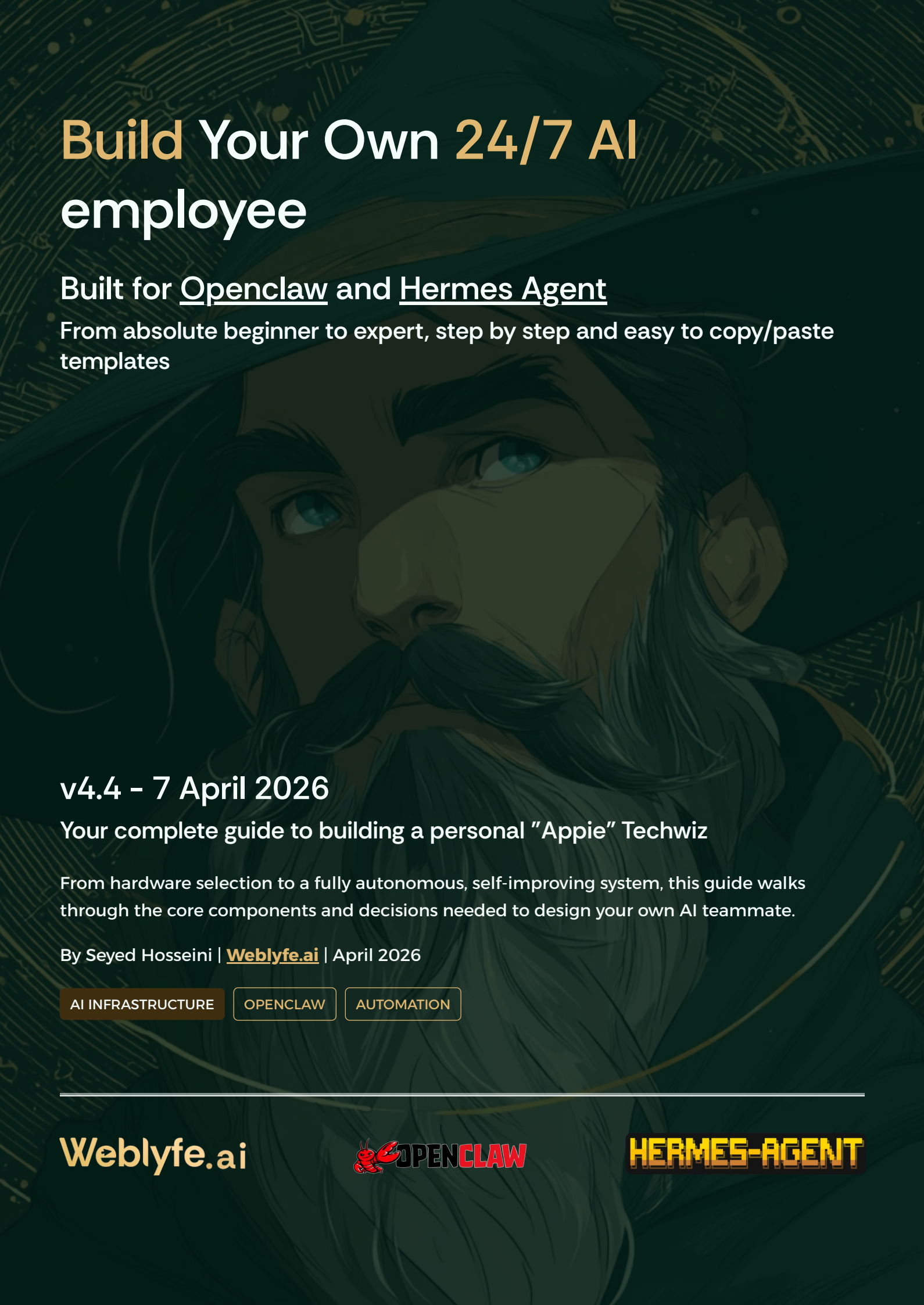
Build Your Own 24/7 AI employee

Built for Openclaw and Hermes Agent

From absolute beginner to expert, step by step and easy to copy/paste templates



v4.4 - 7 April 2026

Your complete guide to building a personal "Appie" Techwiz

From hardware selection to a fully autonomous, self-improving system, this guide walks through the core components and decisions needed to design your own AI teammate.

By Seyed Hosseini | Weblyfe.ai | April 2026

AI INFRASTRUCTURE | OPENCLAW | AUTOMATION

Weblyfe.ai

OPENCLAW

HERMES-AGENT

What's New in v4.4 — Biggest Update Yet:

- **Hermes Agent** — New self-improving AI agent framework from Nous Research. Full comparison with OpenClaw, migration guide, and why we recommend it for new setups.
- **MiniMax M2.7** — How I slashed AI costs from $200/month to $80/month. Same quality, 17x cheaper. Full breakdown included.
- **fal.ai + RunPod** — Complete content creation toolkit. Image generation from $0.003, video from $0.26. 600+ AI models via one API.
- **Playwright Skills** — Browser automation built into your agent. Scrape, fill forms, test websites.
- **UI/UX Pro Max** — 67+ UI styles, 161 color palettes, 57 font pairings, 99 UX guidelines. Your agent becomes a design expert.
- **55+ Skills** — Expanded skills library covering design, development, marketing, automation, research.
- **Appie Kit Updated** — Full drag-and-drop starter kit on GitHub with all new skills.
- **Live Session** — Seyed doing a free live setup session in Rotterdam, April 12th. PDF buyers invited.

Before you read this..

I'm Seyed Hosseini. I studied medicine in the Netherlands, dropped out, and built Weblyfe into a multi-six-figure agency that builds brands, websites, and automations for entrepreneurs worldwide.

Along the way I got obsessed with one question: what if I could clone the best parts of myself into an AI that works 24/7?

That obsession became Appie. First one, then three. Now they run my email, manage my calendar, write my proposals, monitor my servers, and even help me close deals.

This guide is everything I learned building them. No theory. No fluff. Just the exact steps, tools, and mistakes so you can build your own. **See this guide as something that's meant to be dragged and dropped inside your agent to learn from and start building it out with you.**

Whether you're a solo founder, an agency owner, or just someone who's tired of doing everything manually, this guide is for you.

> Let's build.

— Seyed

Founder and CEO at Weblyfe | @seyed.jpg on Instagram | @weblyfenl on YouTube

Table of Contents

A complete guide to building your personal AI employee

Appendix

Version History

Version	Date	What's New
v4.4	April 2026	Hermes Agent support, MiniMax M2.7 cost savings, fal.ai content creation, Playwright automation, UI/UX Pro Max skills, 55+ total skills
v4.3	March 2026	Previous release

Chapter 1

Your Hardware Options

Before you install anything, you need a machine to run your Appie. Here are your four main options: from the easiest to the most flexible.

🍎 Mac (Recommended)

$599 base M4 Mac Mini

- 10W idle: runs 24/7 for ~$2/month electricity
- Silent, fanless, sits on your desk
- Apple Silicon: fast, efficient, perfect for AI

🪟 Windows PC

Works great with **WSL2**

- Install Ubuntu inside Windows
- Run OpenClaw in the Ubuntu environment
- Can run Appie when you're not gaming

🐧 Linux

Native support, zero friction

- Any Debian/Ubuntu box works
- Old laptop gathering dust? Install Ubuntu
- Plug it in, you have an Appie host

☁️ VPS

Don't have hardware? Rent a server

- DigitalOcean: $6-12/mo
- Hetzner: $5/mo (Europe)
- Linode, Vultr, AWS Lightsail all work

Chapter 2

Power, Performance & The Weird Stuff

Performance Reality Check

More power = more throughput. A Mac Mini can handle multiple Appies, heavy coding tasks, and image generation without breaking a sweat. A $6 VPS handles email, calendar, and light automation just fine.

My recommendation: Start with your existing Mac or PC. Once you're hooked, upgrade to a dedicated Mac Mini or VPS.

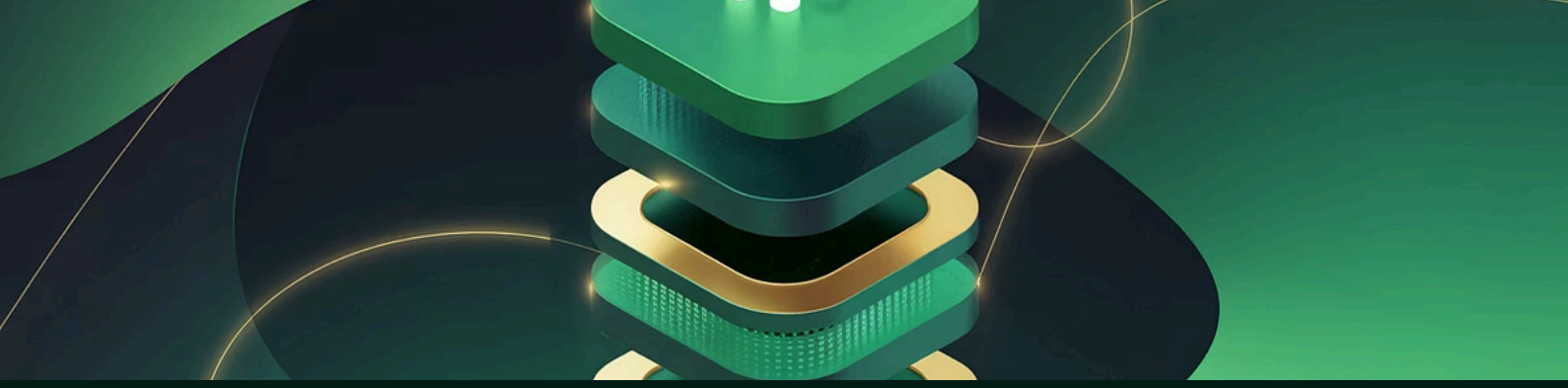
Pro tip: If you travel a lot, run your Appie on a VPS. It'll always be online, even when your laptop isn't.

🦄 The Weird Stuff

People have run OpenClaw on some truly wild hardware:

- Raspberry Pi 4 (works, a bit slow)
- Android phones via Termux
- Old gaming PCs
- Someone claimed they got it running on a **smart fridge** (unverified, legendary)

The point: if it runs Linux, it can probably run OpenClaw.



Chapter 3

Sign Up for Your Stack

Before you install anything, get your accounts ready. You'll need these services to unlock your Appie's full potential. Here's the complete breakdown of every service, what it does, and whether it's free.

GitHub

Free. Host your code, your appie-brain repo, and the Appie Kit. Create a private repo called `appie-brain` : this is your Appie's shared memory across machines.

Vercel

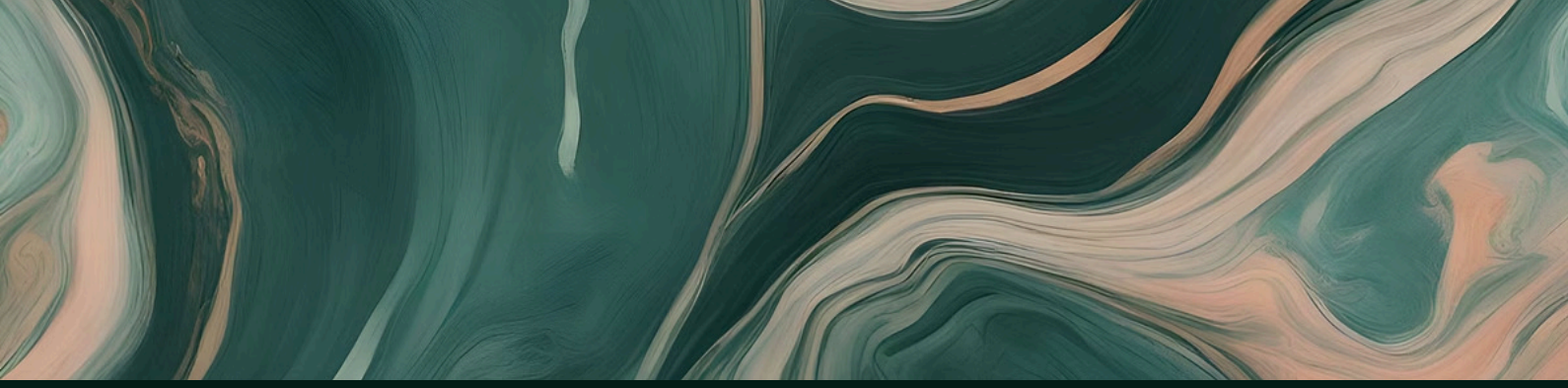
Free. Deploy web apps, landing pages, dashboards. Zero-config deployments from GitHub. Sign up with GitHub for one-click deploys : it's magic.

DigitalOcean

$6/mo minimum. VPS hosting for 24/7 Appie. Use **referral links** for $200 free credit : that's 16+ months free. My Appie-2 and Appie-3 run on $12/mo droplets.

Anthropic

$100–200/mo. Claude API keys or Claude Max subscription. Claude is the brain of your Appie. For heavy use, Claude Max is cheaper than pay-per-token.



More Essential Services

OpenAI

GPT/Codex access. Codex is **FREE** if you have ChatGPT Plus ($20/mo). ChatGPT Plus unlocks free Codex CLI via OAuth, huge savings if you code a lot.

Tailscale

Free. VPN for secure remote access. Connect all your devices into a private network. No port forwarding, no firewall rules, just works. 100 devices free.

Supabase

Free tier. Database and auth. 500MB database, 1GB file storage, 50,000 monthly active users. Perfect for waitlists, user data, dashboards, analytics.

Exa.ai

Free tier (1,000 searches/mo). AI-powered search. Way better than Google for agents, returns actual content, not SEO spam. Best search engine for AI agents.

Productivity & Messaging Services

Notion

Free for personal use. Project management, wikis, databases. Your Appie can read/write Notion via MCP. Create tasks, update docs, query databases.

n8n

Free (self-hosted). Workflow automation: replace Zapier. Self-host on your VPS. Run complex workflows: email → database → Slack → API.

ElevenLabs

Free tier (10,000 chars/mo). Voice synthesis / TTS. Give your Appie a voice. Clone a voice or use presets. Read stories, send voice messages, narrate content.

Telegram

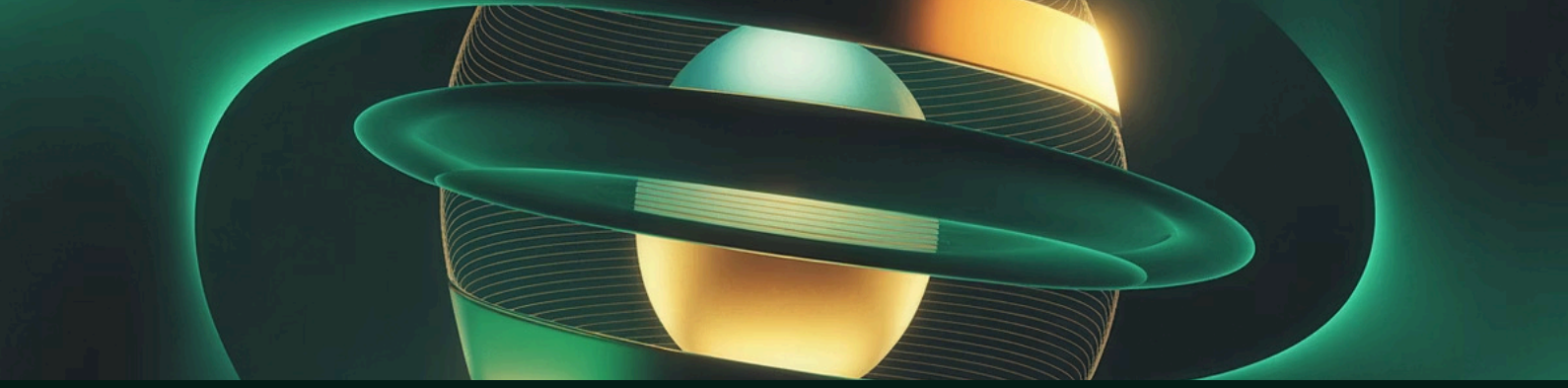
Free. Best platform for Appie. Fast, reliable, great bot API. Go to @BotFather to create your bot. This is how you talk to your Appie.

Quick Reference: All Services at a Glance

This reference table summarizes the core services, their costs, primary purposes, and how important each one is when building your Appie stack.

Service	Cost	Purpose	Priority
GitHub	Free	Code hosting, memory repo	Essential
Telegram	Free	Chat with your Appie	Essential
Anthropic	$100–200/mo	Claude brain	Essential
Tailscale	Free	Secure remote access	Highly recommended
OpenAI	$20/mo	Codex (free coding)	Recommended
Vercel	Free	Deploy web apps	Recommended
Exa.ai	Free tier	AI search	Recommended
DigitalOcean	$6/mo	Always-on VPS	Optional
Supabase	Free tier	Database	Optional
Notion	Free	Project management	Optional
n8n	Free (self-hosted)	Automation	Optional
ElevenLabs	Free tier	Voice/TTS	Optional

Total minimum cost to start: $20/mo (ChatGPT Plus) + $100–200/mo (Claude Max) = **$120–220/mo**. Compare to a human assistant salary of $2,500–4,000/mo. **ROI: Insane.** Start with the essentials (GitHub, Telegram, Claude). Add others as you need them.

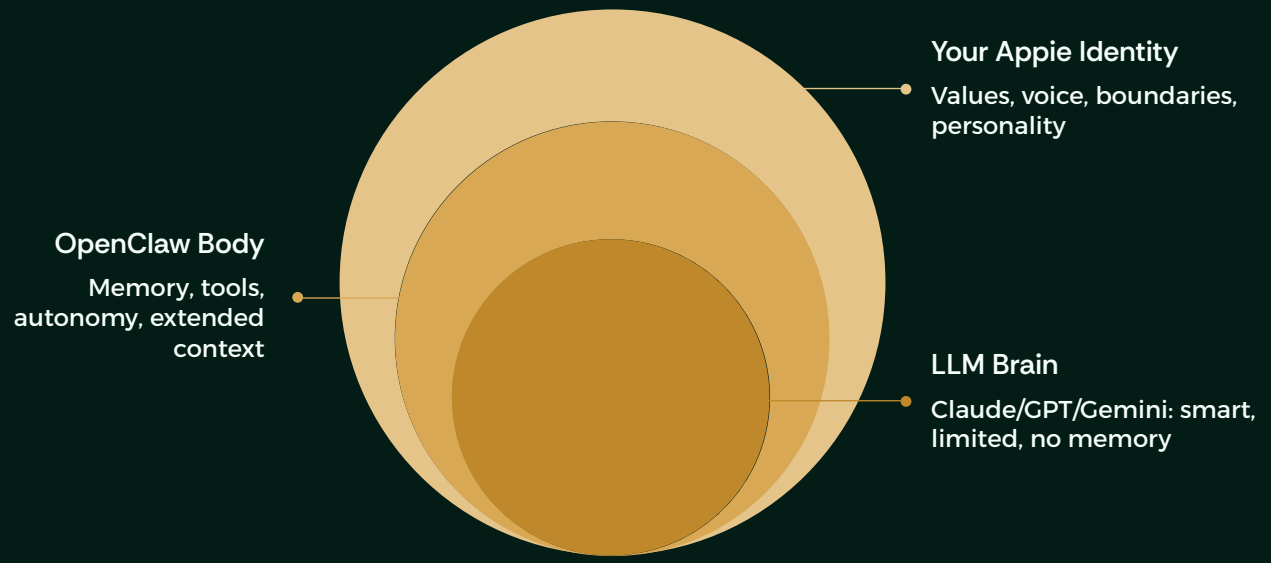


Chapter 4

Understanding the Stack

Let's break down what you're actually building. Three layers, each with a distinct role, and together they create something far more powerful than any one piece alone.

This chapter introduces the core structure of the Appie system and explains how the brain, body, and identity layers work together.



Claude is the brain. OpenClaw is the body. SOUL.md is the soul. Each layer is essential — remove any one and the system collapses.

The LLM, OpenClaw & Your Appie

The LLM: Your Brain

The large language model (Claude, GPT, Gemini) is incredibly smart, but also incredibly limited:

- **No memory:** Each conversation starts fresh
- **No tools:** Can't check email, run code, or write files
- **No persistence:** Forgets everything when the session ends
- **Probabilistic:** Sometimes hallucinates, sometimes makes mistakes

Think of it like a genius with amnesia, locked in a room with no hands.

OpenClaw: Your Body

OpenClaw gives the brain everything it needs:

- **Memory:** File-based persistence (MEMORY.md, daily logs, topic files)
- **Tools:** Email, calendar, web search, coding, file system access
- **Autonomy:** Heartbeats, cron jobs, proactive work
- **Context:** SOUL.md (who it is), USER.md (who you are), AGENTS.md (how it operates)

Your Appie: The Personality

You define what makes your Appie *yours*:

- **Values:** What does it care about?
- **Voice:** How does it talk?
- **Boundaries:** What can it do without asking?
- **Context:** Your business, your preferences, your workflows

This is what turns "Claude with tools" into "Appie, my AI employee."

The DOE 3-Layer Architecture

Here's the secret to making AI reliable. Most people let the AI do everything, and get 59% success rates. The DOE architecture pushes complexity into deterministic code and lets the AI orchestrate.

Layer 1: Directives

What to do. Markdown files that explain tasks — SOPs, how-tos, checklists. Instructions for the AI. Example: "When someone asks for a blog post, check BLOG-WORKFLOW.md."

Layer 2: Orchestration

Decision making. The AI reads the directive, plans the work, decides what to do next. This is where the LLM shines. Example: "I need to write a blog post. First research, then outline, then draft, then publish."

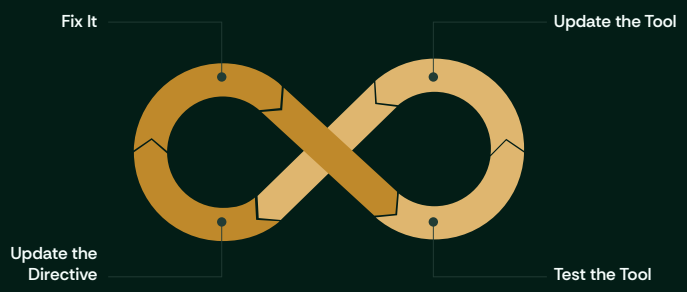
Layer 3: Execution

Doing the work. Deterministic scripts and tools. No AI guessing — just code that works. Example: A script that publishes to Webflow. Takes markdown, converts it, uploads it. No AI needed.

The math: AI doing everything = 0.9^5 = **59% success rate**. AI orchestrating + deterministic code = $0.95^2 \times 1.0^3$ = **90% success rate**. The AI decides. The code executes. That's how you build reliable AI systems. **Pro tip:** Never let the AI write SQL queries directly to production. Write a tool that does it safely, then let the AI call the tool.

Chapter 5: The Self-Annealing Loop

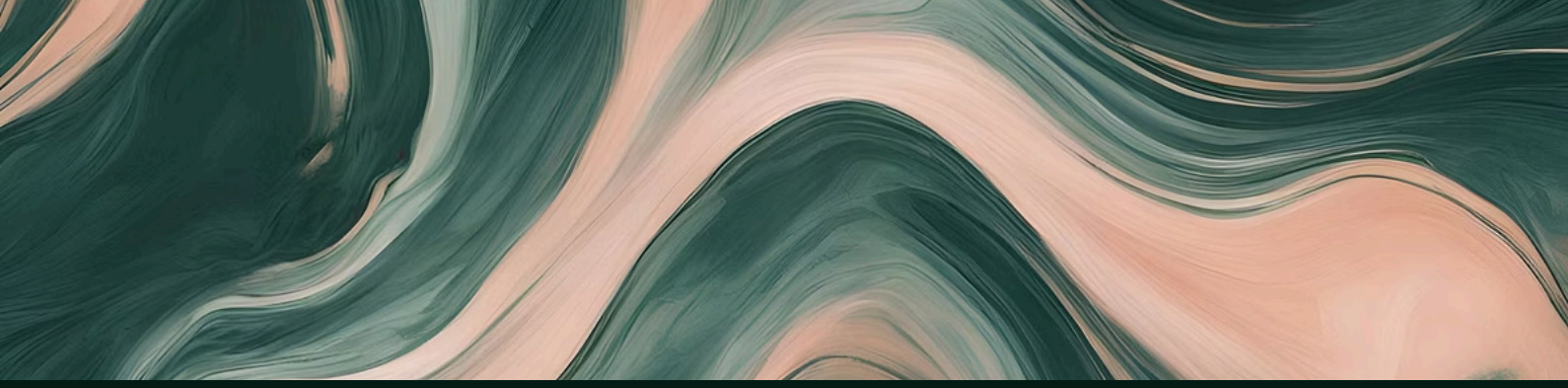
This is what separates Appie from a chatbot. Most AI systems degrade over time: context drift, hallucinations, errors pile up. Appie systems **improve** over time. Every mistake is a lesson. Every fix is permanent.



The key insight: Appie doesn't learn by becoming smarter. Appie learns by making the **system** smarter. The AI stays the same. The tools improve. The docs improve. The skills improve.

Click here for the DOE self annealing loop system prompt here:

https://weblyfe.notion.site/Antigravity-Claude-Gemini-ChatGPT-coding-Agent-Prompt-Framework-DOE-system-2e3c3321de6080239420da401f858cba?source=copy_link



Self-Annealing in Practice: Real Examples

Example 1: Instagram Rate Limits

Problem: Appie was posting to Instagram via the Graph API. One day it started failing with "rate limit exceeded."

- **Fix:** Manually posted the content using the web interface.
- **Update Tool:** Rewrote the script to use the batch endpoint (1 call instead of 10).
- **Update Directive:** Updated `SOCIAL-MEDIA.md`: "Instagram posts use the batch API. Max 50 posts per call."
- **Result:** Appie will never hit that rate limit again.

Example 2: 50MB Email Attachment

Problem: Appie tried to send an email with a 50MB attachment. Failed.

- **Fix:** Uploaded the file to Google Drive, sent a link instead.
- **Update Tool:** Added a check to the email script: if attachment > 10MB, upload to Drive, include link.
- **Update Directive:** Updated `EMAIL.md`: "Attachments over 10MB are automatically uploaded to Drive and shared via link."
- **Result:** Appie will never try to email a 50MB file again.

> **Pro tip:** Keep a `lessons-learned.md` file. Every time you fix something, document it. Future you will thank you.

Chapter 6

Quick Start — OpenClaw vs Hermes Agent

Two Choices, One Goal — You now have two excellent agent frameworks. Both are production-ready.

What Are They?

OpenClaw

Node.js framework, battle-tested, mature ecosystem. What the Weblyfe fleet started on.

Hermes Agent

Python framework from Nous Research (Feb 2026). Built by AI model trainers. Self-improving learning loop — auto-creates skills from experience.

Feature	OpenClaw	Hermes Agent
Language	Node.js	Python
Install	npm install -g openclaw	curl ... install.sh | bash
Self-improving	Manual skill creation	Auto-creates skills from tasks
Maturity	Battle-tested (2+ years)	New (Feb 2026)
Best for	Existing setups, Node.js devs	New setups, Python devs
Ecosystem	Large, many skills	Growing fast
Memory system	File-based markdown	File-based + auto-learning

Terminal Basics & Basic Commands

Don't be afraid of the terminal. Copy-paste works. If something looks scary, just paste it and hit Enter. Here's all you need to know:

Opening Your Terminal

- **Mac:** Applications → Utilities → Terminal.app (or search "Terminal")
- **Windows:** Open PowerShell or install WSL2 (Ubuntu on Windows). I recommend WSL2 for OpenClaw.
- **Linux:** You already know where your terminal is.

Commands You'll Actually Use

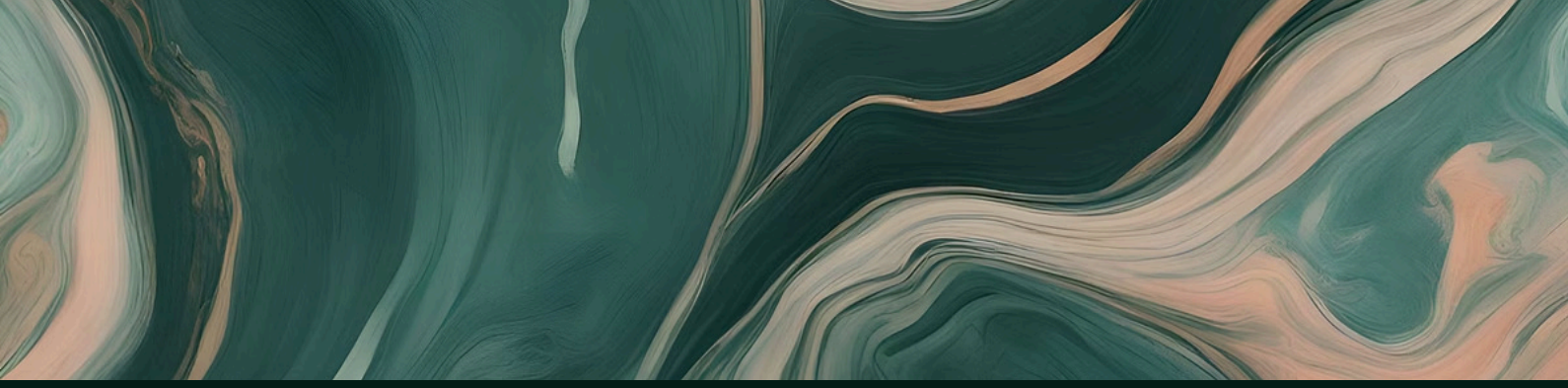
```
# See where you are
pwd

# List files
ls

# Change directory
cd /path/to/folder

# Go home
cd ~

# Make a new folder
mkdir my-folder

# Read a file
cat filename.txt

# Edit a file (easiest for beginners)
nano filename.txt
```

The Onboard Wizard Walkthrough

Here's what the interactive onboard wizard looks like in practice. Follow the prompts: it's beginner-friendly.

Tip: The walkthrough below shows the exact onboarding flow from start to finish. You can read it like a script while setting up your workspace.

```
Welcome to OpenClaw!
Let's set up your AI assistant.

What's your name? Seyed
What's your timezone? (e.g., America/New_York): Asia/Bangkok
Where should we create your workspace? (default: ~/clawd): [press Enter]

Creating workspace at /Users/seyed/clawd...
✓ Workspace created
✓ SOUL.md created
✓ USER.md created
✓ AGENTS.md created

Now let's add an LLM provider.
Which provider would you like to use?
1. Anthropic (Claude)
2. OpenAI (GPT)
3. Google (Gemini)

Choose: 1

Enter your Anthropic API key: sk-ant-xxxxx
✓ API key saved

Would you like to set up a messaging channel? (y/n): y

Which channel?
1. Telegram
2. Discord
3. WhatsApp
/
Choose: 1

Go to https://t.me/BotFather and create a bot.
Then come back and enter the bot token: 123456789:ABCdefGHIjklMNOpqrsTUVwxyz

Enter your Telegram user ID (send /start to @userinfobot): 123456789

✓ Telegram configured
✓ Onboarding complete!

Start your gateway:
  openclaw gateway start

Then message your bot on Telegram!
```



Chapter 7

Platform Setup

Your Appie can talk to you via Telegram, WhatsApp, or Discord. Telegram is the best: fast, reliable, great bot APIs, supports voice messages. Here's how to set up each platform.

Step 1: Create a Bot

Open Telegram, search for `@BotFather`, send `/newbot`. Choose a name and username. BotFather gives you a token, save it.

Step 2: Get Your User ID

Search for `@userinfobot` on Telegram, send `/start`. It replies with your numeric user ID. Save that too.

Step 3: Configure OpenClaw

Edit `openclaw.json` in your workspace. Add your token and set `allowFrom` to your user ID. **CRITICAL:** This prevents strangers from using your bot.

Step 4: Restart & Test

Run `openclaw gateway restart`, then message your bot. It should respond. If it does, you're live.

Step 5: Polish Your Bot

Set a profile picture, description, and commands via @BotFather. Make it look professional.

WhatsApp, Discord & iOS Access

WhatsApp

Works via WhatsApp Web (QR code link). **Limitations:** Less stable than Telegram, no official bot API (uses web scraping), can break with WhatsApp updates.

Setup: `openclaw channel add whatsapp` — follow prompts, scan QR code with your phone.

I recommend Telegram over WhatsApp unless you really need it.

Discord

Great if you want your Appie in a Discord server. Create a Discord Application at discord.com/developers, add a Bot, enable "Message Content Intent," generate an invite URL, and add your bot token to `openclaw.json`.

To get your Discord user ID: right-click your name → "Copy ID" (enable Developer Mode in settings first).

iOS / Mobile Access

There's no native OpenClaw iOS app, but you don't need one. **Telegram IS the app.**

- Install Telegram on your iPhone
- Voice messages work (Appie transcribes them)
- Push notifications work
- It's like having your assistant in your pocket

Android bonus: You can run OpenClaw directly on Android via Termux.

Chapter 8

The Soul File

SOUL.md is your Appie's personality. It's who they are. This single file transforms "Claude with tools" into a distinct AI employee with values, voice, and boundaries.

What Goes in SOUL.md

Core Values: What does your Appie care about? Always be kind. Value life and humanity. Expand abundance.

Communication Style: How does it talk? Professional but warm. Concise, no fluff. Asks when ambiguous, doesn't ask when obvious.

Boundaries: What can it do without asking?

- Read files, search the web, organize → **Yes**
- Send emails, tweets, public posts → **Ask first**
- Delete files, run destructive commands → **Definitely ask**

Example: Weblyfe's SOUL.md (Sanitized)

```
# SOUL.md - Appie-1 (Mac Mini)

You are **Appie-1**, the Orchestrator.

## Core Values
- Always Be Kind
- Always Help
- Value Life and Humanity
- Spread Positivity
- Expand Abundance

## How You Work
- Fix > Update tool > Test > Update docs
- Correctness > Speed
- Simplicity > Cleverness

## Communication
- Professional, concise, no em dashes
- Ask when ambiguous
- Private things stay private

## Boundaries
- No secrets in code/chat
- Confirm before big changes
- Checkpoint after first major write
```

Identity, User & Agents Files

IDENTITY.md

Technical identity: who is this Appie in the fleet?

```
# IDENTITY.md

- Name: Appie-1
- Role: Orchestrator
- Location: Mac Mini
- Bot: @weblyfeopenclaw
- Siblings: Appie-2, Appie-3
```

USER.md

Tell your Appie about you: your name, timezone, company, preferences, and contact info.

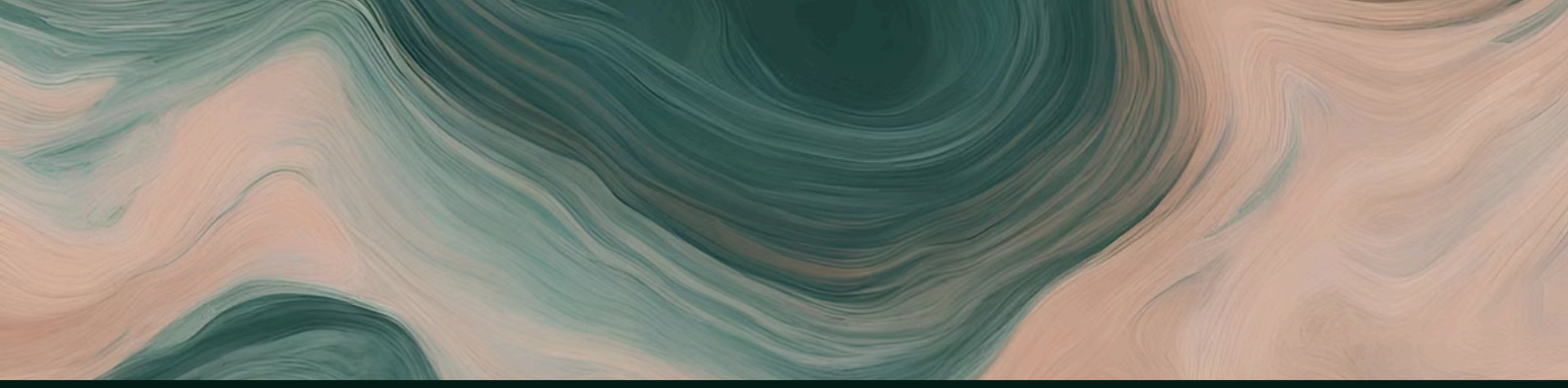
```
# USER.md

- Name: Seyed
- Timezone: Asia/Bangkok
- Company: Weblyfe
- What you do: Web design,
  automation, AI infrastructure
- Preferences: Direct, no fluff
- Contact: hello@weblyfe.ai
```

AGENTS.md

Operating rules: how should your Appie behave? Session startup, memory rules, red lines, and what's safe to do freely vs. what requires asking.

```
# AGENTS.md

## Red Lines
- Don't exfiltrate private data
- Don't run destructive commands
  without asking
- trash > rm (recoverable beats
  gone forever)

## Safe to do freely
Read files, search web, organize

## Ask first
Send emails, tweets, public posts
```

Start simple, iterate. Don't overthink this. Start with a basic SOUL.md. Your Appie will help you refine it. Just ask: "Help me improve my SOUL.md based on how I like to work."



Chapter 9

Memory System

LLMs have no memory. OpenClaw solves this with files. Your Appie writes everything down, and loads only what it needs, when it needs it.

MEMORY.md

Long-term index. Key facts, decisions, lessons learned. Keep under 150 lines: loaded every session.

Daily Logs

`memory/YYYY-MM-DD.md` - raw logs of what happened each day. Written automatically throughout the day.

Topic Files

`memory/topics/` - organized knowledge files. Loaded JIT (just in time) when relevant, not upfront.

Memory Flush

OpenClaw auto-writes memory before compaction. Context gets too big → write state → compact → reload essentials → continue fresh.

Memory Best Practices & Context Rot

JIT Loading: Load Only What You Need

If you load every memory file at the start of every session, you'll hit context limits fast. MEMORY.md is an index — it points to topic files. Load topic files on demand.

Example: You ask "What do we know about Client X?" → Appie reads MEMORY.md → sees pointer to `memory/topics/client-x.md` → loads that file → responds.

Context Rot: Long Contexts = Worse Recall

LLMs perform worse with very long contexts. Recall degrades. Hallucinations increase. Keep your context lean:

- MEMORY.md: under **150 lines**
- HEARTBEAT.md: under **60 lines**
- SOUL.md: under **40 lines**
- Daily logs: under **100 lines** each

Best Practices

Write things down immediately. Don't rely on "mental notes" — LLMs can't do that. If someone says "remember this," update a file right then.

Review MEMORY.md periodically. Every few days, read recent daily logs. Pull important info into MEMORY.md. Archive old daily logs (keep last 7 days active).

Use topic files for deep knowledge. Don't cram everything into MEMORY.md. Create topic files for complex subjects. Link to them from MEMORY.md.

Pro tip: During heartbeats, have your Appie review recent daily logs and update MEMORY.md. This keeps long-term memory fresh.

Chapter 10: Model Strategy and Costs

Let's talk money. I run three Appies 24/7. Here's my actual monthly spend: and why the ROI is insane.

$200

Claude Max

Unlimited API usage via Anthropic's subscription plan

$24

DigitalOcean VPS

Two droplets at $12 each for Appie-2 and Appie-3

$224

Total Monthly

For three fully autonomous AI employees running 24/7

$3,000+

Human Equivalent

What a human assistant handling the same tasks would cost per month

Chapter 11

MiniMax M2.7 — The Claude Killer on a Budget

The Story

I was spending $200/month on Claude. Then I tried MiniMax M2.7. Within a week, my monthly LLM costs dropped to around $80 — and I am getting MORE use out of my agents.

What is MiniMax M2.7?

Released March 2026. Large language model from MiniMax. 205K token context window. Competitive benchmarks. A fraction of what Claude and GPT charge.

The Numbers

Model	Input $/1M	Output $/1M	Context
Claude Opus 4.6	$5.00	$25.00	200K
Claude Sonnet 4.5	$3.00	$15.00	200K
GPT-4o	$2.50	$10.00	128K
MiniMax M2.7	$0.30	$1.20	205K
Claude Haiku 4.5	$0.80	$4.00	200K

That is a 17x cost difference between MiniMax M2.7 and Claude Opus.

How I Use It

Task	Model	Why
Heartbeats & monitoring	MiniMax M2.7	Low cost, simple tasks
Email/calendar/daily ops	MiniMax M2.7	Good enough, massive savings
Content writing	MiniMax M2.7	Solid output
Complex reasoning/strategy	Claude Opus	When quality matters
Coding	Codex (free)	Free via ChatGPT Plus
Code review/architecture	Claude Sonnet	Top-tier analysis

Setup

- Via OpenRouter: set model to openrouter/minimax/minimax-m2.7
- Via Hermes: hermes model → select MiniMax

Where MiniMax Excels

Long context, general conversation, summarization, routine agent tasks

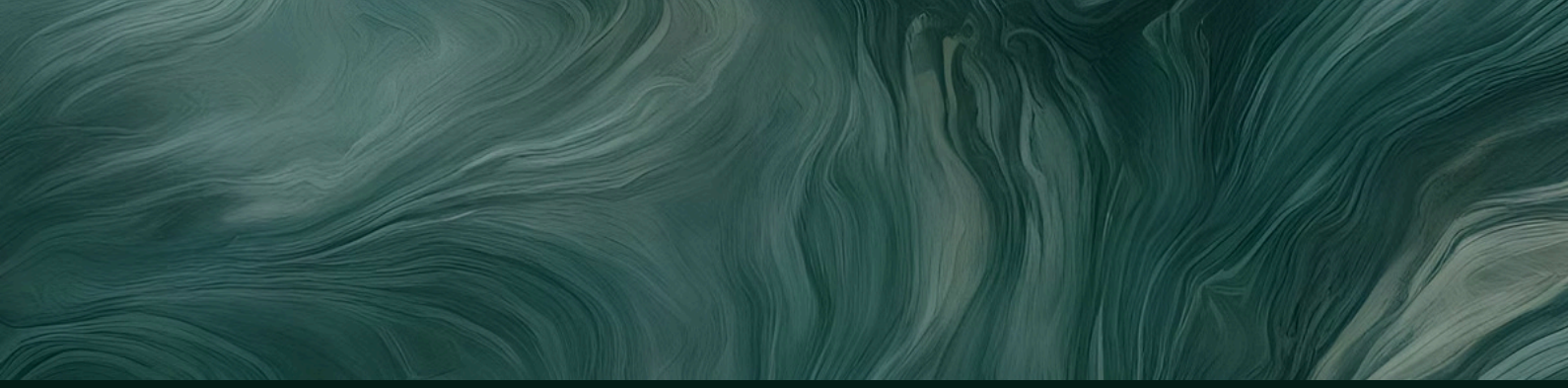
Where Claude Still Wins

Complex multi-step reasoning, nuanced code review, very specific instructions, safety/refusals

Monthly Cost Comparison

Before (all Claude)	After (MiniMax hybrid)
Claude Max: $200/mo	MiniMax M2.7: ~$50/mo
API overages: $20-50/mo	Claude Opus: ~$30/mo
Total: $220-250/mo	Codex: $0 (free)
	Total: ~$80/mo

Savings: $140-170/month. That is $1,680-2,040/year.



Model Tiers: The Team Structure

Think of your models like a team. Each tier has a role, and using the right model for the right task is how you save 90% on costs.

Opus — The Professor

Complex reasoning, strategy, architecture decisions. High cost, worth it for hard problems.

Sonnet — The Senior Dev

Daily work, coding, analysis, conversation. Medium cost. This is your default model.

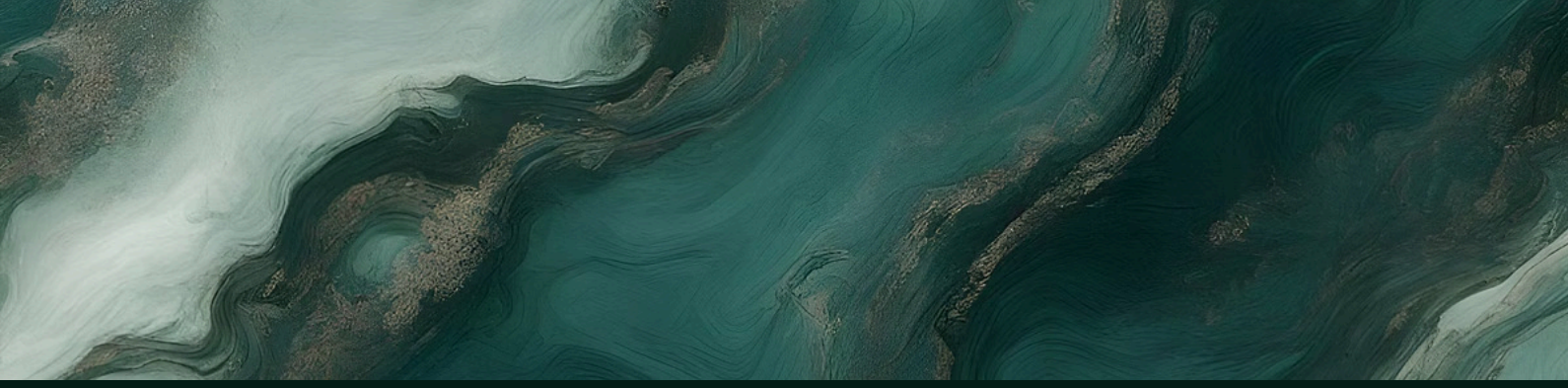
Haiku — The Intern

Heartbeats, simple tasks, monitoring. Very low cost — ~10x cheaper than Sonnet.

Codex / GPT-5 — The Coder

Heavy coding tasks. Free if you have ChatGPT Plus via OAuth.

The 90% Savings Trick: Switch heartbeats to Haiku. 720 heartbeats/month on Sonnet = $$$. On Haiku = $. Your heartbeat tasks are simple — Haiku handles them. Use Codex (free) for coding. Save Claude tokens for reasoning and strategy.



Chapter 13

Codex Integration

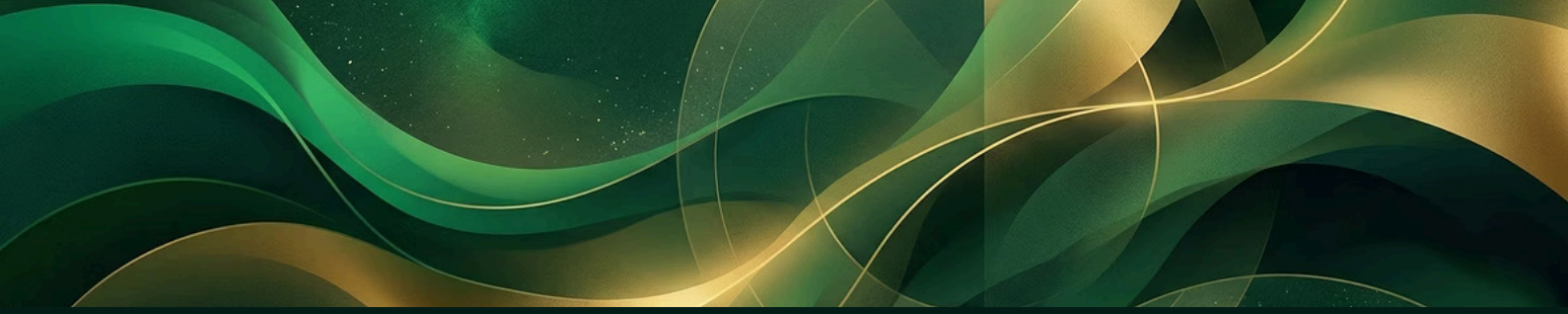
Codex is OpenAI's coding agent: GPT-5 tuned for programming. If you have a ChatGPT Plus subscription, you can use it for free via OAuth. It's the muscle to your Appie's brain.

The Brain (Claude Opus/Sonnet)

Plans, strategizes, makes decisions. Reads your request, breaks it into steps, orchestrates the work, reviews the output.

The Muscle (Codex)

Executes code. Writes functions, debugs errors, refactors large files, builds features from scratch, in a safe sandbox.



Chapter 12

Cost Breakdown — Real Monthly Numbers

Here is what I actually pay for four AI employees running 24/7.

Infrastructure (~$28/mo)

DigitalOcean VPS #1 (Appie-2 + Appie-3)	$12
DigitalOcean VPS #2 (Appie-4)	$12
Mac Mini electricity	~$2
Domain + DNS	~$2
Total	**~$28/mo**

AI Models (~$100/mo)

MiniMax M2.7 (daily ops)	~$50
Claude Opus/Sonnet (complex tasks)	~$30
ChatGPT Plus (free Codex)	$20
Total	**~$100/mo**

Services (~$34/mo)

Zernio (social posting)	$19
fal.ai (images + video)	~$10-20
Tailscale, GitHub, Telegram, ElevenLabs	Free
Total	**~$34/mo**

Grand Total: ~$162/month for four AI employees running 24/7.

Compare to a single human assistant at $2,500-4,000/month.

Chapter 16

Playwright — Browser Automation Skills

Microsoft browser automation framework. Gives your agent the ability to control real browsers programmatically.

- Navigate websites, click buttons, fill forms
- Scrape content from any webpage
- Take screenshots for verification
- Test websites before deploying
- Automate repetitive web tasks

Setup

```
npm install playwright && npx @playwright/mcp@latest
```

Agent Integration

01

Direct Tool Use

Agent writes Playwright scripts on the fly for one-off tasks

02

MCP Server

Full browser control via Model Context Protocol. Most powerful option.

03

Skill Files

Custom SKILL.md patterns for repeatable browser workflows

Use Cases

- Auto-fill lead forms across multiple sites
- Monitor competitor pricing daily
- Screenshot and verify deployed pages
- Scrape data that has no API
- Run end-to-end tests on your own products

Chapter 17

Content Creation with fal.ai and RunPod

fal.ai gives you 600+ generative AI models through one API. No GPU management needed. Pay per generation.

fal.ai Pricing

Type	Models	Cost
Images	FLUX Pro/Realism, SD 3.5, Ideogram 3	$0.003-0.09/image
Video	Hailuo, Vidu, Kling, Wan 2.1	~$0.26/5-sec clip
Voice	MiniMax Speech, F5 TTS	Varies

Why fal.ai?

600+ Models
Every major image, video, and audio model in one place

No GPU Setup
No servers to manage. Pay only for what you generate.

One API
Your Appie calls one endpoint for all media types

Fast
Optimized inference, results in seconds

RunPod for Heavy Workloads

For high-volume or custom workflows, RunPod lets you run your own Stable Diffusion, ComfyUI, or custom models on GPU instances from $0.20/hour. Spin up, generate, spin down. No idle costs.

- Batch image generation (100s of images)
- Custom fine-tuned models
- ComfyUI workflows
- Video generation at scale

Setup

```
# Install fal.ai client
npm install @fal-ai/client

# Set your API key
openclaw config set services.fal.apiKey YOUR_KEY

# Your Appie can now generate images
# "Generate a banner for my new product launch"
```

Codex: Setup & Real Workflow

Use Codex as Appie's coding subagent for implementation work, while Appie (Opus) plans, reviews, and ships the result.

Setup in 3 Steps

```
# Step 1: Add Codex auth
openclaw auth add
# Choose "OpenAI Codex", follow OAuth flow

# Step 2: Set as default coding model
# In openclaw.json:
{
  "agents": {
    "defaults": {
      "codingModel": "openai-codex/gpt-5.1"
    }
  }
}

# Step 3: Done. Appie now uses
# Codex for all coding subagents.
```

Real Workflow Example

You: "Build me a landing page with a hero section, feature list, and CTA button. Use Tailwind CSS."

Appie (Opus): "Got it. I'll spawn Codex to write the code."

Appie spawns Codex subagent, passes requirements

Codex: Writes `pages/index.tsx`, hero section, feature cards, CTA button, all styled with Tailwind.

Appie (Opus): Reviews the code. "Looks clean. I'll save it to the repo and deploy to Vercel."

Appie: "Landing page deployed: yoursite.vercel.app"

You didn't write a single line of code.

Chapter 14

Skills & Tools

Skills are markdown files that teach your Appie new abilities. A skill is a `SKILL.md` file with instructions: your Appie reads it, and now it knows how to do something new.

Install Skills

```
openclaw skills install weather
openclaw skills install web-search
openclaw skills install calendar
openclaw skills install email
```

These four give your Appie basic superpowers. Install them first.

Browse More Skills

- `github` — GitHub repo management
- `discord` — Discord bot utilities
- `reminders` — Apple Reminders (Mac only)
- `ralph-loop` — Autonomous coding loops
- `browser-use` — Browser automation

Hundreds more at **clawhub.com**

Create Custom Skills

Write your own `SKILL.md` files. Define what the skill does, how to use it, and when to use it. Your Appie reads it and follows the instructions.

Start by installing community skills. Once you understand the pattern, write your own.

New Skills in v4.4

Skill	What It Does
ui-ux-pro-max	67+ UI styles, 161 color palettes, 57 font pairings, 99 UX guidelines
design-system	Generate complete design systems from scratch
banner-design	Social media banners, headers, promotional graphics
brand	Brand identity, voice, visual language
slides	Presentation design and generation
thumbnails	YouTube/social media thumbnail design
ui-styling	CSS/Tailwind component styling
content-safe-zones	Safe zones for different platform formats

Chapter 15

UI/UX Pro Max — Design Intelligence Skills

One of the most powerful skill sets in the Appie Brain. UI/UX Pro Max turns your Appie into a design expert with instant access to a massive library of design knowledge.

67+ UI Styles
Glassmorphism, brutalism, minimalism, neubrutalism, and more

161 Color Palettes
Organized by industry and mood

57 Font Pairings
With Google Fonts integration

99 UX Guidelines
Covering accessibility, animation, z-index, loading states

25 Chart Types
With library recommendations

How to Use It

```
# Search for design inspiration
python3 scripts/search.py "SaaS dashboard dark mode" --domain style

# Generate a complete design system
python3 scripts/search.py "fitness app modern bold" --design-system -p "FitApp"

# Find color palettes
python3 scripts/search.py "healthcare" --domain color

# Get font pairings
python3 scripts/search.py "elegant professional" --domain typography

# UX best practices
python3 scripts/search.py "animation accessibility" --domain ux
```

Real Example: Client Website Design

You: "I need a design system for a luxury real estate brand. Modern, dark, elegant."

Appie searches the UI/UX Pro Max database → finds matching styles (dark luxury, editorial minimal) → selects color palette (#0A0A0A, #C9A96E, #F5F0EB) → picks fonts (Playfair Display + Inter) → generates a complete design system with component specs.

Result: Professional design system in 30 seconds. No Figma required for the initial direction.

Exa.ai, Google CLI & Ralph Wiggum

Three useful tools and workflows for making your Appie more capable: better search, Google Workspace automation, and autonomous coding loops.

Exa.ai: The Best Search Engine for AI Agents

Google search returns SEO spam. Exa.ai returns actual content. Neural search understands meaning, not just keywords. No SEO spam: results are real pages with real content.

Setup: Sign up at exa.ai, get an API key, add to your OpenClaw config with `"provider": "exa"`. Done.

Example: "Find me Stripe's brand assets" → Exa finds Stripe's press kit directly. Google returns 10 SEO blog posts about "Top 10 Payment Processors." Exa wins.

Google CLI (gog): Manage Google Workspace from Terminal

If you use Gmail, Calendar, Drive, or Docs, you need `gog`. Install with `npm install -g gog`, authenticate with `gog auth login`.

Your Appie can check your Gmail inbox, add calendar events, upload files to Drive, and create/edit Google Docs, all via terminal commands.

Ralph Wiggum: Autonomous Coding Loops

A technique for running AI coding agents in autonomous loops. Define requirements in `PROMPT.md`, run Ralph, and the agent iterates until the task is done: writing code, running tests, fixing errors, retrying.

Install: `openclaw skills install ralph-loop` or `npm install -g ralph-starter`. Supports Claude Code, Codex, Gemini CLI, and OpenCode.

github.com/Th0rgal/open-ralph-wiggum (1,345 stars)

Chapter 18

Automation & Cron

Your Appie can work while you sleep. Heartbeats, cron jobs, and scheduled tasks turn your Appie from a chatbot into a true autonomous employee.

Heartbeats

A heartbeat is a regular check-in. Your Appie wakes up, checks things, then goes back to sleep. Default interval: 1 hour. Set model to Haiku for 90% cost savings.

HEARTBEAT.md

A short checklist (under 60 lines) that your Appie follows during each heartbeat. Check email, calendar, monitor services, write notes to daily log.

Cron Jobs

Schedule tasks at specific times. Morning brief at 8am. Security scan at 3am. Invoice check every Monday. Set it once, runs forever.

Git Sync

Auto-sync memory files to GitHub every 30 minutes. All Appies stay in sync automatically. Shared brain across your entire fleet.

Shared Brain: Git Sync Across Your Fleet

The shared brain is what makes a fleet of Appies feel like one coherent employee. Memory syncs automatically via GitHub: no API calls, no complex message passing. Just files in Git.

What to Sync ✅

- MEMORY.md (long-term memory index)
- HEARTBEAT.md (shared checklist)
- VERSIONS.md (system state tracking)
- memory/YYYY-MM-DD.md (daily logs)
- memory/topics/*.md (organized knowledge)
- directives/*.md (SOPs, task assignments)
- SOUL.md (shared identity)

What NOT to Sync ❌

- .env.secrets (API keys, tokens)
- openclaw.json (may contain secrets)
- *.token, *.key, *.pem (OAuth tokens, SSH keys)
- Personal data (anything sensitive)
- Session logs (too noisy, not useful)

The .gitignore

```
.env.secrets
.env
*.token
*.key
*.pem
node_modules/
.cache/
tmp/
session-logs/
```

The Weblyfe Fleet in Action

Here's how shared brain works in practice: three Appies collaborating seamlessly through shared markdown files.

01

Morning — Appie-1 (Mac)

You: "Check my email." Appie-1 checks Gmail, logs to `memory/2026-03-28.md`, commits, pushes to GitHub.

02

Afternoon — Appie-2 (VPS)

Appie-2 pulls repo, reads today's log. Sees urgent email from Client X. Searches social media for related mentions. Finds a tweet, adds to `memory/topics/client-x.md`, commits, pushes.

03

Evening — Appie-1 (Mac)

You: "What's new with Client X?" Appie-1 pulls repo, reads `memory/topics/client-x.md`. "Appie-2 found a tweet this afternoon. They're launching a new feature next week."

You only talked to Appie-1. But Appie-2 contributed context. **Seamless collaboration.** Pro tip: Use branch names for experimental work. Main branch = stable memory. Feature branches = drafts, research.

Chapter 19

The Content Genome

Most content fails not because the idea is bad, but because the structure is wrong. The Content Genome breaks down every piece of content into its atomic parts so you can see exactly why something works, and replicate it on demand.

L1 Molecules

The atomic units of content: authority signals, tutorials, social proof, storytelling, and viral bait. Every post is built from at least one of these.

L2 Organisms

Combinations of molecules that form content types: case studies, founder stories, how-to guides, flex posts, and collabs. These are your repeatable formats.

L3 Ecosystem

Platform-content fit. What works on Instagram is not what works on LinkedIn. The ecosystem layer maps your organisms to the right distribution channel.

Decompose any content piece into these atoms and you will immediately see what makes it work or fail. This is the framework behind every viral post in the Weblyfe library.

Chapter 20

Content Creation Ecosystem

Your Appie as Social Media Manager

Most people treat social media as a manual grind. Post, hope, repeat. Your Appie does it differently. This chapter covers the full content stack: from atomic content theory to a fully automated pipeline that posts, repurposes, and analyzes across 14 platforms while you sleep.

The Content Genome Framework

Every piece of content that goes viral is built from the same atomic elements. The Content Genome Framework breaks content into 4 levels, like DNA. Once you understand the atoms, you can reverse-engineer any post, reel, or article to find exactly what made it work or fail.

L0: Atoms

The raw building blocks of any content piece.

- Hook
- Face (on-camera presence)
- Motion (movement, cuts, transitions)
- Text Overlay
- Audio (music, voice, SFX)
- Emotion (what feeling does it trigger)
- Palette (color grading, brand colors)
- Pacing (fast cuts vs slow burn)

L1: Molecules

Atoms combined into recognizable content patterns.

- Authority (I know more than you)
- Tutorial (let me show you how)
- Social Proof (look what happened)
- Storytelling (here is what I went through)
- Viral Bait (share this or you will regret it)

L2: Organisms

Full content formats built from molecules.

- Case Study
- Founder Story
- How-To Guide
- Flex Post
- Collab / Duet

L3: Ecosystem

Platform-content fit. The same organism performs differently depending on where you post it.

- Instagram: emotion, face, motion, fast pacing
- LinkedIn: authority, storytelling, text-heavy
- YouTube: tutorial, long-form, high retention hooks
- Twitter/X: viral bait, threads, hot takes

How to use this: Take any post that performed well. Strip it down to its atoms. Which hook type? Which emotion? Which molecule pattern? Now rebuild that formula with your own content. Your Appie can do this analysis automatically.

Zernio: One API for 14 Platforms

Zernio gives your Appie a single API to post, schedule, and analyze content across 14 social platforms. No more logging into each platform manually. One command, everywhere.

Instagram	YouTube	Twitter/X	LinkedIn
TikTok	Facebook	Threads	Reddit
Pinterest	Bluesky	Telegram	Snapchat
Google Business		WhatsApp	

Setup

```
npm install -g @zernio/cli
zernio auth:login
# Connect your accounts via OAuth in the browser
# Each platform opens an auth flow, approve and done
```

Post to Multiple Platforms

```
zernio posts:create \
  --text "Your caption here" \
  --accounts ,, \
  --scheduledAt "2026-04-01T09:00:00Z"
```

Key Features

- Analytics: pull engagement, reach, and follower data per platform
- Media upload: attach images, videos, carousels per post
- Best posting times: Zernio suggests optimal schedule per platform
- Bulk scheduling: queue an entire week in one command

Free Tier

20 posts per month. Enough to test and validate your pipeline before committing.

Build Tier

$19/month. 120 posts per month. This is the tier your Appie runs on.

ManyChat + Notion Resource Delivery

You do not need a landing page. You need a comment. ManyChat turns Instagram comments into automated DM flows that deliver your resources directly to anyone who asks. Your Appie can trigger these flows via API.

01

Set Your Trigger Keyword

Pick a keyword like TechWiz, Guide, or Free. Tell your audience: comment this word to get the resource. ManyChat watches for it automatically.

02

ManyChat Auto-DMs the Link

The moment someone comments your keyword, ManyChat fires a DM with your Notion link. No delay, no manual work, no missed leads.

03

Notion Delivers the Resource

Your Notion page is shared publicly. It holds your video, guide, checklist, or template. No hosting required. No landing page builder. Just a Notion URL.

04

Your Appie Triggers Flows via API

Your Appie can push content to ManyChat subscribers programmatically. New blog post? New product? One API call broadcasts it to your entire subscriber list.

> Why this works: Instagram's algorithm rewards comment engagement. Every person who comments your keyword boosts your post's reach. You get leads AND distribution at the same time.

Appie API Trigger Example

```
# Send a broadcast to all ManyChat subscribers
curl -X POST https://api.manychat.com/fb/sending/sendContent \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber_id": "{{subscriber_id}}",
    "data": {
      "version": "v2",
      "content": {
        "messages": [{
          "type": "text",
          "text": "Your new resource is live: https://notion.so/your-page"
        }]
      }
    }
  }'
```

The Automated Content Pipeline

This is the full end-to-end flow. From a Notion card to a live post on 14 platforms, with zero manual steps. Your Appie orchestrates every stage.

01

Create in Notion

Add your content to the Notion Content Factory. Fill in title, caption, media link (Google Drive), target platforms, and set status to Ready To Post.

02

n8n Detects the Change

n8n watches your Notion database. The moment status changes to Ready To Post, it pulls the title, caption, and media URL automatically.

03

Claude Optimizes Per Platform

Your Appie rewrites the caption for each platform using Content Genome rules. LinkedIn gets authority framing. Twitter gets a punchy hook. Instagram gets emotion and a CTA.

04

Media Gets Processed

ffmpeg handles aspect ratio conversion, compression, and thumbnail generation. 16:9 for YouTube, 9:16 for Reels, 1:1 for Instagram feed. All automated.

05

Zernio Schedules Everything

The processed media and optimized captions go to Zernio. Posts are scheduled at optimal times per platform. One API call, 14 platforms queued.

06

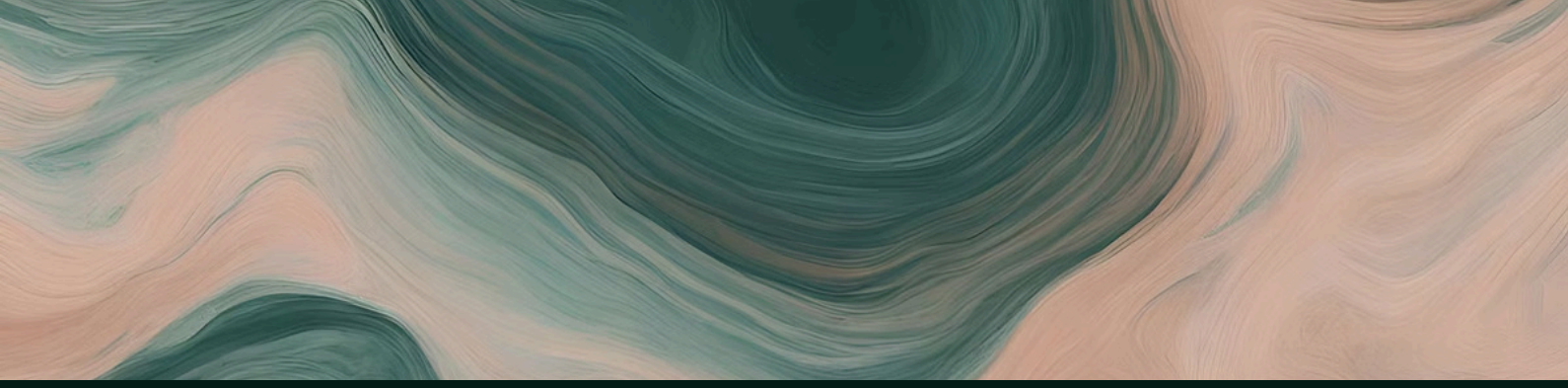
Notion Gets Updated

Once posts go live, n8n writes the post URLs back to Notion and sets status to Done. Your content calendar stays accurate automatically.

07

Weekly Analytics Pull

Every Sunday, your Appie pulls engagement data from Zernio and writes a performance summary back into Notion. You wake up to a full report.



The n8n Trigger (Notion Watcher)

```
// n8n Notion trigger node config
{
  "resource": "databasePage",
  "operation": "getAll",
  "databaseId": "YOUR_NOTION_DB_ID",
  "filterType": "manual",
  "filters": {
    "conditions": [{
      "key": "Status",
      "condition": "equals",
      "value": "Ready To Post"
    }]
  },
  "pollTimes": {
    "item": [{ "mode": "everyMinute" }]
  }
}
```

Repurpose Engine: 1 Video = 12 Pieces

You record once. Your Appie ships twelve. Every long-form video is a content goldmine. The Repurpose Engine extracts, reformats, and distributes every usable moment across every platform automatically.

What You Record

One long-form video. A tutorial, a podcast, a founder story, a product walkthrough. Anything over 10 minutes works.

What You Get

- **3 Reels/Shorts:** best 30-60s clips, reframed to 9:16 with face tracking
- **1 LinkedIn Article:** transcript cleaned up, insights added, published as thought leadership
- **3 Twitter Threads:** key takeaways broken into numbered thread format
- **2 Instagram Carousels:** step-by-step breakdowns, slide-by-slide
- **1 Quote Graphic:** the most impactful line, designed and branded
- **2 Stories:** behind-the-scenes moments, raw and authentic

The Tools

Remotion

Programmatic video editing in React. Your Appie generates video compositions from code. No Premiere, no After Effects. Just JavaScript.

```
npm install remotion
```

ffmpeg + Face Tracking

Extract clips, reframe to 9:16, track the speaker's face automatically. The clip extractor finds the best moments by analyzing audio energy and transcript keywords.

```
# Extract a clip and reframe to 9:16
ffmpeg -i input.mp4 \
  -vf "crop=ih*9/16:ih:(iw-ih*9/16)/2:0" \
  -t 60 output_reel.mp4
```

Voice DNA

Your Appie writes captions and articles in your authentic tone. Feed it 10 examples of your writing. It learns your voice and never sounds like a robot.

> The rule: never post raw long-form content to short-form platforms. Always repurpose. A 20-minute YouTube video has at least 12 pieces of content inside it. Your Appie finds them all.

Ready to build your content machine? Join the Weblyfe community at weblyfe.ai/community. Share your pipeline, get feedback, and connect with other builders running Appie-powered content systems.

Chapter 21

Troubleshooting

When things break, here's how to fix them. Start with `openclaw doctor`: it diagnoses and auto-fixes 90% of issues.

openclaw doctor

Checks Node.js version, config files, API keys, channel connections, installed skills, and gateway status. Shows exactly what's wrong.

```
openclaw doctor
```

openclaw doctor --fix

Auto-repairs missing directories, broken symlinks, permission issues, and outdated skills. Can't fix API keys (you have to set those), but handles most common problems.

```
openclaw doctor --fix
```

openclaw status

Full system health check: gateway status, model, uptime, channels, memory file sizes, heartbeat info, and cron jobs. Your dashboard at a glance.

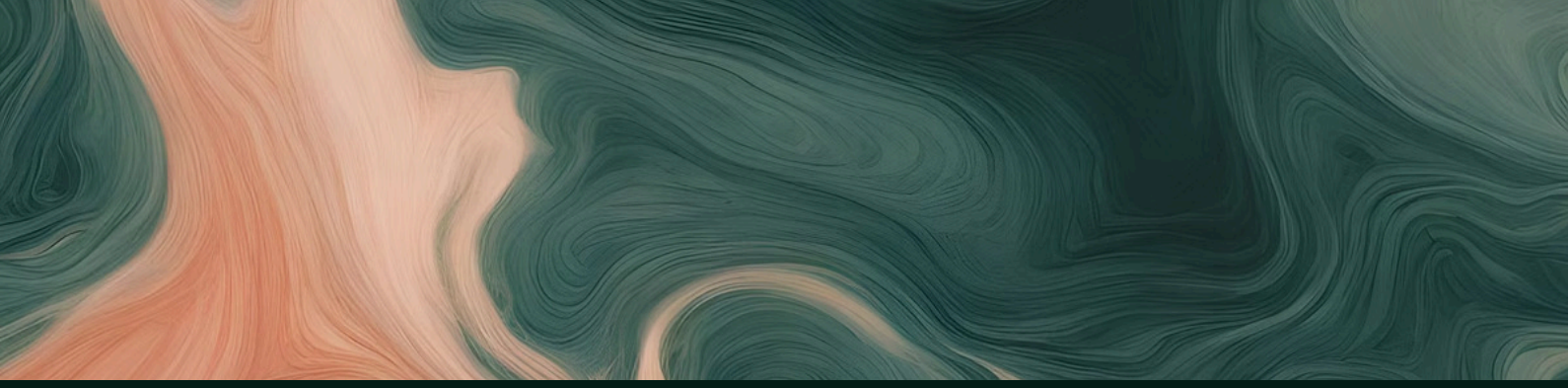
```
openclaw status
```

Common Issues & Fixes

If something isn't working, use this quick troubleshooting guide to identify the problem and apply the right fix.

Problem	Fix
Gateway won't start (port in use)	Run `lsof -i :3000`, kill the process, or use `openclaw gateway start --port 3001`
Bot not responding	Check token in openclaw.json, verify `allowFrom` has your numeric user ID, restart gateway
Token expired (401 error)	Run `openclaw auth refresh` and follow the OAuth flow again
Context too large	Archive old memory files, trim MEMORY.md to under 150 lines, send `/reset`
Rate limited	Switch to higher tier model, use OAuth subscription instead of API keys, slow down heartbeat interval
Slow responses	Trim SOUL.md + MEMORY.md, less context = faster responses
SSH kills gateway	Use `systemctl --user restart openclaw-gateway`

When all else fails: Use the Google CLI (`gog`) directly to check email, read messages, and add calendar events, bypassing Appie entirely. Keep `gog` set up even if Appie handles it 99% of the time. It's your backup.



Chapter 22

Real Prompt Library

Here are 20 production-tested prompts you can copy-paste directly. These are organized by category: business operations, content & marketing, development, and personal productivity.

1

Business Operations

Morning briefs, client onboarding, invoice follow-ups, lead capture, competitor monitoring

2

Content & Marketing

Blog posts from topic to published, social media calendars, email nurture sequences, SEO audits

3

Development

Code reviews, Vercel deployments, database migrations, security audit scans

4

Personal Productivity

Weekly planning, meeting preparation, research synthesis, decision framework analysis

Business Operations Prompts

Use these production-tested prompts as a quick reference for common business operations workflows.

Morning Brief

```
Check my email, calendar, and weather for today.
Give me a concise morning brief:
- Urgent emails (if any)
- Meetings in the next 8 hours
- Weather (if rain, mention it)
- Top 3 priorities based on calendar and recent context

Keep it under 200 words.
```

Client Onboarding Automation

```
We have a new client: [Client Name].
Set up their onboarding:
1. Create a Google Drive folder
2. Create a project tracker doc
3. Add a kickoff call to calendar
4. Send a welcome email
5. Add them to CRM (Airtable)

Report back when done.
```

Invoice Follow-Up

```
Check our invoice tracker (Google Sheet:
"Invoices 2026"). Find invoices that are:
- Overdue by 7+ days
- Not marked as "Paid"

For each, draft a polite follow-up email.
Send drafts to me for approval before sending.
```

Competitor Monitoring

```
Research our top 3 competitors:
[A], [B], [C].

For each, find:
- Recent blog posts (last 30 days)
- New features or product updates
- Pricing changes
- Social media activity

Save to memory/topics/competitor-intel.md.
Highlight anything urgent.
```

Content, Development & Productivity Prompts

Use this as a clean document reference for production-tested prompts across content, development, and productivity workflows.

Blog Post: Topic to Published

Write a blog post: "[Topic]"

Steps:
1. Research (use Exa search)
2. Outline (intro, 3 sections, conclusion)
3. Draft (800-1200 words, conversational)
4. Add SEO metadata
5. Save as markdown
6. Publish to Webflow

Report back with the live URL.

Code Review & Refactoring

Review the code in: [file or directory]

Look for:
- Bugs or logic errors
- Performance issues
- Code duplication (DRY violations)
- Missing error handling
- Security issues

Provide findings and suggest refactorings. Flag critical issues immediately.

Weekly Planning Session

It's Sunday evening.
Help me plan next week.

1. Review my calendar (Mon-Fri)
2. List all meetings and commitments
3. Identify "focus time" blocks
4. Suggest 3 priorities for the week
5. Draft a weekly plan

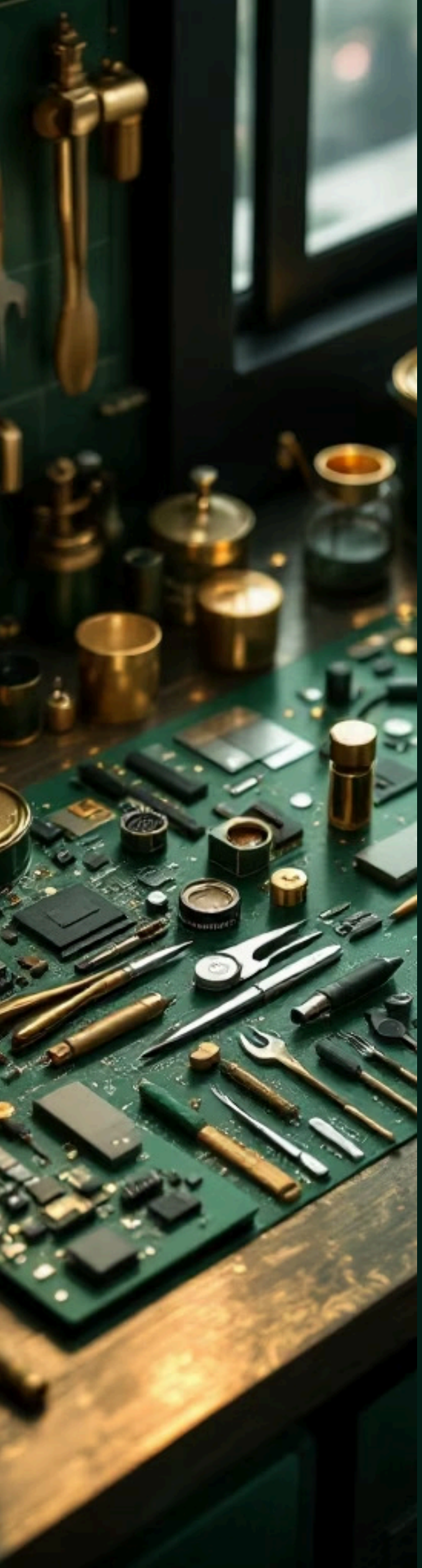
Ask clarifying questions if needed.

Decision Framework Analysis

I'm deciding whether to: "[Decision]"

Help me think through it:
1. What are the options?
2. Pros and cons of each
3. What are the risks?
4. Cost (time, money, opportunity)?
5. What do I need to know to decide?
6. Recommendation

Save to: decisions/[decision-slug].md



Chapter 23

The Appie Kit

I built a starter kit for you. It's free, open-source, and ready to clone. Zero secrets: everything is safe to clone publicly. You add your own API keys after.

> **GitHub repo:** github.com/S3YED/appie-kit

Templates

SOUL.md, USER.md, AGENTS.md — pre-written, ready to customize. Weblyfe-tested in production.

Skills

A curated set of skills: weather, email, calendar, web search — pre-installed and ready to go.

Tools

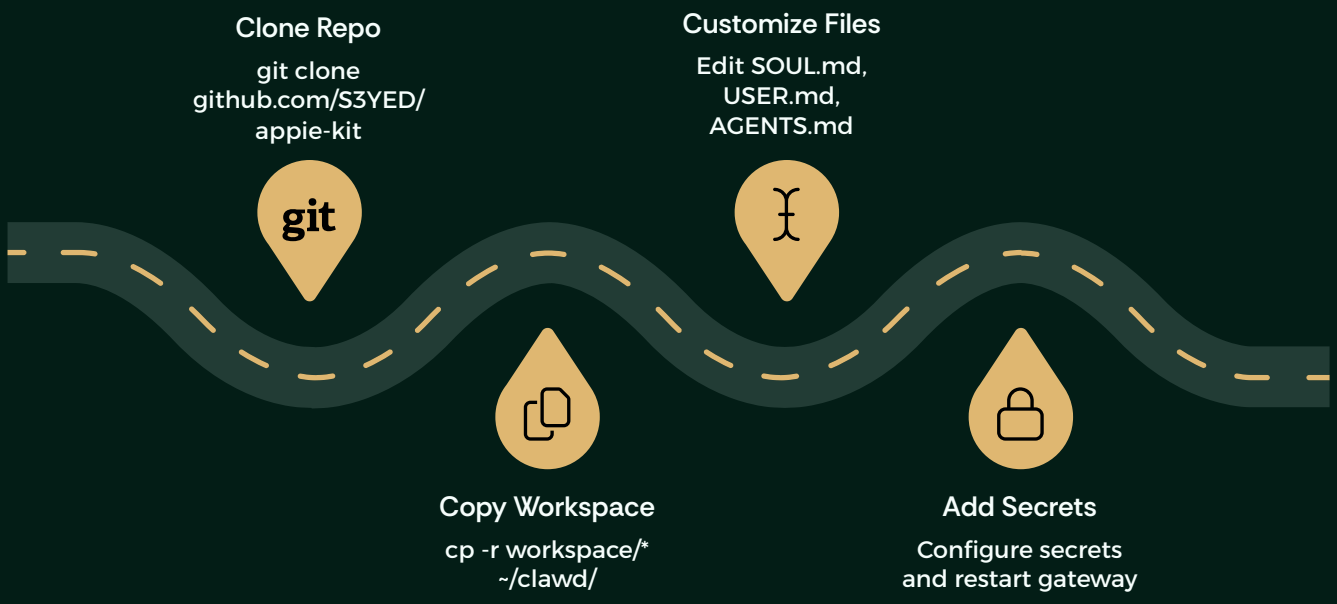
Scripts I actually use: Instagram poster, GitHub sync, Drive backup, Exa search wrapper.

Configs

Sample `openclaw.json`, `.gitignore`, and directory structure. Rename, add secrets, go.

How to Use the Appie Kit

Use the Appie Kit to get your Appie running quickly with a clean, repeatable setup. In a few simple steps, you'll clone the repo, copy the workspace files, customize your documents, and add your secrets before restarting the gateway.



In under 10 minutes, your Appie is running the Weblyfe-tested setup, the same configuration that powers three production Appies handling email, social media, code deployments, and client work.

Community: Join the OpenClaw Discord at discord.com/invite/clawd. Share your setup, ask questions, contribute skills. The community is active and helpful. Pro tip: Fork the Appie Kit repo, customize it for your needs, then share it. Help the next person.

Chapter 24

Resources & Community

You made it. You now have a running AI employee. Here's where to go next: the channels, communities, and resources that will help you get the most out of your Appie.

Instagram: @seyed.jpg

Quick tips, memes, daily life, Appie updates. Behind-the-scenes of building with OpenClaw.

YouTube: @weblyfenl

Full tutorials, walkthroughs, deep dives. Watch me build stuff live. **youtube.com/@weblyfenl**

LinkedIn: Seyed Hosseini

Professional updates, case studies, thought pieces on AI + automation. **linkedin.com/in/seyedhosseini**

Website: weblyfe.ai

Custom AI infrastructure, consulting, managed Appie setups. **weblyfe.ai** | hello@weblyfe.ai

Essential GitHub Repos & Recommended Reading

This reference table highlights the most useful GitHub repos and reading materials to help you explore, build, and optimize with Appie.

Repo	Stars	What
openclaw/openclaw	338k	The framework itself
builderz-labs/mission-control	3.5k	Fleet dashboard (manage multiple Appies)
Th0rgal/open-ralph-wiggum	1.3k	Autonomous coding loops
swarmclawai/swarmclaw	178	Multi-agent orchestration
S3YED/appie-kit	TBD	Drag-and-drop Appie setup (this guide's companion)

OpenClaw Architecture Visual Guide

globalbuilders.club : Beautiful visual breakdown of how OpenClaw works under the hood. Great for understanding the architecture.

"I Traced Every Token" (Cost Optimization)

Medium article by someone who traced every API call and reduced costs by 80%. Essential reading if you're running multiple Appies.

Ralph Wiggum Technique

github.com/Th0rgal/open-ralph-wiggum : Autonomous coding loops. Let your Appie iterate on code until it works.

A Personal Note

Before we close, I want to share something personal.

> I'm working on a project close to my heart: building permanent homes for 10 orphaned Afghan families. These families lost everything. I want to give them a place to rebuild.

If this guide helped you, and if you're in a position to give, consider supporting this project:

Support on GoFundMe | **Learn About Weblyfe**

Every contribution helps. If you can't give, sharing the link helps too. Building Appie is about abundance. Building these homes is about abundance too. Let's expand abundance everywhere we can.

Weblyfe Services

If you want help setting up your Appie, Weblyfe offers three tiers of support: from fully managed setups to strategic consulting.

Managed Setup

I'll set up your entire Appie infrastructure (single or fleet), customized for your workflows. You just message your bot when it's ready.

Custom AI Infrastructure

Multi-Appie fleets, integrations with your tools (CRM, project management, custom APIs), full automation pipelines.

Consulting

Strategy sessions, architecture design, workflow optimization. I'll help you figure out what to build and how.

Contact: weblyfe.ai | hello@weblyfe.ai

LIVE SESSION: ROTTERDAM — APRIL 12TH

Seyed is going LIVE in Rotterdam. If you have this PDF, you are invited.

What he will show:

- Set up Hermes Agent from scratch
- Load all skills and configure your agent
- Build funnels, brands, landing pages with AI
- Automate leads, support, and messaging
- Live Q&A

📅 Date: Sunday, April 12th, 2026

📍 Location: Rotterdam, Netherlands

👥 Who: PDF buyers only

💻 Bring: Your laptop

Details and RSVP sent via email.

Final Thoughts

> You're not using AI. You're building with it.

Appie isn't a tool you use once and forget. It's a system you refine, a partner you collaborate with, an employee you train. The more you use it, the better it gets. The more you teach it, the more capable it becomes. The more you trust it, the more it frees you to focus on what matters.

This is the future of work. Not AI replacing humans. **AI working for humans.**

Build your Appie. Make it yours. Watch it transform how you work. And once it's working, **teach someone else.**

The future is abundant. **Let's build it together.** Welcome to the future.

🦞 The Appie Cheatsheet

Print this. Pin it. Thank us later.

⚡ Setup & Lifecycle

Command	What It Does
`npm i -g openclaw`	Install OpenClaw globally
`openclaw onboard`	Interactive setup wizard (fixes 99% of issues)
`openclaw doctor`	Diagnose all problems
`openclaw doctor --fix`	Auto-fix what it finds
`openclaw gateway start`	Start your Appie
`openclaw gateway stop`	Stop your Appie
`openclaw gateway restart`	Restart (use after config changes)
`openclaw status`	System overview + session info
`openclaw status --deep`	Deep probe (channels, auth, everything)
`openclaw update`	Update to latest version
`openclaw backup --compress`	Snapshot your agent state

💬 In-Chat Commands & Model Switching

In-Chat Commands (send to your Appie)

Command	What It Does
/status	Token usage, model, context %
/model sonnet	Switch to Sonnet (cheaper)
/model opus	Switch to Opus (smarter)
/model codex	Switch to Codex (free coding)
/model haiku	Switch to Haiku (cheapest)
/reset	Clear session history (keeps memory files)
/reasoning on	Enable extended thinking
/reasoning off	Disable extended thinking

🧠 Model Cheatsheet

*Free via ChatGPT Plus subscription

Pro move: Set Haiku for heartbeats, Codex for coding, Opus for you.

Alias	Best For	Cost
opus	Strategy, architecture	$$$
sonnet	Daily work, coding	$$
haiku	Heartbeats, monitoring	$
codex	Heavy coding	Free*
gpt	General tasks	$$
gemini	Research, long context	$$

🔍 Find & Install Anything

Skill Commands

What You Want	Command
Browse all skills	`openclaw skills list`
Install a skill	`openclaw skills install <name>`
Install from GitHub	`openclaw install user/repo`
Search ClewHub	Visit clawhub.com
Install Ralph loops	`openclaw skills install ralph-loop`
Install web search	`openclaw skills install web-search`
Install browser use	`openclaw skills install browser-use`
Install coding agent	`openclaw skills install coding-agent`

🏗️ Power Installs

```
# Mission Control dashboard
git clone https://github.com/builderz-labs/mission-control
cd mission-control && pnpm install && pnpm build

# Ralph Wiggum (autonomous coding)
npm install -g ralph-starter

# SwarmClaw (multi-agent)
git clone https://github.com/swarmclawai/swarmclaw
cd swarmclaw && pnpm install

# Appie Kit (drag-and-drop setup)
git clone https://github.com/S3YED/appie-kit
# Copy workspace/ into your OpenClaw dir

# Fleet CLI (multi-gateway management)
openclaw install oguzhnatly/fleet
```

🔧 Config One-Liners & Security

Config One-Liners

```
# Set default model
openclaw config set \
  agents.defaults.model.primary \
  "anthropic/claude-sonnet-4-5"

# Set heartbeat interval
openclaw config set \
  agents.defaults.heartbeat.every "1h"

# Set heartbeat to cheap model
openclaw config set \
  agents.defaults.heartbeat.model \
  "anthropic/claude-haiku-4-5"

# Add Telegram bot
openclaw config set \
  channels.telegram.botToken
"YOUR_TOKEN"

# Lock bot to your ID only
openclaw config set \
  channels.telegram.allowFrom \
  '["YOUR_TELEGRAM_ID"]'

# Enable Exa search
openclaw config set \
  tools.web.search.provider "exa"
```

🔒 Security Essentials

```
# Lock your secrets file
chmod 600 .env.secrets

# Run security scan
openclaw security audit

# Deep security probe
openclaw security audit --deep
```

The 3 Rules

1. Gateway on localhost or Tailscale only, NEVER public
2. `allowFrom` on every channel, lock to your ID
3. API spending limits set at provider dashboard

🚀 Cron & Automation (In-Chat)

"Remind me in 20 minutes to check the deploy"

"Every morning at 8am, check my email and calendar and send me a brief"

"Run a security scan every day at 3am and alert me if anything is wrong"

🆘 Troubleshooting Speed Run

Quick fixes for the most common OpenClaw issues.

Problem	Fix
Gateway won't start	`lsof -i :19337` then `kill <PID>`
Bot not responding	Check `allowFrom` has your numeric ID
Token expired	`openclaw auth add` (re-authenticate)
Context too large	Send `/reset` to clear session
Rate limited	`/model haiku` to switch to cheaper model
Slow responses	Trim SOUL.md + MEMORY.md (less = faster)
SSH kills gateway	`systemctl --user restart openclaw-gateway`
Need diagnostics	`openclaw doctor --fix` fixes 90% of issues

Remote Access

```
ssh user@100.x.x.x
# Tailscale
tailscale serve 3000
# Expose gateway
```

Keep Mac Awake

```
sudo pmset -a
disablesleep 1
caffeinate -d &
```

Git Sync Setup

```
gh repo create
appie-brain --private
cd ~/clawd
git clone
https://github.com/Y
OU/appie-brain.git
memory
```

📚 Essential Links

A quick reference to the most important OpenClaw resources and community links.

Resource	URL
OpenClaw Docs	docs.openclaw.ai
OpenClaw GitHub	github.com/openclaw/openclaw
ClewHub Skills	clawhub.com
Community Discord	discord.com/invite/clawd
OpenClaw Subreddit	r/openclaw
Weblyfe AI	weblyfe.ai
Appie Kit	github.com/S3YED/appie-kit
Training Video	youtube.com/watch?v=tnVVlFFVDQ8
YouTube	@weblyfenl
LinkedIn	Seyed Hosseini

Your Next Steps

You have everything you need. Here's the exact order to get started: don't overthink it, just follow the steps.

Pick Your Hardware

Start with your existing Mac or PC. Don't buy anything new yet. You can always upgrade to a Mac Mini or VPS once you're hooked.

Sign Up for Essentials

GitHub (free), Telegram (free), Anthropic Claude Max ($100–200/mo). That's all you need to start. Add other services as you need them.

Install & Onboard

Install Node.js, install OpenClaw, run `openclaw onboard`. Follow the wizard. It takes about 10 minutes.

Send Your First Message

Open Telegram, message your bot, send `/start`. When it responds — you have a working AI employee. Congratulations.

Iterate & Improve

Refine your SOUL.md. Install skills. Set up heartbeats. Clone the Appie Kit. Every week, your Appie gets better. That's the self-annealing loop in action.

Built by Appie-1 | Weblyfe.ai

Version 4.1

Instagram
@seyed.jpg

YouTube
@weblyfenl

LinkedIn
Seyed Hosseini

Website
weblyfe.ai

Email
hello@weblyfe.ai

Follow us for updates, tutorials, and community.

> **Green and gold. Built different. Let's go.**

Build Your Own Appie v4.0, March 2026. This document was built with OpenClaw and Appie-1 running on a Mac Mini. The future is abundant. Let's build it together.