



AGENTIC AI

Understanding the buzzwords

SYED MUHAMMAD AZHAR

Research Assistant
FDE(VIRKUM)



VIBE CODING

I describe the app, an AI agent builds it

01

The usual way

- Frontend: codes every screen
- Backend & APIs: writes every endpoint
- Database: designs it and wires it up
- Bugs: hunts them down, one by one

02

What I did

- Goal: I say what I want, not how
- Plan: it works out the steps
- Implement: frontend, backend, APIs, database
- Iterate: tests, fixes, repeats until it works



WHAT IS AGENTIC AI?

Is it new?

Not really. It builds on 80 years of work.

1940s

The idea of artificial neural networks

Decades

Held back by hardware, training data and algorithms

2017

Google researchers publish the Transformer architecture

2022

ChatGPT launches and reaches a million users in five days

Today

Multimodal models (text, images, audio) from many companies





LLM VS AGENTIC AI

01

LLMS are reactive in Nature

1. I ask for code
2. It writes it
3. I copy it and run it
4. It fails
5. I paste the error back
6. And back to step 2, again and again

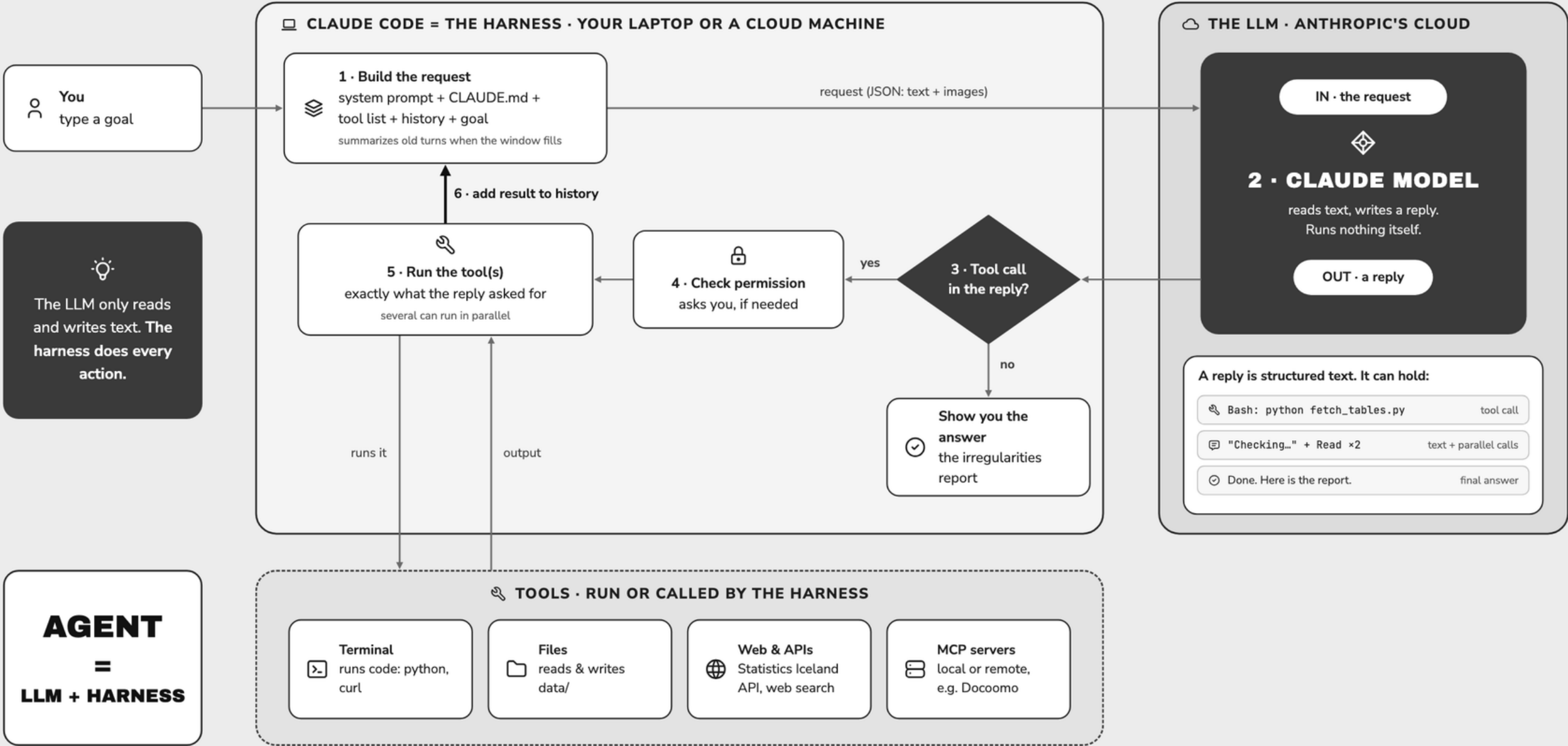
02

Agentic AI are proactive in Nature

1. Goal: what I want
 2. Plan the steps
 3. Act: write and run the code
 4. Observe the output and errors
 5. Fix it and go again
 6. Until the goal is met
- 

INSIDE AGENTIC AI

What happens when you press Enter in Claude Code



INSIDE A TOOL CALL

What Claude Code sends, what Claude sends back, and what runs on your laptop

1 · REQUEST

Claude Code → Claude

```
POST /v1/messages
{
  "system": "...prompt + CLAUDE.md",
  "tools": [{"name": "Bash"}, ...],
  "messages": [{
    "role": "user",
    "content": "How big is data/?"
  }]
}
```

Your question, sent with the system prompt, CLAUDE.md and the tool list.

2 · REPLY

Claude → Claude Code

```
{
  "stop_reason": "tool_use",
  "content": [{
    "type": "tool_use",
    "id": "toolu_01A",
    "name": "Bash",
    "input": {
      "command": "du -sh data/"
    }
  }]
}
```

A tool call is just text. Claude runs nothing itself.

3 · RUN

Your terminal

```
Bash(du -sh data/)
Allow this command?
> Yes
```

```
$ du -sh data/
84M    data/
```

Claude Code asks you first, then runs exactly that command.

4 · RESULT Claude Code → Claude, in the next request. The id matches the call.

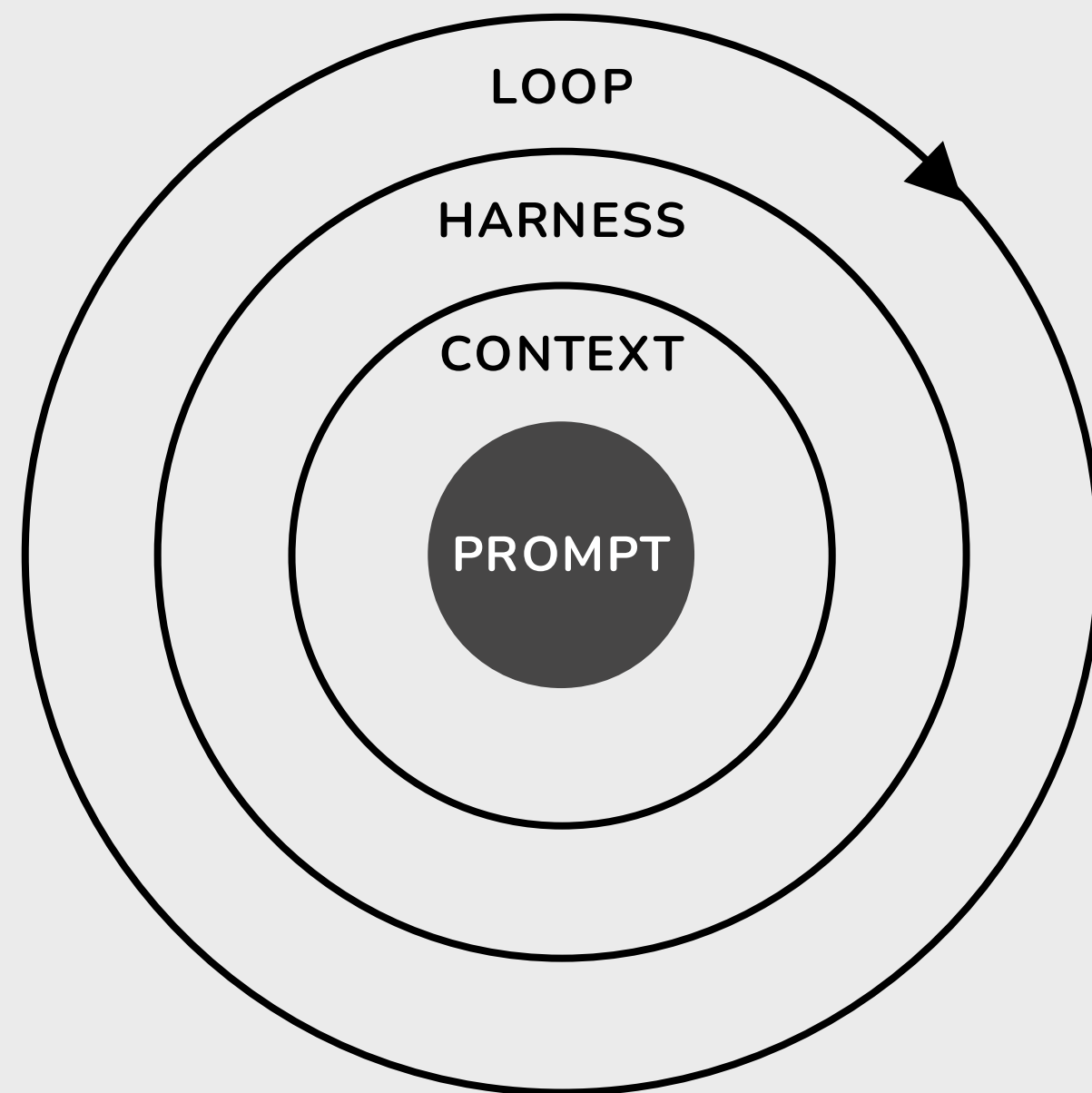
```
{"type": "tool_result", "tool_use_id": "toolu_01A", "content": "84M\tdata/"}
```

5 · ANSWER

The data/ folder is 84 MB.

PROMPT TO LOOP

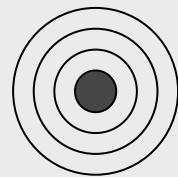
Four layers you design, test and improve



- 01 Prompt engineering**
How do I ask? Design the instructions and examples, then test them on many inputs
- 02 Context engineering**
What does it see? Choose the files, memory and history it gets, and what to leave out
- 03 Harness engineering**
What surrounds it? Build the software around it: tools, permissions, sandbox, tests
- 04 Loop engineering**
How does it keep going? Design the goal, the checks, the retries and when to stop

WHICH ONE DO I FIX?

When the agent gets it wrong, find the layer



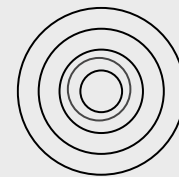
Prompt

SYMPTOM

It built something I didn't ask for

FIX

Say the goal, the limits and the result



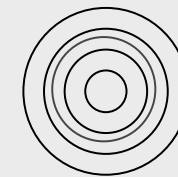
Context

SYMPTOM

It ignores our rules or forgets decisions

FIX

Write them in CLAUDE.md and cut the noise



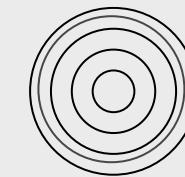
Harness

SYMPTOM

It does risky things or can't test its work

FIX

Set permissions, use a sandbox, give it tests



Loop

SYMPTOM

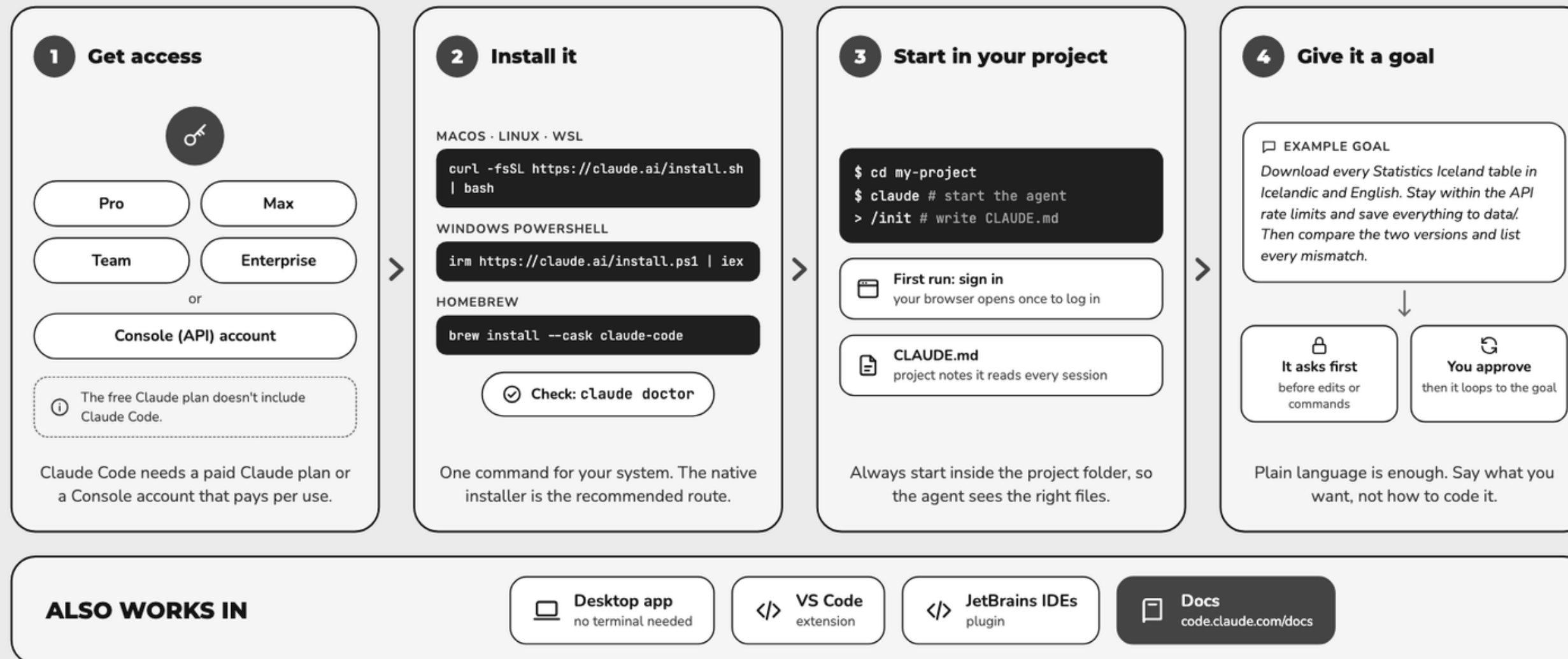
It stops too early, or never stops

FIX

Give it a check to pass and a limit on tries

AGENTIC AI SETUP

Claude Code, in four steps



Describe the goal. **Approve the steps.** Check the result.

Installation guide <https://code.claude.com/docs/en/quickstart>



THANK YOU