

Claude: 12 Things I Learned the Hard Way Using Claude

Practical lessons on context, tools, tokens, agents, and memory



Prompts Need Context, Not Personas

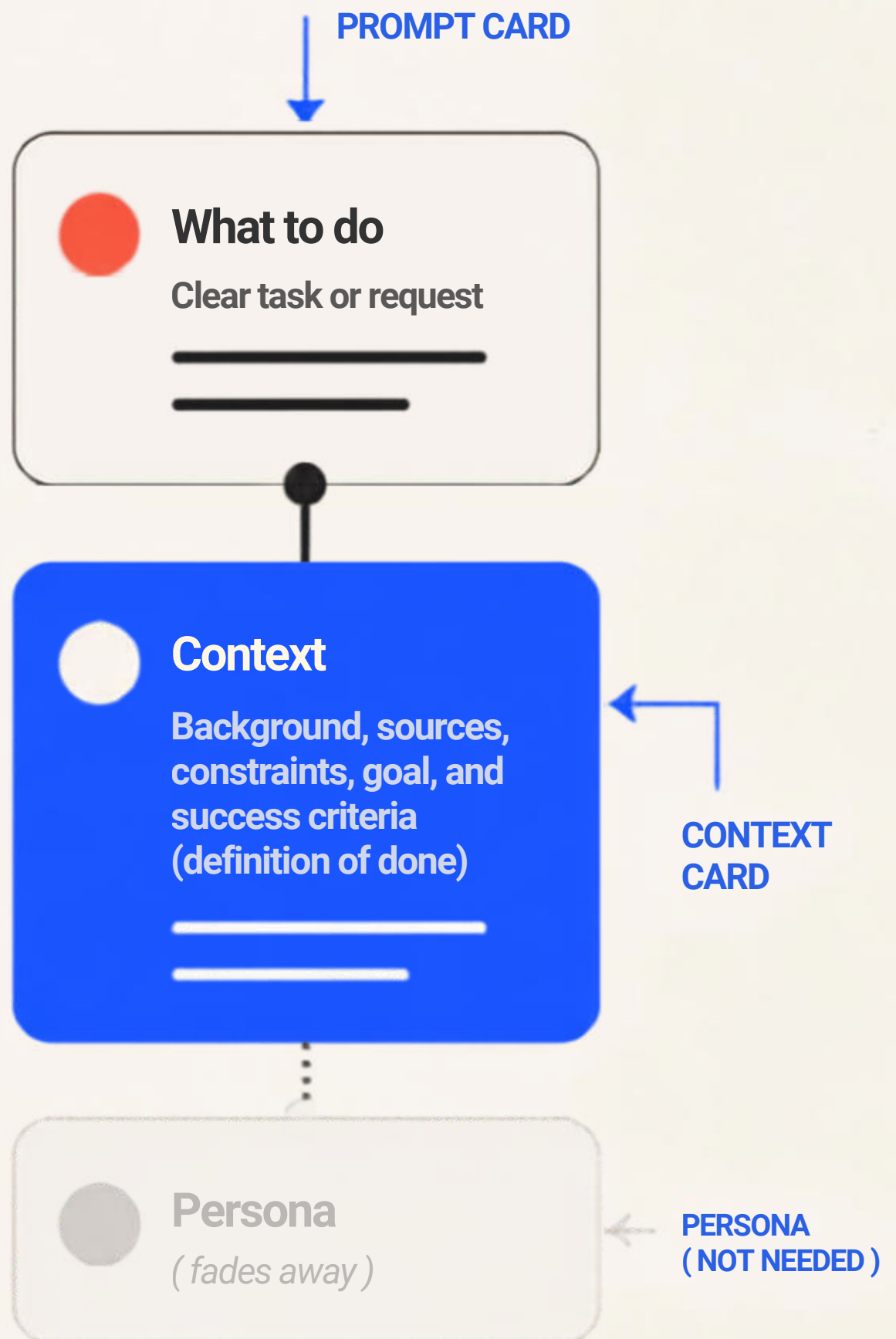
1. Adding personas doesn't improve output.

Focus on context, definition of done, and a self-check.

2. Avoid 'do not do X' instructions.

Tell Claude what to do instead: 'Write a brief paragraph.'

02



Tools Have a Hidden Cost

03

3.

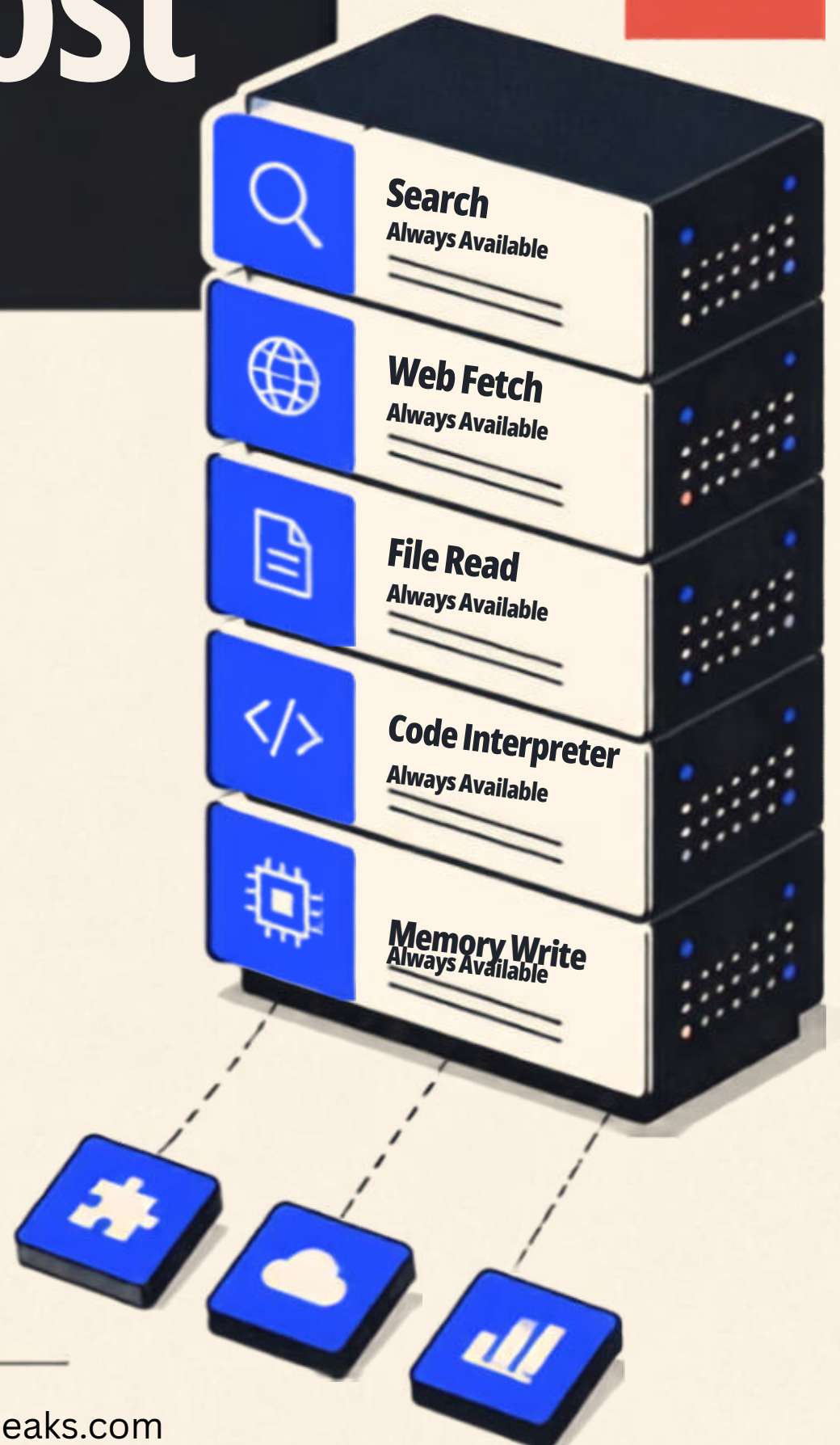
Routines load with full tool access.

Remove unnecessary connectors before running them.

4.

Tool definitions load upfront when access is always available.

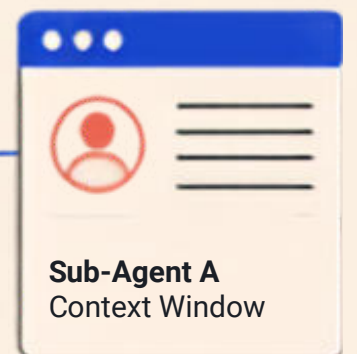
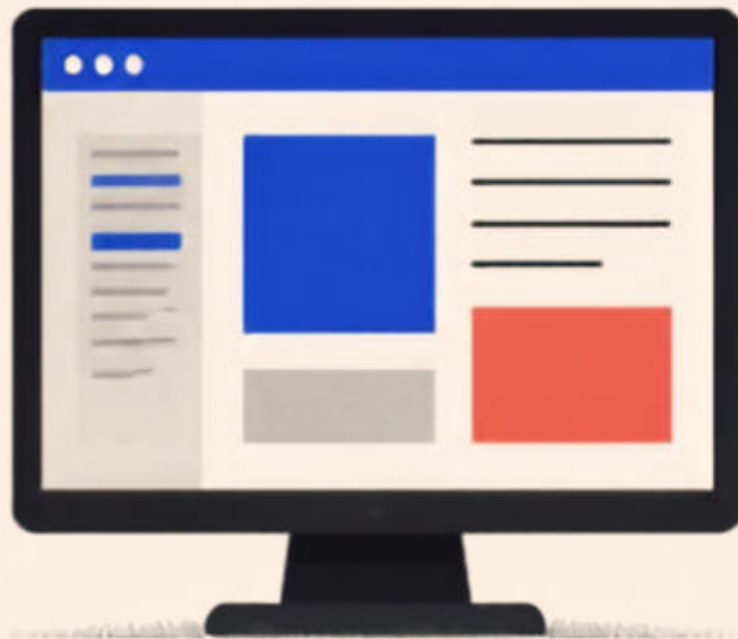
Be specific about the tools Claude needs.



Access Doesn't Equal Reuse

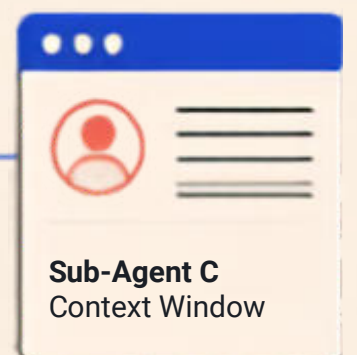
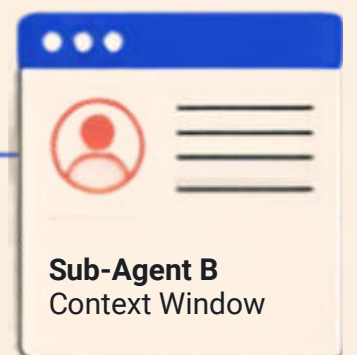
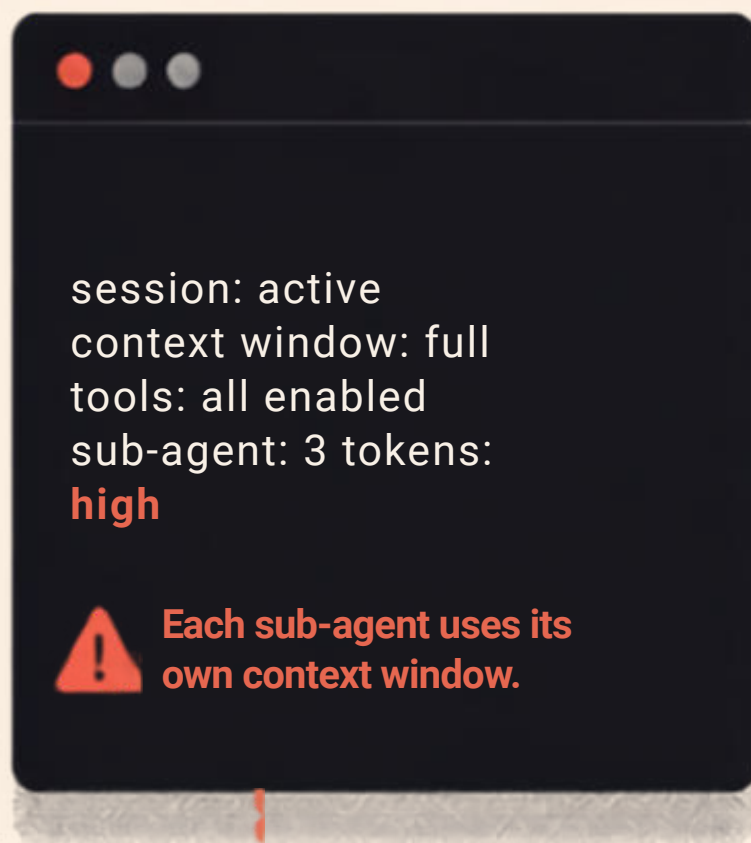
5. **Skills don't transfer** automatically between code and desktop environments.

Add them as IPs or recreate them where needed.



6. **SubAgents** can get expensive because each agent gets its own context window.

Use the main session for context-heavy tasks.



04

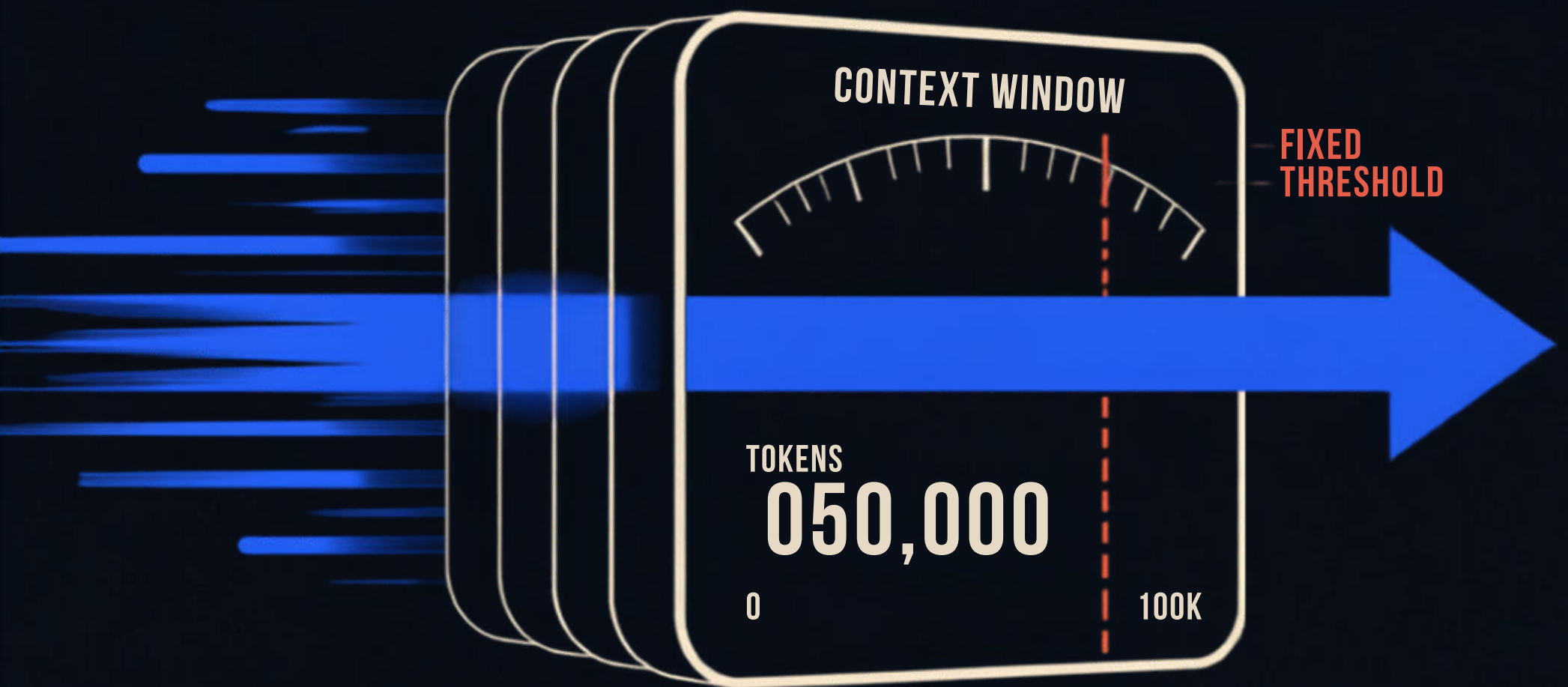


More context windows mean more tokens and higher costs.

Speed Can Burn Credits

05

LESSONS LEARNED
USING CLAUDE



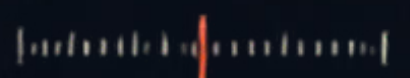
7.

Fast mode is expensive:
it runs on Opus, is **2.5x faster**,
and uses API credits.

8.

Set **auto-compact** to a fixed
token count such as **50k** or **100k**,
not a percentage. Keep tasks
within one context **window**.

TOKEN COUNT
050,000



FIXED THRESHOLD

Preserve Context

06



✓ 9.

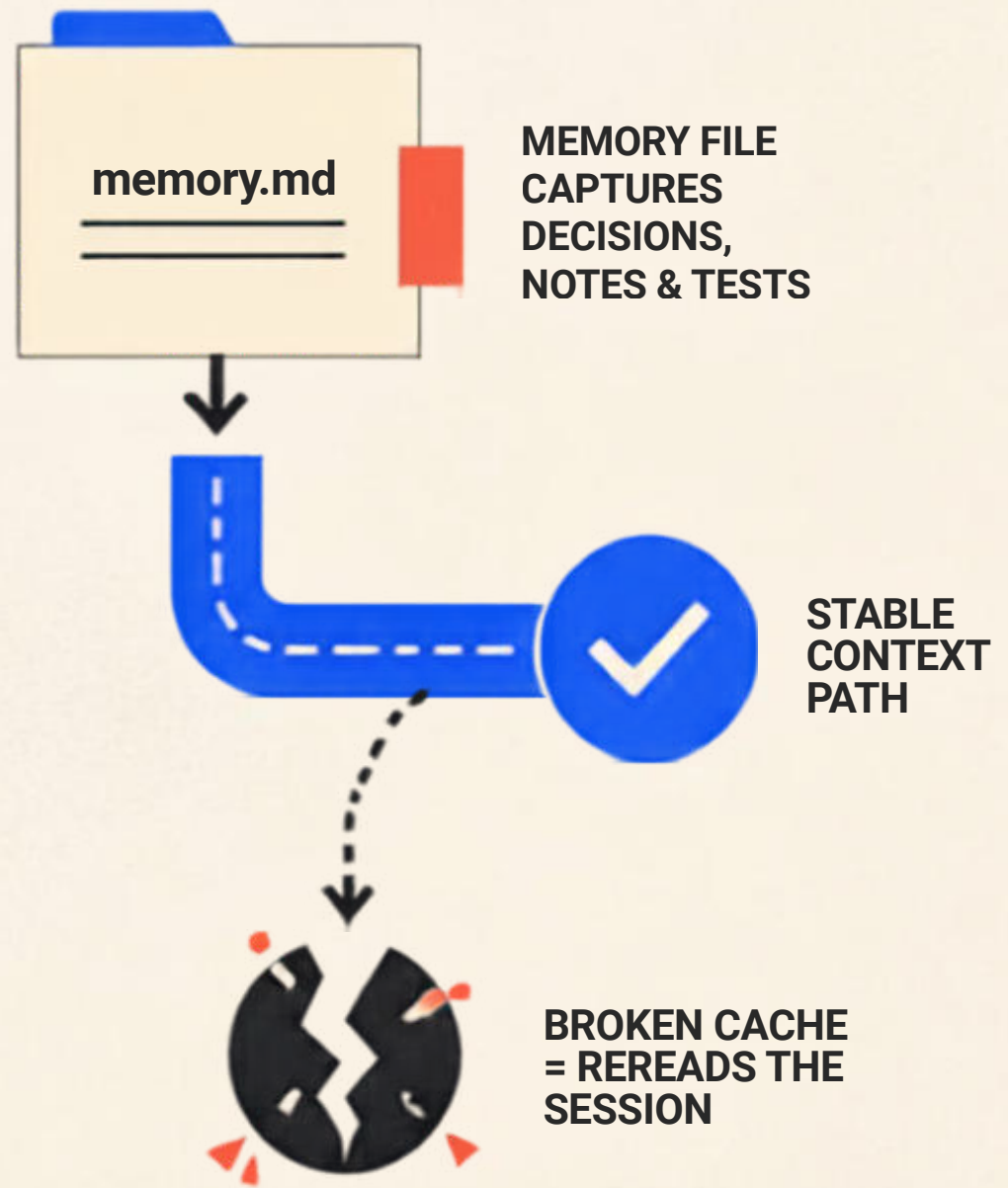
Cheaper models can cost more when switching; breaks the cache and the new model rereads the session.

✓ 10.

Keep a memory.md file with decisions, implementation notes, and tests from each session. Saves rework.



Bonus:
keep project insights and session learnings easy to retrieve.



Use memory deliberately.
Protect context.
Spend tokens where they matter.