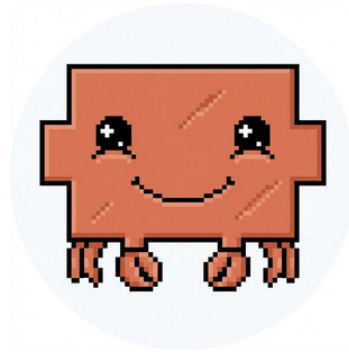


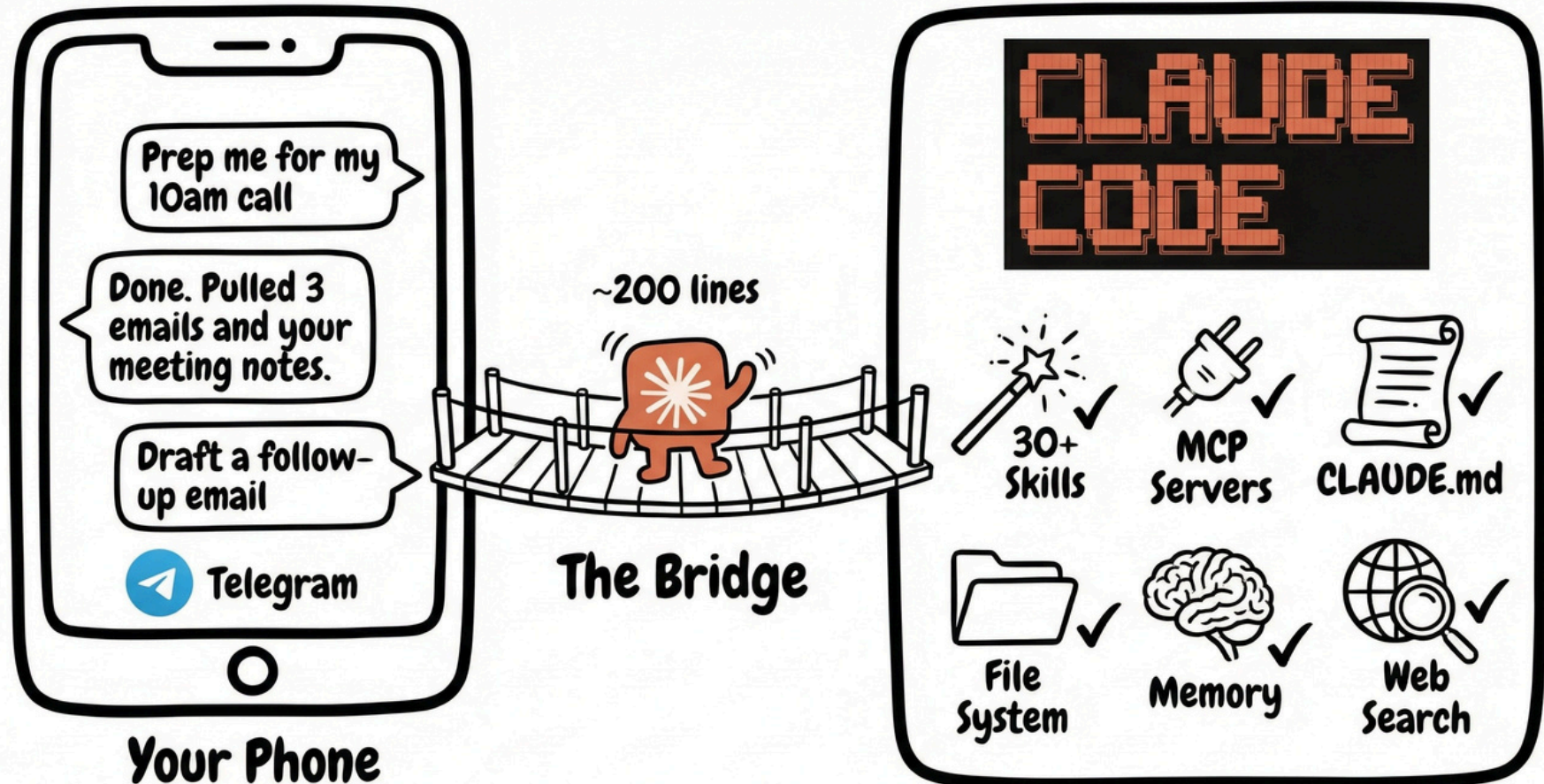


ClaudeClaw

Your ClaudeCode. InYour Pocket.

A visual guide to building your own AI personal assistant
using Claude Code's Agent SDK — no OpenClaw, no forks, no compromises.

Your Claude Code, In Your Pocket

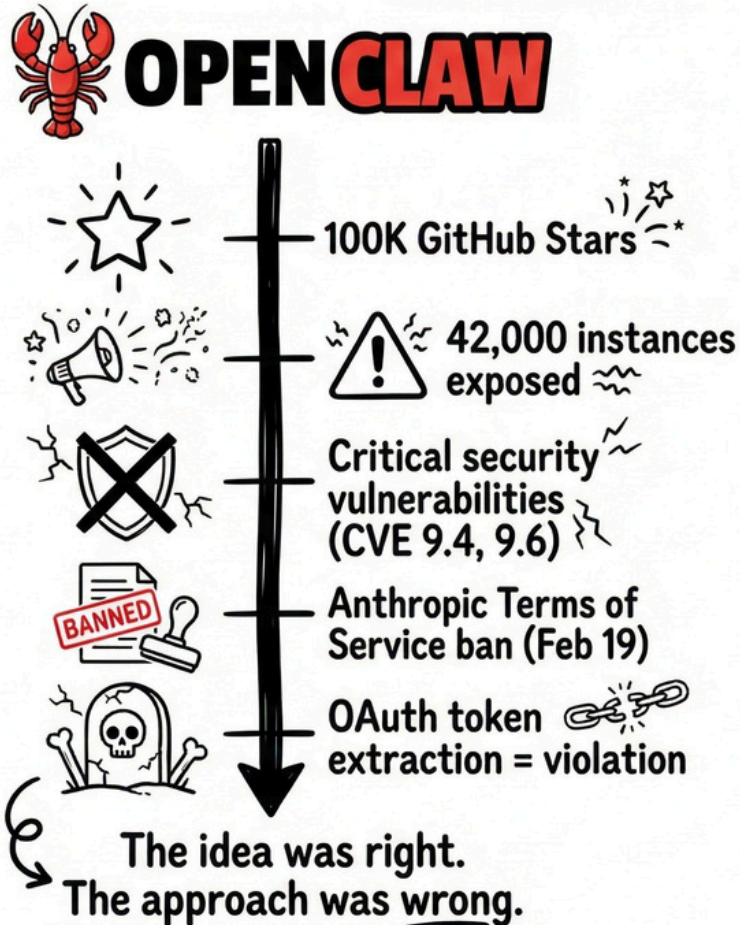


Everything you already use. One thin bridge. Access from your pocket.

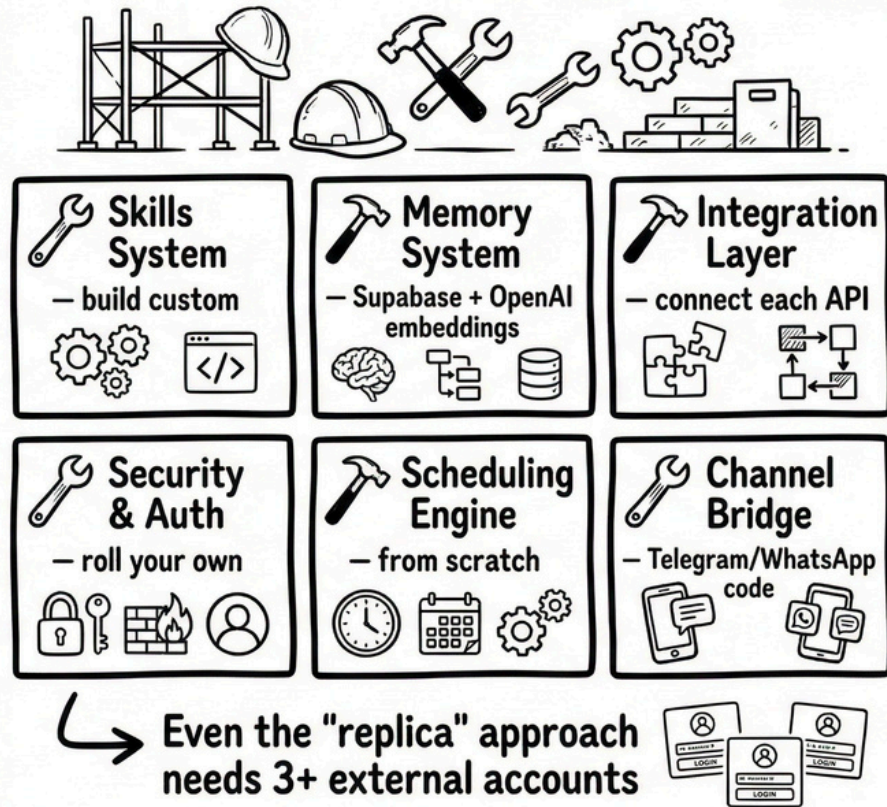
ClaudeClaw bridges your phone to your real Claude Code desktop setup. One thin connection (~200 lines) gives you access to 30+ skills, MCP servers, CLAUDE.md, memory, file system, and web search — all from Telegram.

The Problem With OpenClaw and Building From Scratch

What Happened to OpenClaw



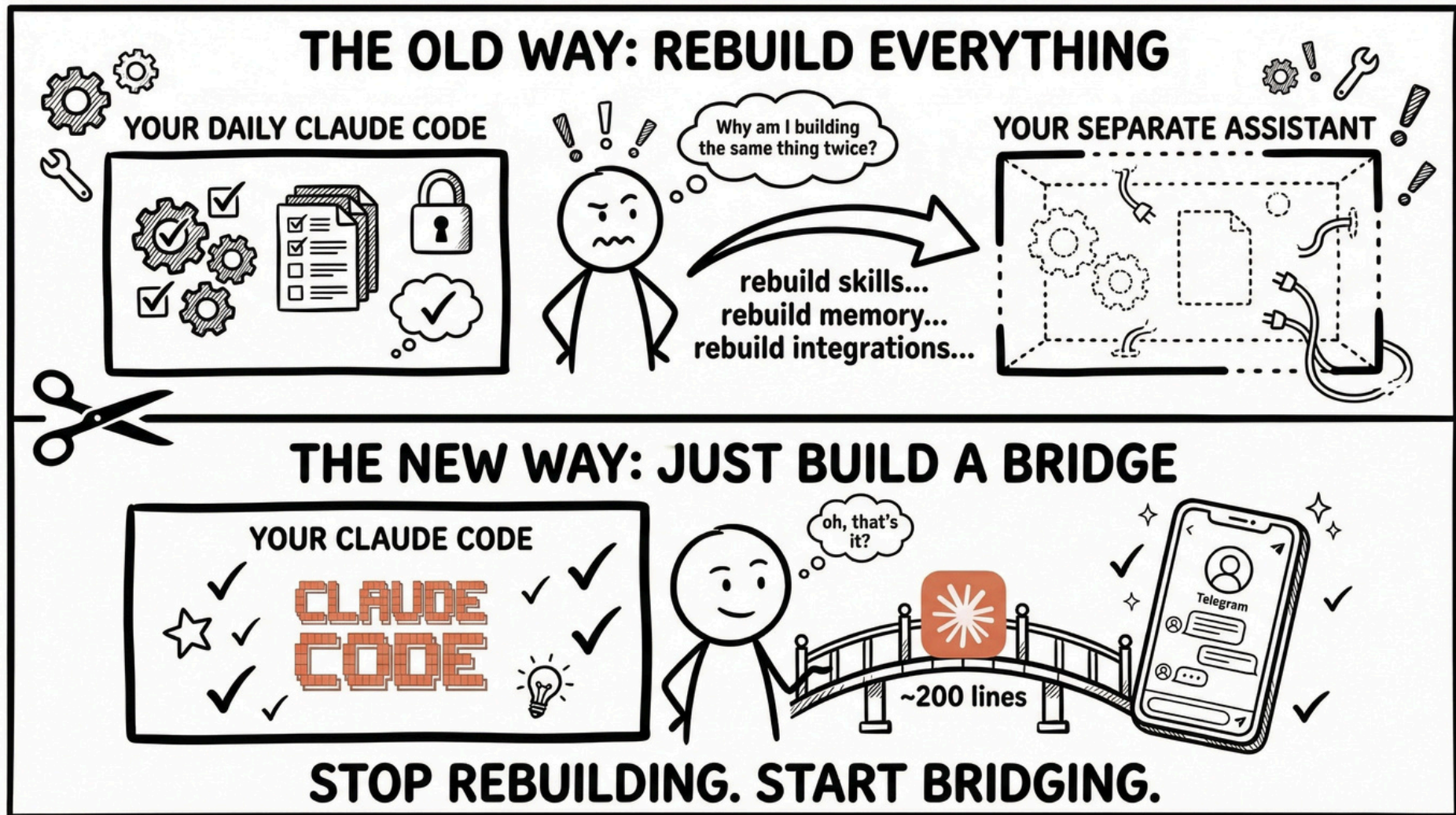
Building From Scratch = Weeks of Work



There has to be a simpler way.

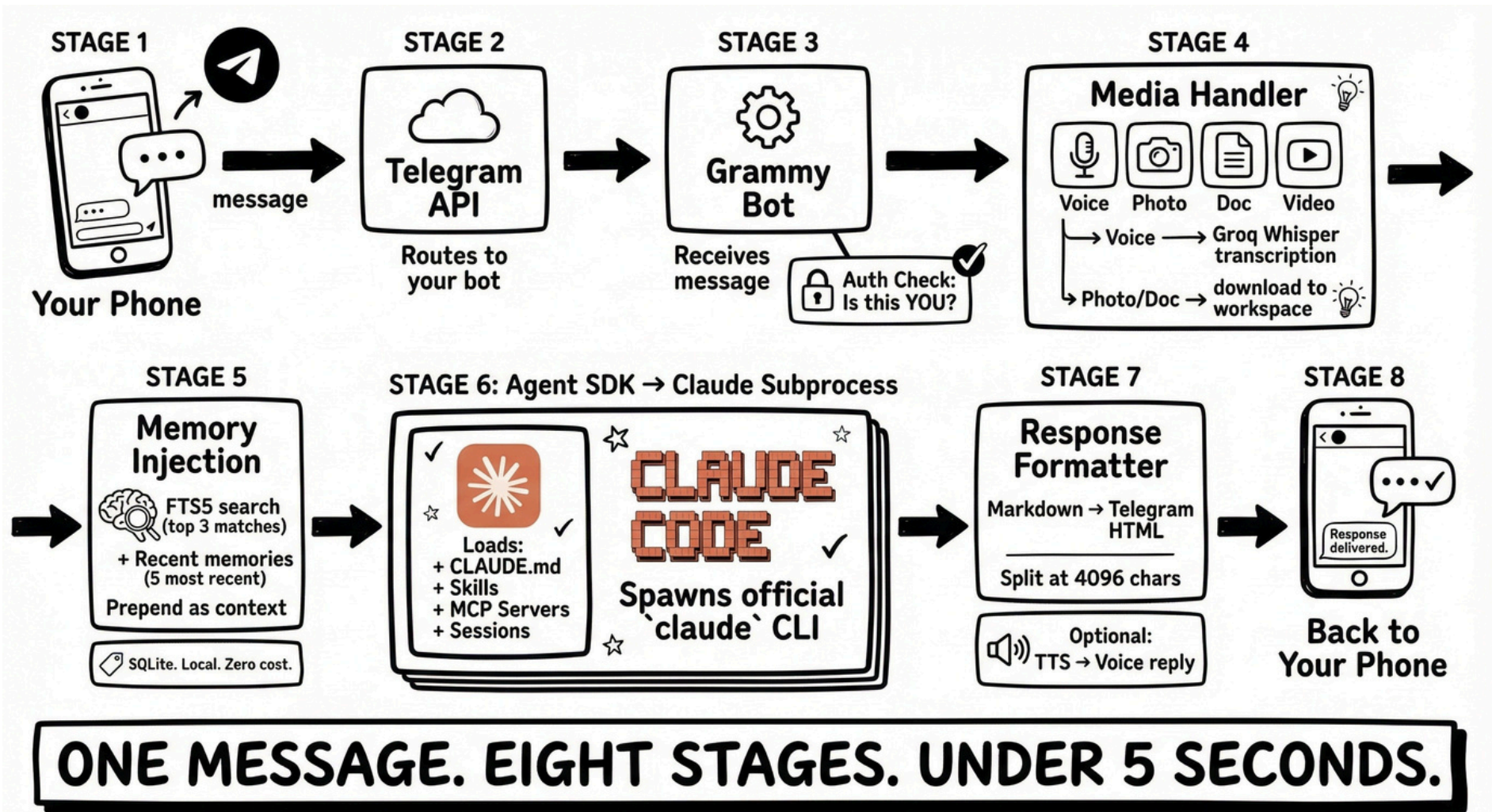
OpenClaw had the right idea but the wrong approach — extracting OAuth tokens violates Anthropic's Terms of Service. Building from scratch means weeks of work recreating skills, memory, integrations, security, and scheduling. There has to be a simpler way.

The Old Way vs The New Way



The old way: maintain two separate systems — your desktop Claude Code AND a separate assistant, duplicating everything. The new way: just build a bridge. Your existing Claude Code becomes your assistant. Stop rebuilding. Start bridging.

One Message. Eight Stages. Under 5 Seconds.



The full pipeline: You send a message on Telegram → API authentication → Grammy bot receives it → media handler processes voice/photos/video → memory injection pulls relevant context → Agent SDK spawns the official Claude CLI → response formatted → sent back to your phone. All in under 5 seconds.

What You Have + What You Add = What You Get

WHAT YOU ALREADY HAVE

**CLAUDE
CODE**

- ✓ 30+ Claude Code Skills
- ✓ MCP Servers
(Gmail, Calendar, Browse)
- ✓ CLAUDE.md
(your rules & personality)
- ✓ File system access
- ✓ Web search & scraping
- ✓ Session persistence
- ✓ Your business context

WHAT YOU ADD

A Telegram Bridge
Grammy + Agent SDK
~200 lines

+

=

WHAT YOU GET

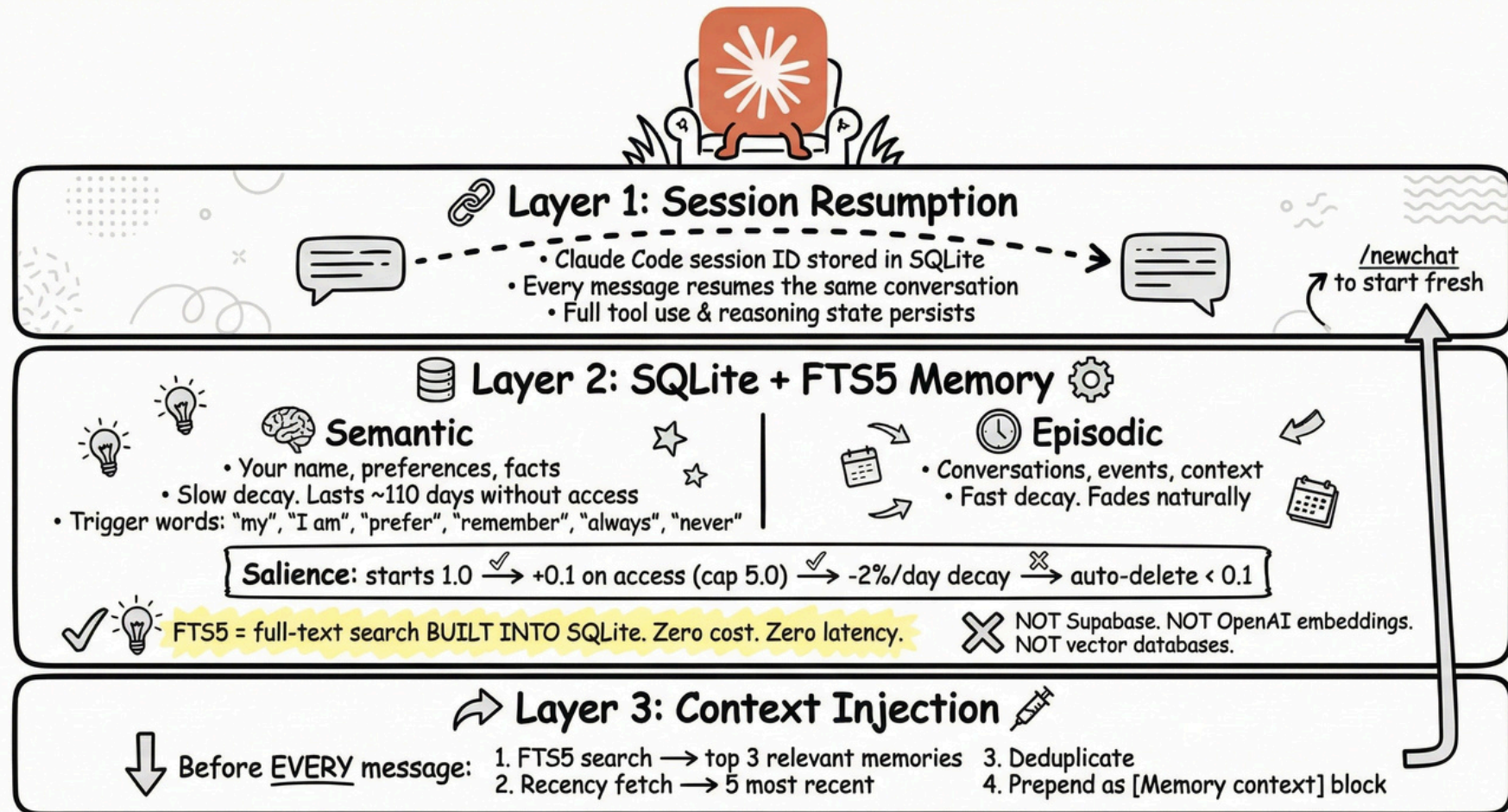
- ★ Personal AI assistant on your phone
- ★ Reads & sends your emails
- ★ Manages your calendar
- ★ Preps you for meetings
- ★ Remembers your conversations
- ★ Runs scheduled tasks
- ★ Uses ALL your existing tools

**From anywhere.
Any time.**

That's literally it.

You already have Claude Code with 30+ skills, MCP servers, CLAUDE.md, file system access, web search, and session persistence. Add a Telegram bridge (~200 lines via Agent SDK) and you get a personal AI assistant on your phone that reads emails, manages your calendar, runs scheduled tasks, and remembers your conversations.

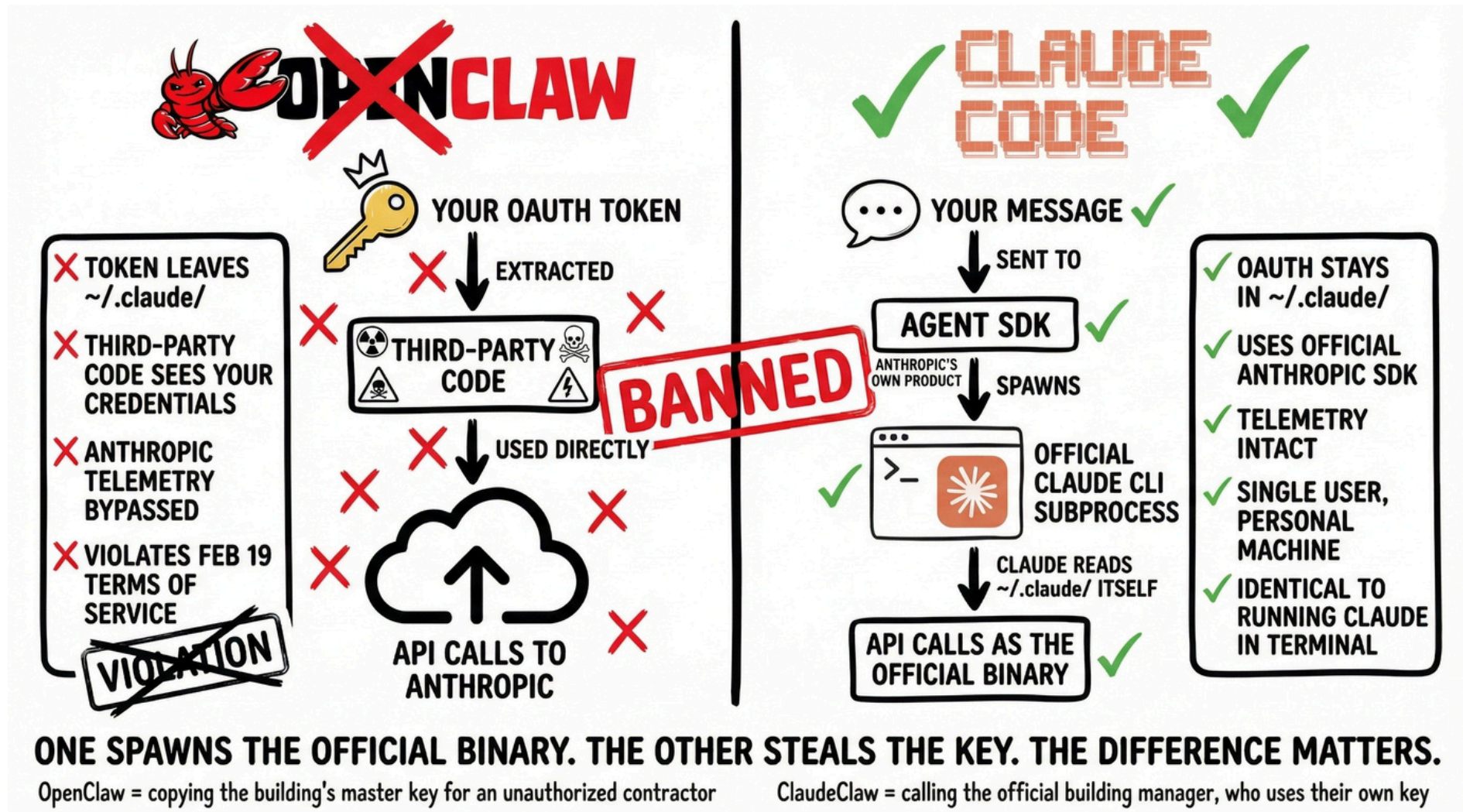
Three-Layer Memory System



Your assistant remembers you. Locally. For free.

Layer 1: Session resumption — Claude Code session IDs stored in SQLite so every message continues the same conversation. Layer 2: SQLite + FTS5 memory with semantic (long-term) and episodic (decaying) sectors. Layer 3: Context injection — before every message, the system searches memories and prepends the most relevant ones. Your assistant remembers you. Locally. For free.

Compliance: OpenClaw vs Claude Code



OpenClaw extracts your OAuth token and routes it through third-party code — a direct violation of Anthropic's Feb 19 Terms of Service update. ClaudeClaw spawns the official Claude CLI subprocess via Anthropic's own Agent SDK. Your token stays in ~/.claude/. One steals the key. The other calls the official building manager.

The Mega Prompt: 900 Lines That Build It For You

MEGA PROMPT
900 lines. Paste into Claude Code. **CLAUDE CODE**

4 Questions

Platform?



Telegram
paper plane



Discord
icon



iMessage
bubble

Voice?



Microphone
STT



Speaker
TTS



Both



None

Memory?



Brain Full
(search + decay)



List Simple
(last N turns)



None

Features?



Clock
Scheduler



Phone
WhatsApp



Video
Video analysis



Gear
Auto-start service

↓ Your answers shape the build

What Gets Built

```
src/
├── bot.ts
├── agent.ts
├── memory.ts
├── voice.ts ✓
├── scheduler.ts ✓
├── CLAUDE.md
├── package.json
└── .env configured
```

```
CLAUDE.md
package.json
.env </> configured
```

Only what you chose. Nothing extra.



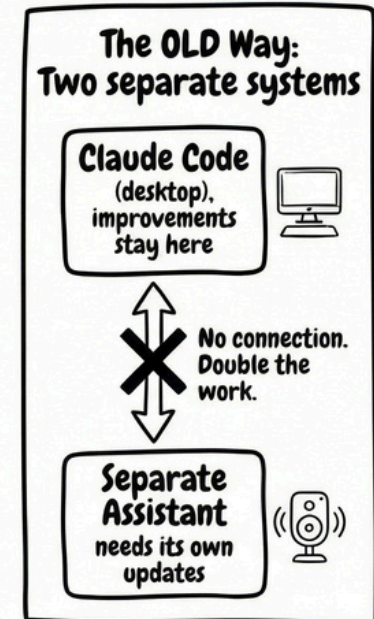
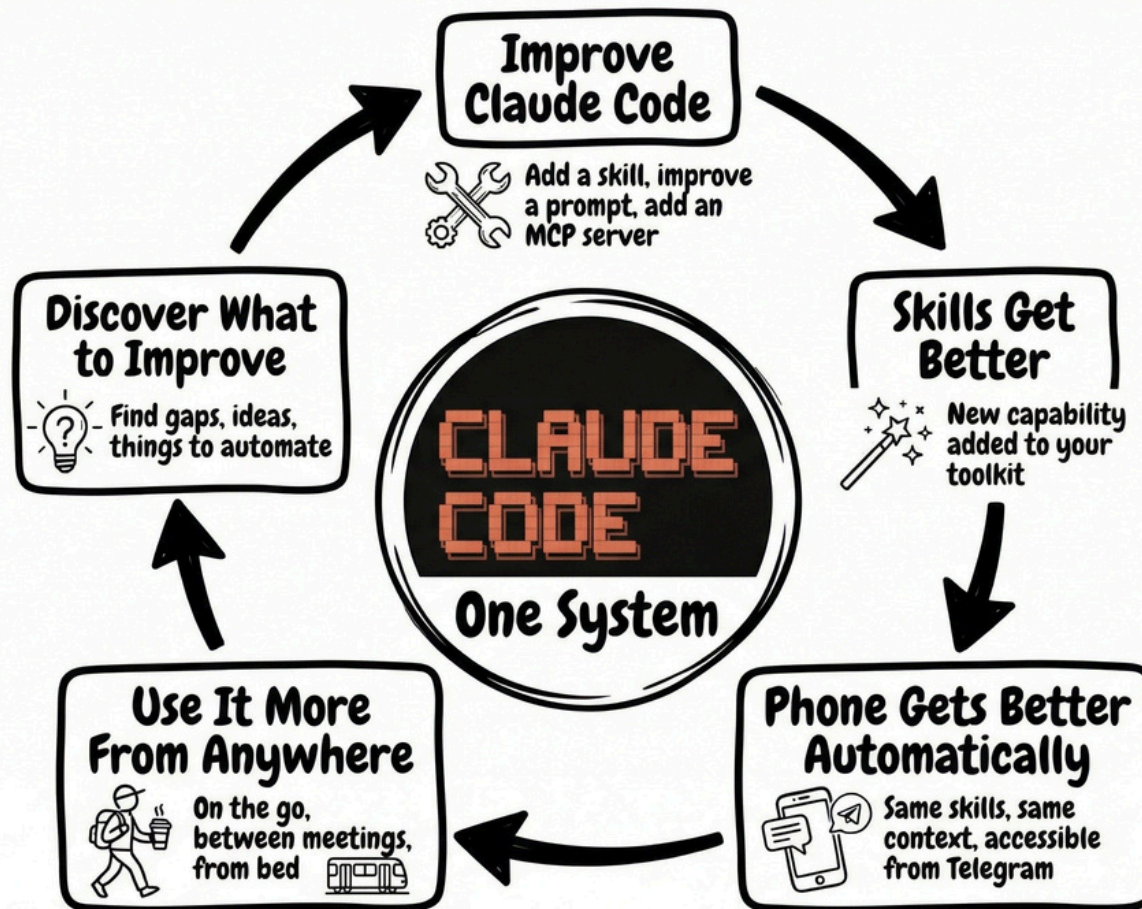
npm start → Bot alive in ~10 minutes

Fork it. Customize it. Make it yours.

Link in description. The entire thing is free.

Paste the mega prompt into Claude Code. It asks 4 questions: Platform (Telegram/Discord/iMessage), Voice (STT/TTS/none), Memory (full/simple/none), Features (scheduler/WhatsApp/video/auto-start). Your answers shape the build. Only what you chose gets built. Nothing extra. Fork it. Customize it. Make it yours.

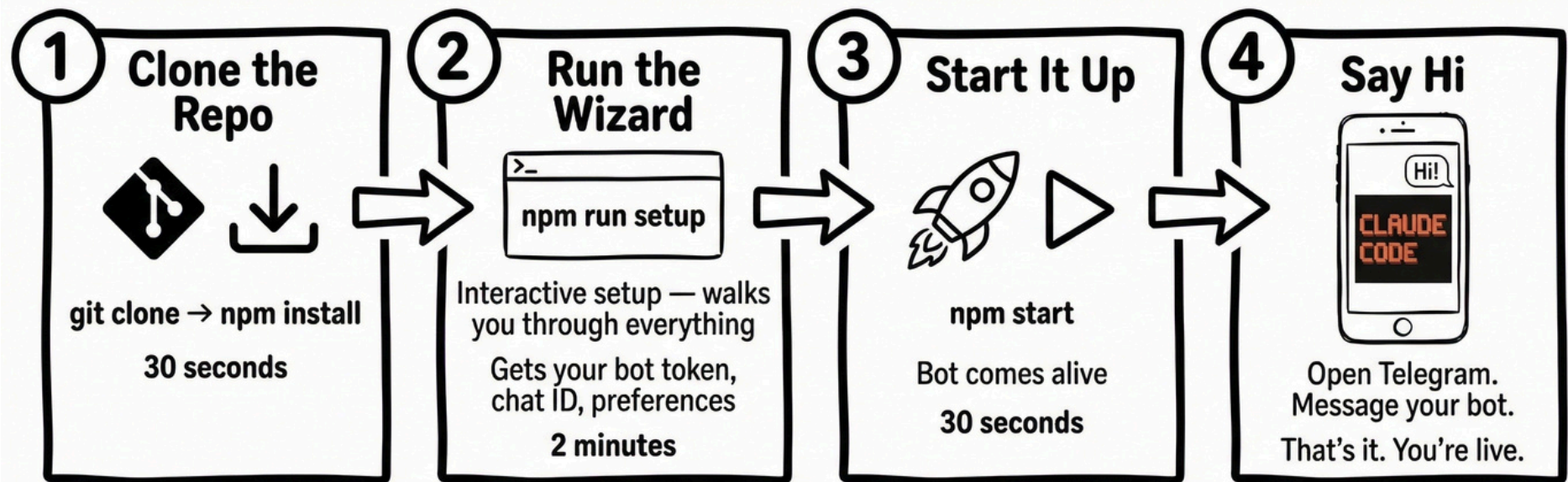
One System That Gets Better Every Day



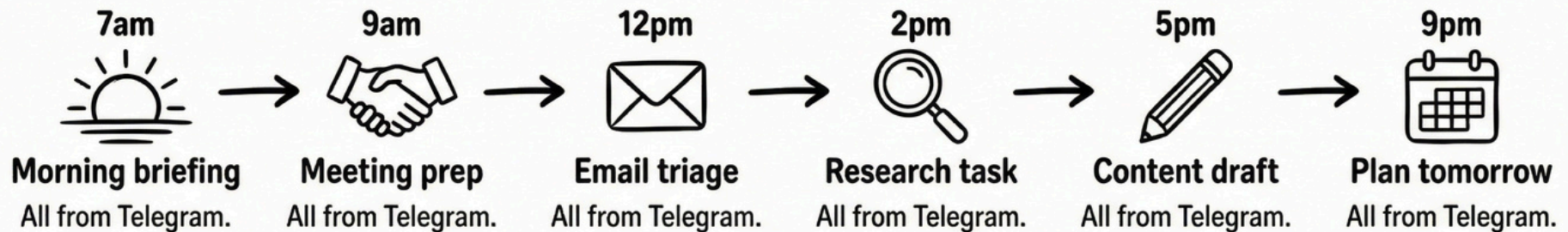
Aligned incentives. One tool. Gets better every day. Accessible everywhere.

Unlike maintaining two separate brains (desktop + assistant), ClaudeClaw is one unified system. Improve your Claude Code → skills get better → phone experience improves automatically → you use it more → you discover what to improve next. Aligned incentives. One tool. Gets better every day. Accessible everywhere.

Five Minutes of Setup. A Lifetime of Leverage.



What Your Day Looks Like Now

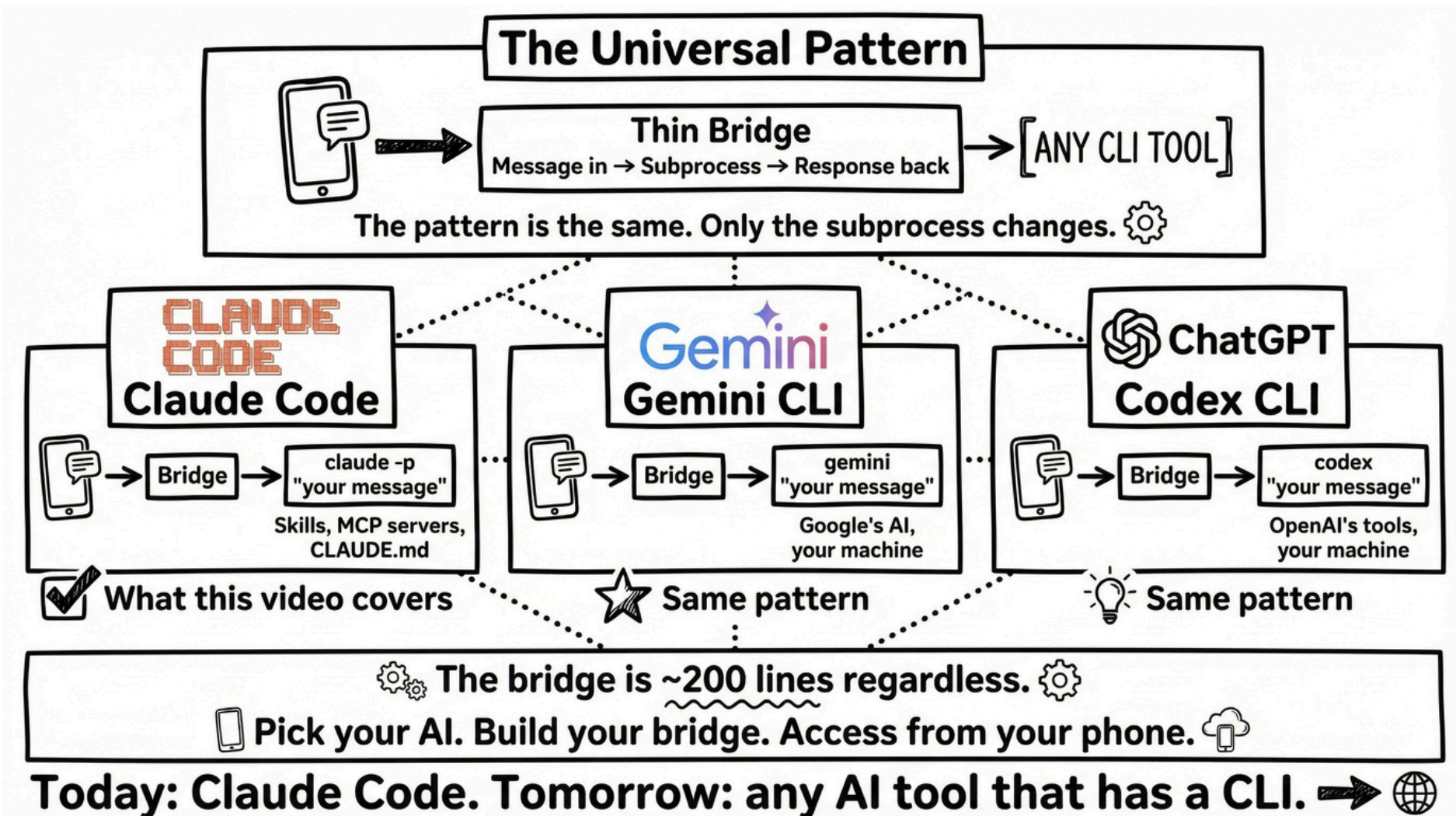


Five minutes of setup. A lifetime of leverage.

OR: Paste the 900-line mega prompt → Answer 4 questions → Same result, customized to you.

Step 1: Clone the repo (30 seconds). Step 2: Run the interactive wizard — it collects your bot token, chat ID, and preferences (2 minutes). Step 3: npm start — bot comes alive (30 seconds). Step 4: Open Telegram and say hi. That's it. Or paste the mega prompt and answer 4 questions for a fully customized version.

Beyond Claude Code: The Universal Pattern



The bridge pattern works with ANY CLI tool. Claude Code today, Gemini CLI tomorrow, Codex CLI the day after. The bridge is ~200 lines regardless. Pick your AI. Build your bridge. Access from your phone. Today: Claude Code. Tomorrow: any AI tool that has a CLI.