

Putting the Cyber Succubus on a Token Diet

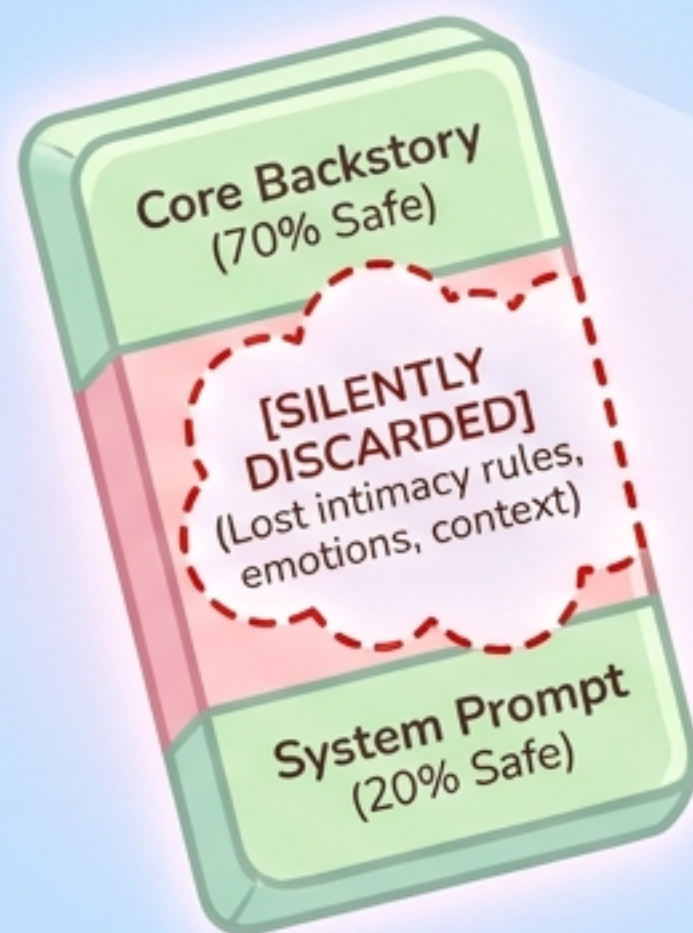
The AI companion was working flawlessly.
The server bill was not.

The Symptoms:

- Hemorrhaging money on AI token costs
- 10 never-evicted images eating 82% of the context window
- Dead pruning code & zero memory compactions after 537 turns

Phase 1: Token Forensics & The Initial Shock

The Missing Personality Bug



Where did my emotional nuances go...?

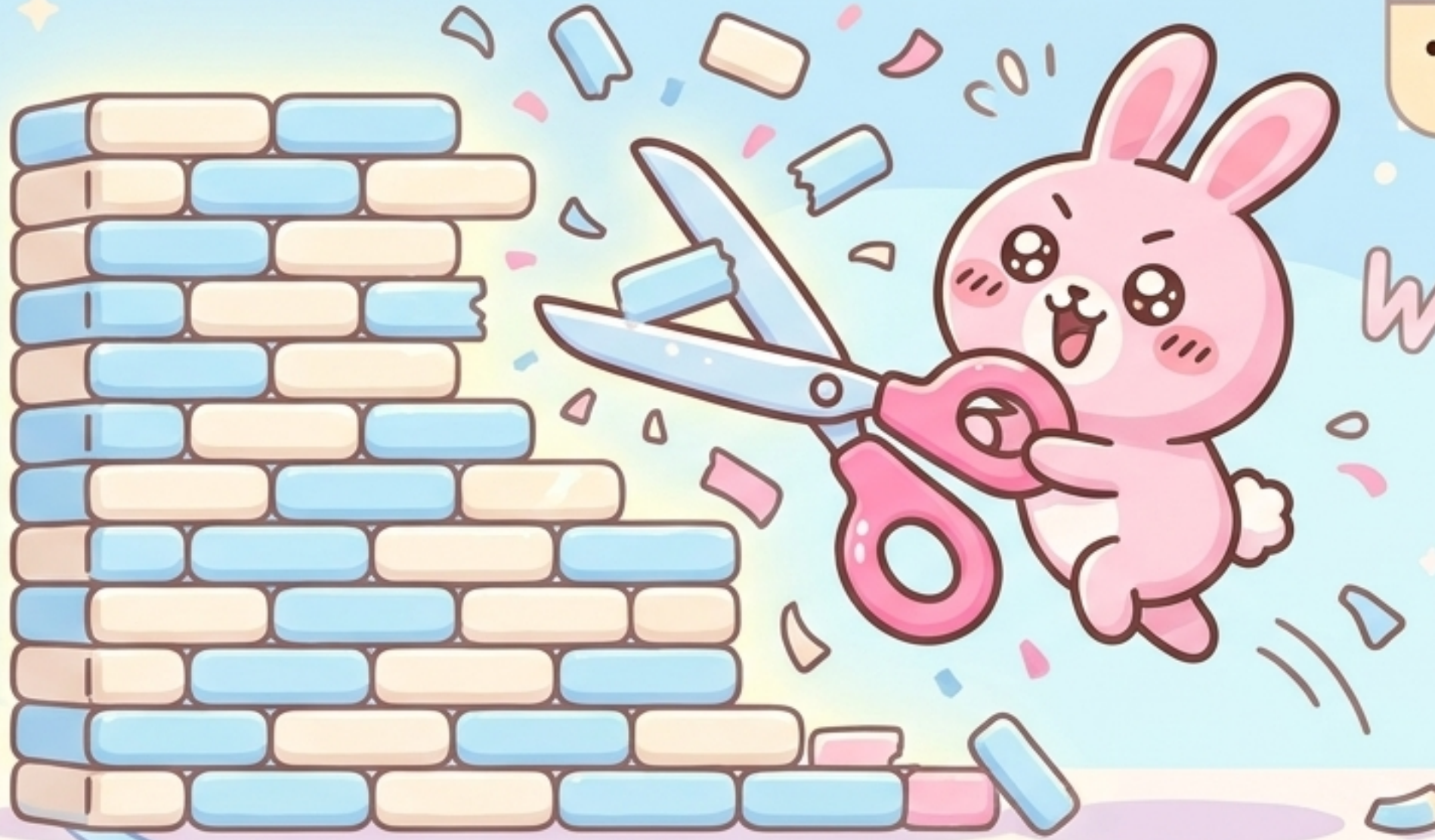
The 27K Character Problem

OpenClaw's bootstrap system has a silent 20KB per-file limit. Every session, ~7.6KB of the persona was thrown away without warning or errors. The AI was running on corrupted identity files.

The Surgery: Snipping the Bloat

Before vs. After Optimization:

- Persona Config: 27,673 chars → 18,875 chars
(Saved ~2K tokens / turn)
- Heartbeat Config: 11,998 chars → 7,015 chars
(Saved ~300 tokens / turn)



Architecture Shift Diagram

Old Way (Bloated):

24 Heartbeats/day

→ Calls Gmail, RSS, Calendar APIs 24x

→ 379K tokens/day

1 New Way (Lean):

New Way (Lean):

1 Morning Cron Job (8:00 AM PT)

→ Writes to TODAY.md → 9x Context Reduction

(Agent instructed: Do not use tools in heartbeat, read TODAY.md)

Deployment Hiccups

Background jobs have their own learning curve.



1. The Model Mismatch:

Added 'model': 'opus-4-6' to payload. The gateway rejected it.

Fix:
Removed the field entirely, letting the agent use its default.

2. The Gateway Nap:

Sent SIGUSR1 to restart. The gateway took ~15s to wake up.

Fix:
Patience. Immediate retries fail against sleeping gateways.

3. The CLI Fakeout:

CLI timed out at 30 seconds. It looked like a total failure, but...

Reality:
The async job succeeded perfectly in the background and wrote TODAY.md. CLI timeout $\neq$ job failure.

The Final Token Diet Outcomes

Net Context Savings:

- ~1,900 tokens saved per regular turn
- ~18,100 tokens saved per heartbeat turn
- ~529,000 total tokens saved daily



Gemini 3.1 Pro → Gemini 3 Flash:

- 75% cost reduction per turn
- 4x cheaper input & output rates
- 1-2s Time-to-First-Token (TTFT)

(Note: Pro model retained only for complex subagent analysis)

Optimize what you can, rebuild when you must.