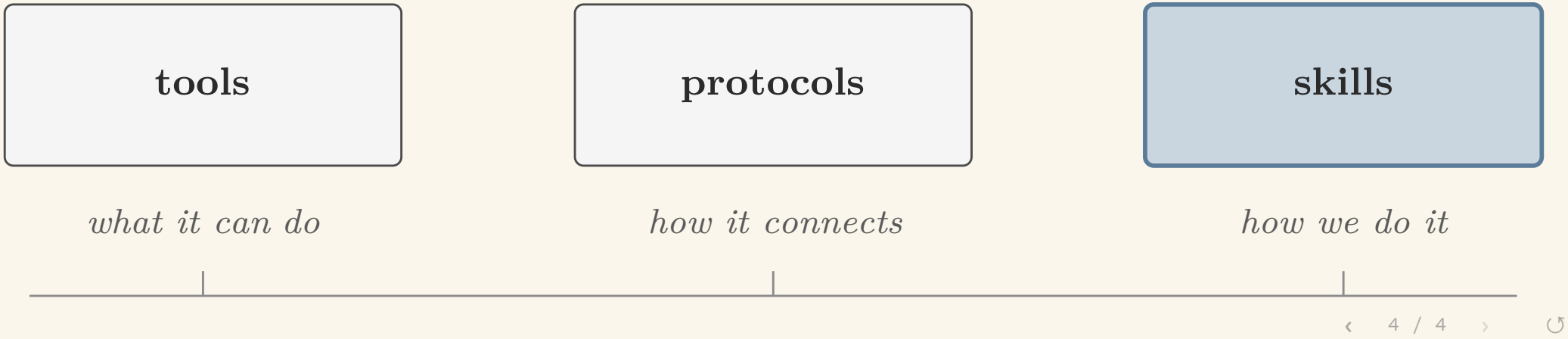


Figure from Chapter 6

- **Tool**: one action, a verb: `query_db`, `send_email`
- **Protocol**: an agreed way to discover and call capabilities, without bespoke glue
- **Skill**: the know-how for using the verbs well, to your standard

Chapter 6, “Skills versus Tools versus Protocols” · three layers, three questions

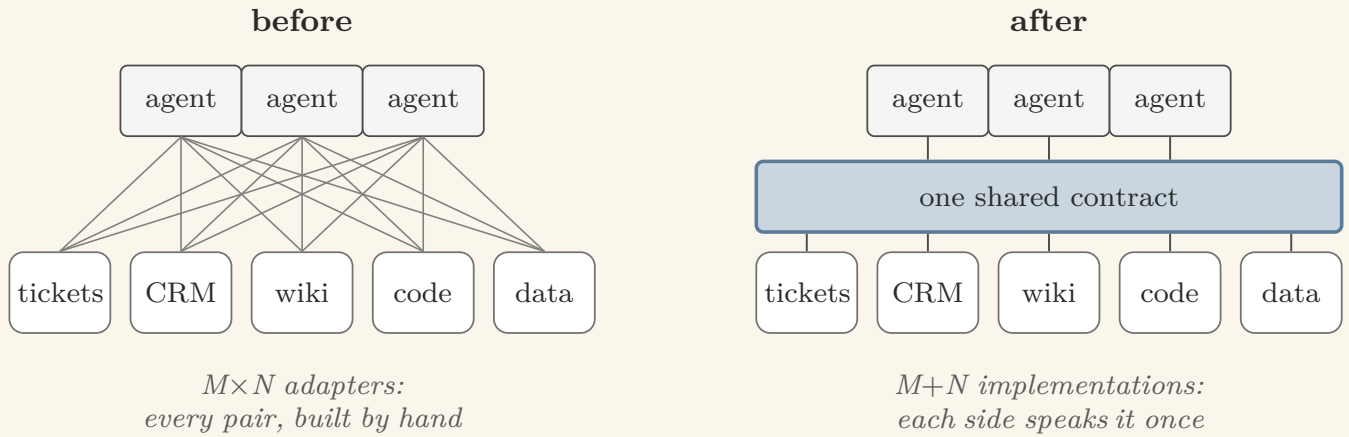
“What kind of thing is missing?”

The agent…	The gap is	The fix
cannot reach the system at all	connectivity	install or implement a server
reaches it, but has no safe, task-shaped action	a tool	cut one (week 3)
has reach and actions, works like a talented stranger	expertise	write the skill

Your turn: (a) it opens tickets but ignores your triage rules; (b) it has never heard of the new incident tracker; (c) it can read invoices but has no way to issue a refund

Engineering AI Agents · Week 4 · CC BY 4.0

A multiplication becomes an addition



‹ 4 / 4 ›

Figure from Chapter 6 · three agents, five systems

30 seconds: 4 agents and 12 systems. How many adapters without a standard, how many implementations with one, and what does a 13th system cost in each world?

Two kinds of protocol · what stands at the far end of the wire?

Agent ↔ tools and data

The far end is a capability

A **server** advertises what it offers; a **client** in the agent asks “what do you offer?” and calls it

Labeled example: the Model Context Protocol

Agent ↔ agent

The far end reasons too

A capability advertisement, a **task** with its own lifecycle, **artifacts** back from a black box

Labeled example: Agent2Agent

“Server” and “client” name roles, not places: a server may run on your laptop or be a stranger’s hosted service

Chapter 6, “Protocols and Interoperability” · each example one instance of a category that will keep evolving

Three costs, and a fourth unlike them

1. **Young and moving:** specs revise, authorization matures unevenly; version your integrations
2. **Drift:** the far end changes without asking; degrade gracefully when a tool vanishes
3. **Over-standardization:** a lowest common denominator plus indirection; one tight integration you control is simpler as a direct call

And the fourth: “every socket is a door.”

Engineering AI Agents · Week 4 · CC BY 4.0

Five shapes for a skill's contents

- **Tool wrapper**: conventions for a system; start here
- **Generator**: template fixes structure, style guide the bar
- **Reviewer**: *what* in a checklist, *how* in the body
- **Inversion** and **pipeline**: gates that stop the agent

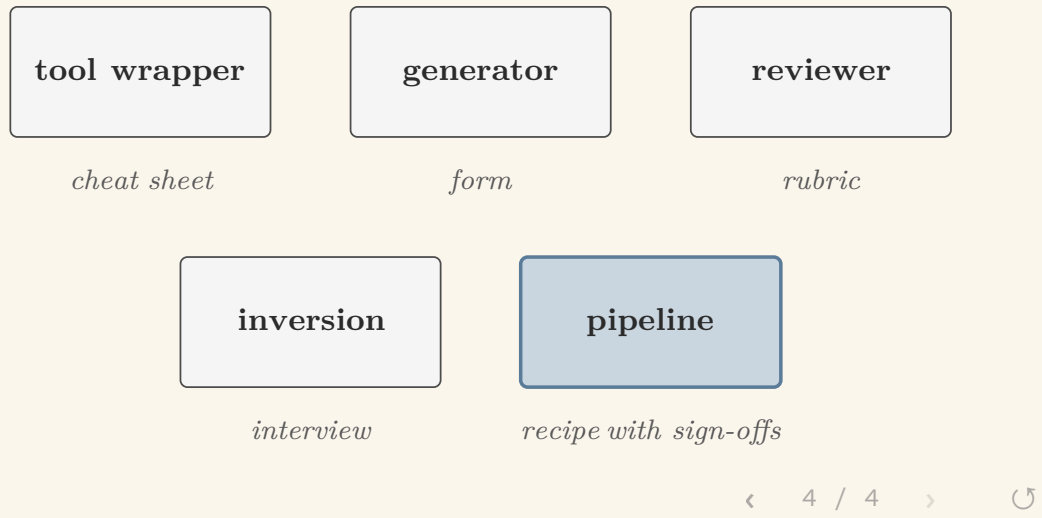


Figure from Chapter 6 · a folk taxonomy, not a standard

Chapter 6, "Skill Design Patterns and Composition" · "The checklist drove the behavior, not the agent's pre-training." (source quoted there)

Where good skills come from

From failures, incrementally, with measurement

- Start from **observed gaps**, not the encyclopedia you imagine needing
- Keep each skill **narrow**: one job done reliably
- Measure: the same tasks **with and without** the skill; the pass-rate delta

Skills that write skills

Curated skills can substantially improve success; self-generated ones may degrade it.

“let the agent draft the skill, and let something that is not the agent decide it is good”

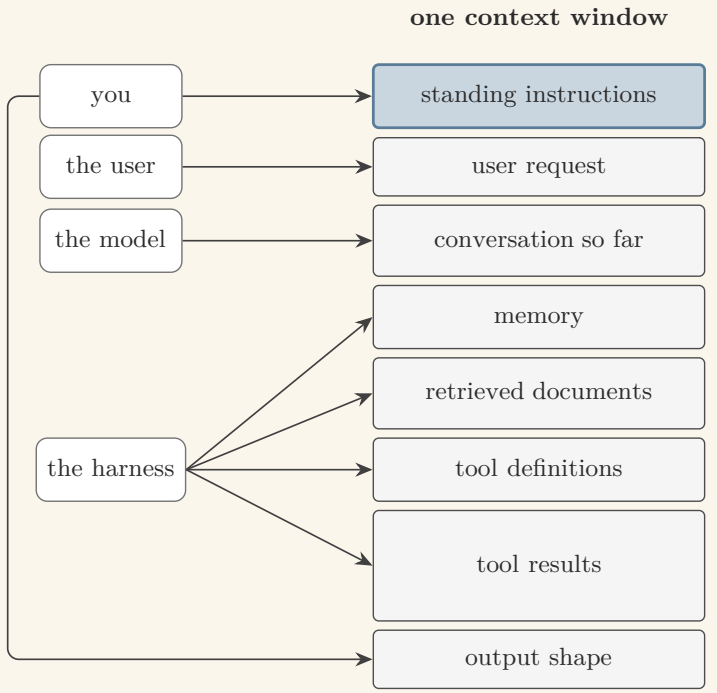
Chapter 6 · survey finding (Jiang et al., 2026) as quoted there · a skill is a dependency that ships instructions

Part 2 · Chapter 7 (full book)

Managing the context window

One window, four authors

- **Scope**: a prompt is one slice
- **Dynamic**: assembled by code on every call
- **Never concludes**: “A prompt is edited. A context is curated, and curation is a loop.”



The harness fills the rest; prompt engineering owns one slice, context engineering the whole window.

Figure from Chapter 7

‹ 4 / 4 ›

Predict before the next slide

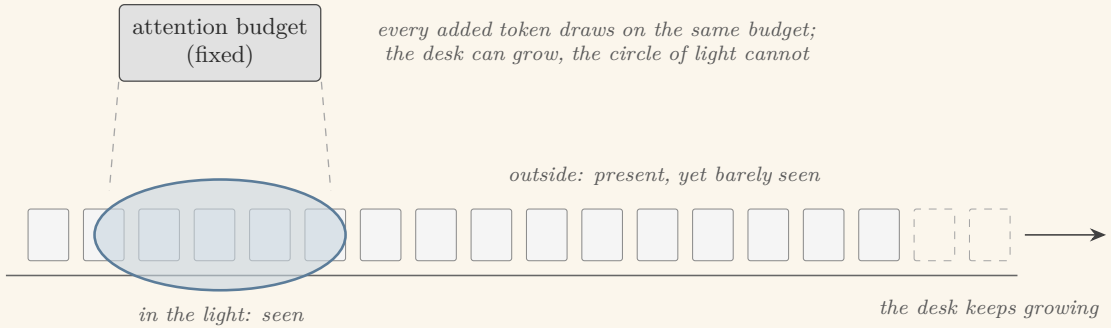
Ask a question with the one document it needs. Then ask it again, with the same document buried in two hundred pages of related, irrelevant material.

Nothing is wrong, everything fits. Same quality of answer?

Chapter 7, “A Scarce, Attention-Limited Resource”

The measured answer: no

Context rot and the attention budget



Every added token draws on the same budget, so scale alone dilutes attention.

Figure from Chapter 7

‹ 5 / 5 ›

- Eighteen models, task fixed, only length varied: decay long before the window is full
- Aim for “the smallest possible set of high-signal tokens that maximize the likelihood of some desired outcome”

Chapter 7 · Hong, Troynikov and Huber (2025) · second quotation: a vendor essay quoted there

How much of the window can you actually use?

The soft limit and the dumb zone

Soft limit vs hard limit

“The advertised window size is a fact about storage: how much fits. It promises nothing about attention.”

Retrieval vs reasoning

Finding a needle is the easy case; reasoning over a pile degrades sooner

- **Dumb zone**: it agrees with every correction (“you’re absolutely right”) while the work gets no better
- Keep **utilization** low: about **40%** circulates as the ceiling, *an illustration, never a constant*

Chapter 7 · “Treat the number as an illustration; treat the practice as sound” · “The emptier the desk, the sharper the model.”

Twenty minutes into a good run · name each failure

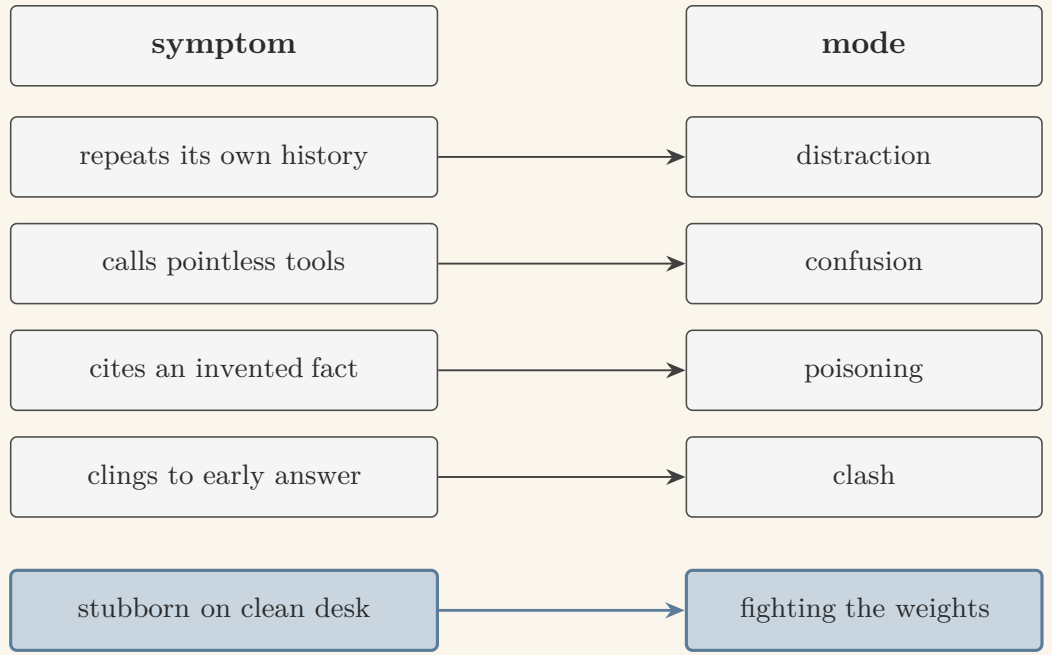
1. It re-runs a test it has already run three times
2. It searches the docs for an answer sitting forty lines up the transcript
3. It builds its next three actions on a “fact” it invented an hour ago

The weights did not change. Only the desk did.

Chapter 7, “How Context Fails” · the scene opens the section

Five ways a context fails: a bedside chart

- **Poisoning**: a later correction doesn't remove it
- **Distraction**: the log drowns the training
- **Confusion**: 46 tools failed, the 19 relevant passed
- **Clash**: sharded requests, ~39% worse
- **Fighting the weights**: survives a clean desk



Only fighting the weights survives a clean desk, so a fresh minimal retry diagnoses it.

‹ 5 / 5 ›

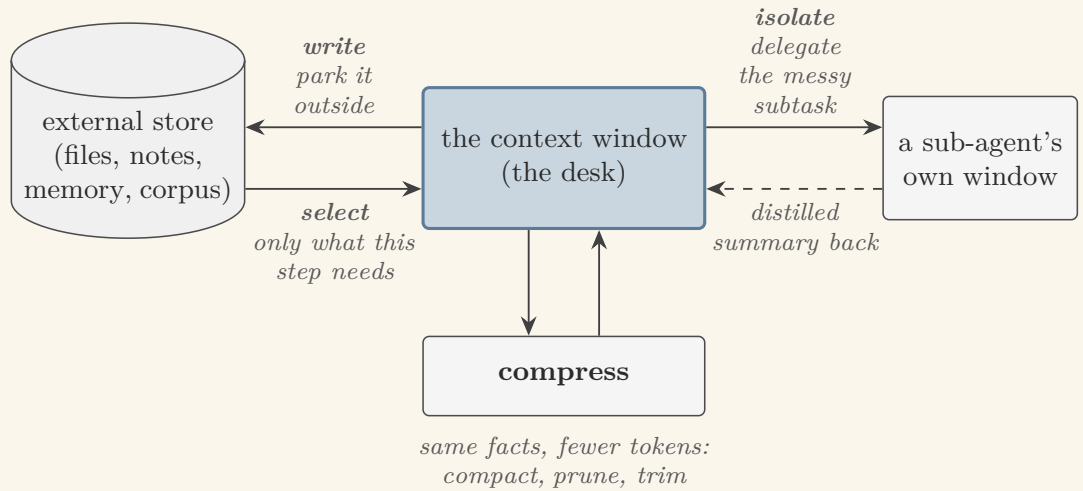
Figure from Chapter 7

Chapter 7 · names: Breunig (2025) · numbers from studies cited there, illustrative · the fifth mode has less measurement yet

Every remedy is one of four verbs, or a combination

Write, select, compress, isolate

- **Write**: the filing cabinet and the sticky note
- **Select**: the one folder today needs
- **Compress**: the one-page summary
- **Isolate**: a colleague's memo, not their drafts



Isolate gives a messy subtask its own window; every context tactic combines these four operations.

‹ 5 / 5 ›

Figure from Chapter 7

Chapter 7 · grouping from Lance Martin (2025) · "I have yet to meet a context remedy that is not some combination of these."

The four operations at the grain you will build

Operation	Forms	Buys	Costs
Write	scratchpad, progress file (*offloading*); memory stores	survival past compaction or reset	extra reads and writes; a store to manage
Select	retrieval; *tool loadout*; just-in-time identifiers	only this step's slice	latency; an agent may wander
Compress	compaction (*summarization*), *pruning*, tool-result trimming	same facts, fewer tokens	lossy, silently
Isolate	a subagent in *quarantine*	the mess stays sealed	tokens multiply, several-fold by common report

Chapter 7 · italics: the six-tactic vocabulary (Breunig, 2025) · weeks 5 and 6 go deeper

A hypothetical coding agent, a multi-hour migration

All four in one run

`small rules file at launch · **select**` → `search, open on demand · **select**` →

`progress file · **write**` → `drop applied file contents · **compress**` →

`compact near the soft limit · **compress**` → `dependency study to a subagent · **isolate**`

“Most tasks need only the two cheap moves—trim the tool results, write the plan to a file.”

Chapter 7 · the over-curation caution: “The trade is always recall against focus; there is no universal setting”

In class · teams of three · 20 minutes

Context rot, then budget a desk

1. Watch the explainer: Context rot and the four operations (about 4 min)
2. Open the budget planner with Chapter 2's desk and growth of 1,000 tokens per step
3. Window **200,000**, then **100,000** (hypothetical): utilization, steps to the 40% ceiling, steps to the hard limit
4. One operation per component, with a concrete move
5. One sentence: what did doubling the window buy, and what not?

instructions 500	history 8,000	two documents 40,000	tool result 3,000

Chapter 2's sketch: "made-up but realistic numbers" · 40% ceiling illustrative · after class, with Lab 3: your revised desk and what it could lose

Lab 3 and reading

Lab 3 · extend Lab 2

- A **skill folder**: name + description, a lean body, one reference, one script
- Loaded in levels; the script **run, never read**
- **Trim** tool results once acted on
- A **progress file** the agent writes
- Scripted test: 30 steps, history **stops growing**

Reading for week 5

- Ch. 8, Retrieval and Knowledge `full book`
- Ch. 9, Memory `full book`
- Ch. 10, Workflows and Composition Patterns `full book`

This week’s reading: Ch. 6 · Ch. 7 (full book) · no paid API needed for the lab

Five things to keep

- A skill is expertise in a folder, loaded in levels; the **description** decides whether it fires
- Ask **what kind of thing is missing**: reach, an action, or know-how
- Protocols turn M×N into M+N; every socket is a door
- A fuller desk is a duller model: **context rot** starts long before the window is full
- Name the failure, then **write, select, compress, isolate**

Engineering AI Agents · Week 4

Is everything in this context earning its keep?

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/

