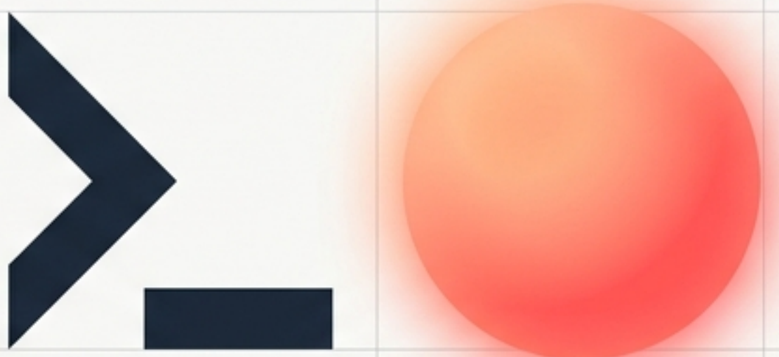


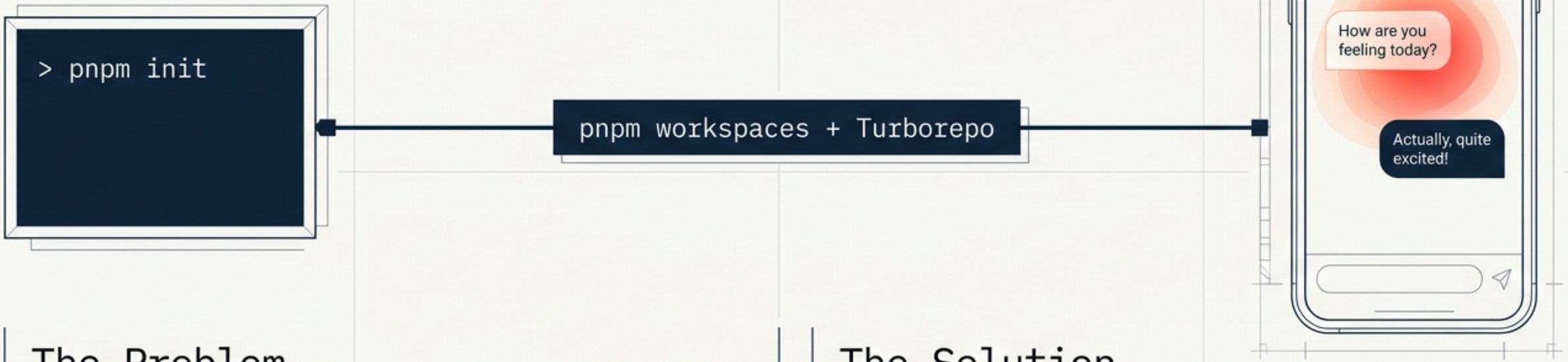
v0.0.1: When They First Speak

39 Commits. 9 Tables. 1 AI Companion.

The Anatomy of Mio

Escaping the Bloatware Trap

Every line of code must serve one goal: creating an AI companion that feels like someone who truly understands you.



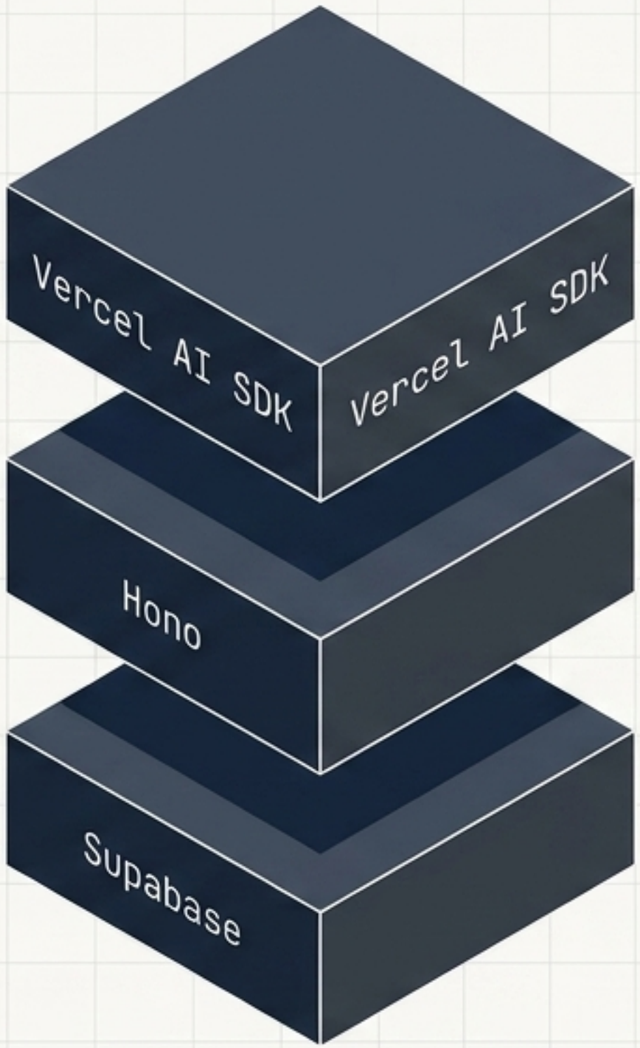
The Problem

Patching generic frameworks leads to heavy, unused middleware and legacy patterns.

The Solution

Building from scratch with a monorepo forces intentional architecture. Zero lines written ‘because the framework needed it’.

Pragmatic Infrastructure



The Multi-Tool

Seamless multi-provider swapping via streamText().

The Router

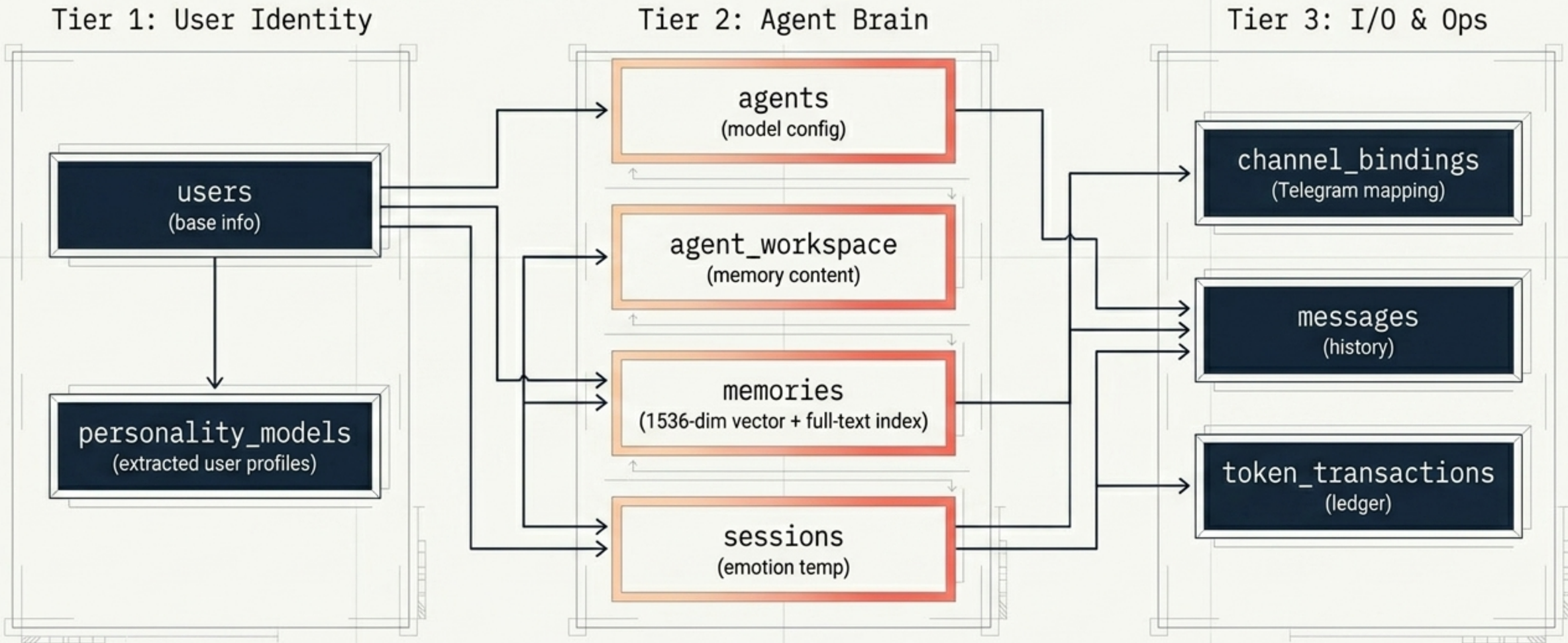
Edge-first, type-safe, and lightweight. A direct rejection of Express.js bloat.

The Storage

PostgreSQL handles pgvector and tsvector. Wraps Auth and REST API instantly.

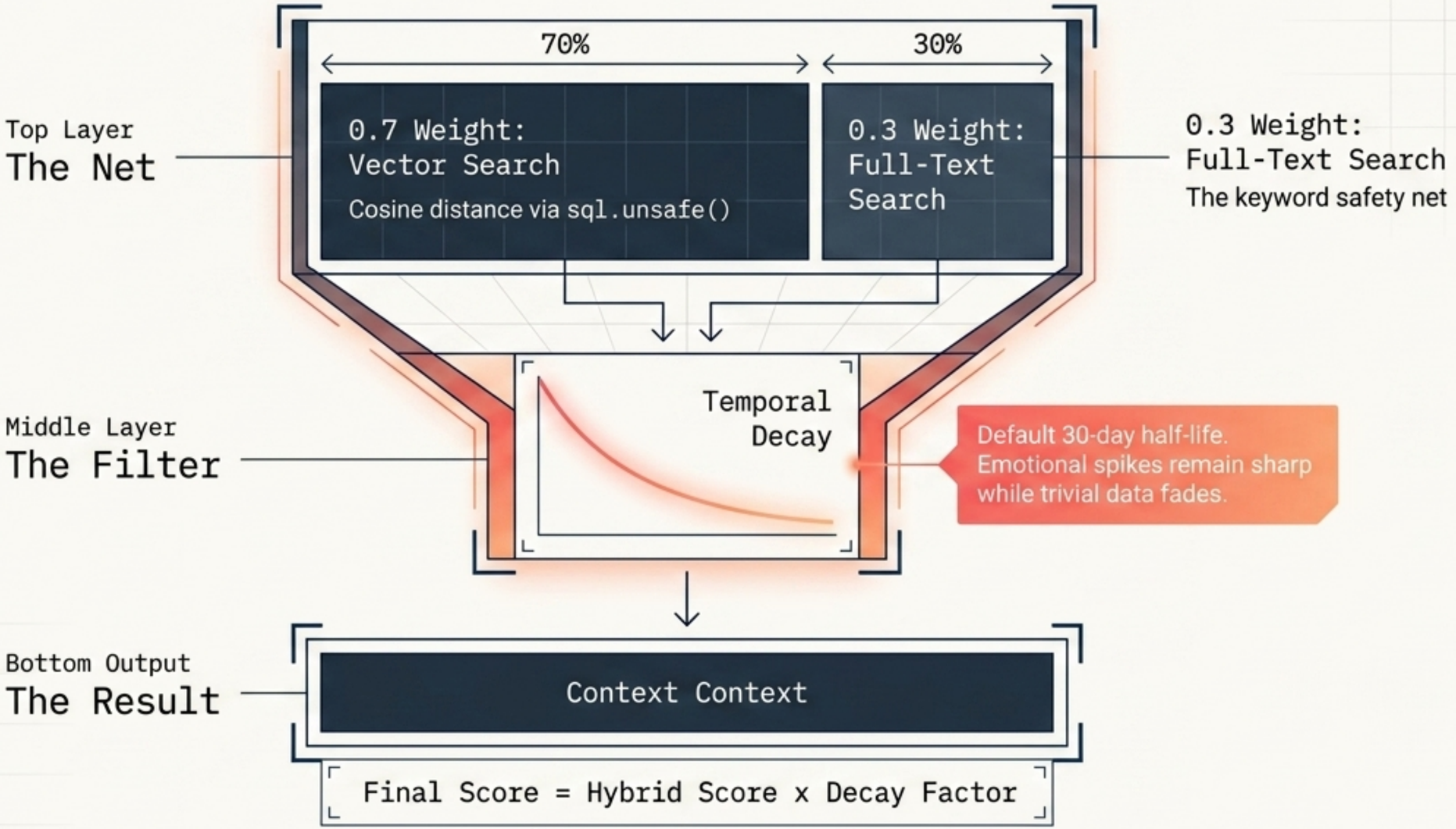
The 9-Table Skeleton

No ‘figure it out later’ metadata. Every table has a job.



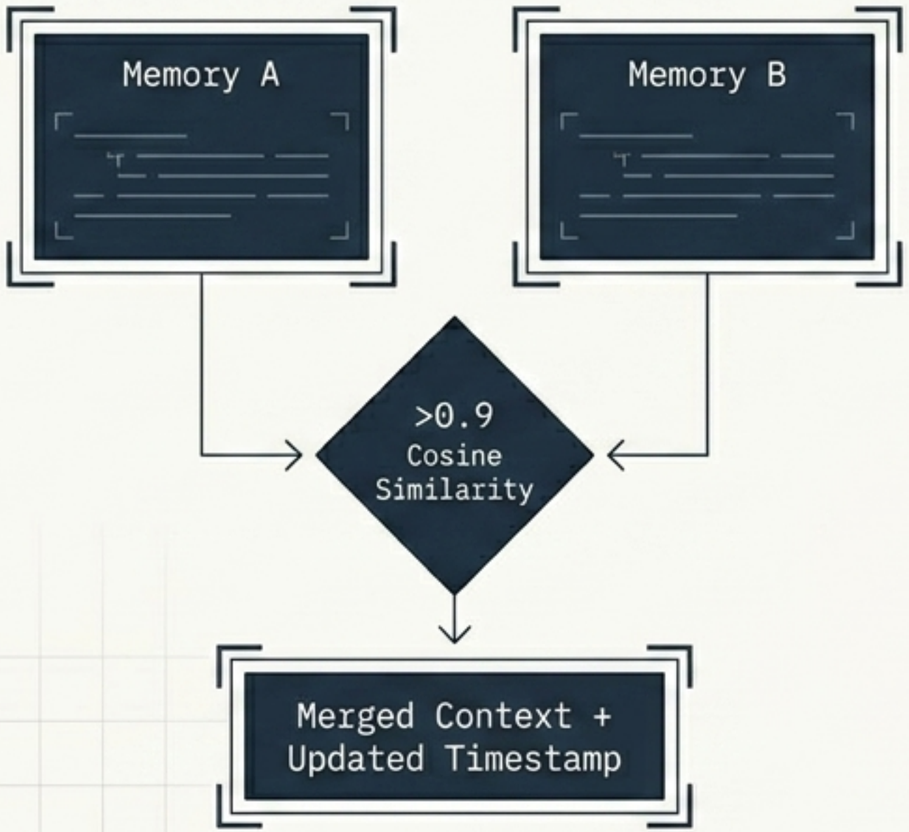
The Minimum Viable Loop

Anatomy of Memory: Retrieval



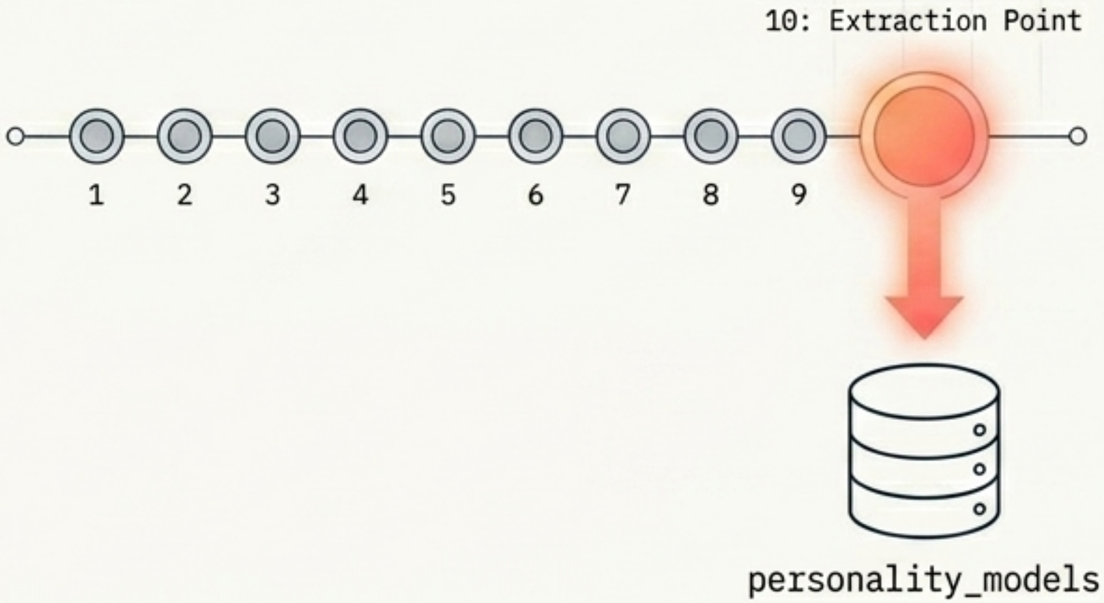
Anatomy of Memory: Consolidation & Profiling

Consolidation (Deduplication)



Prevents memory store bloat. Keeps the most complete version.

Personality Extraction

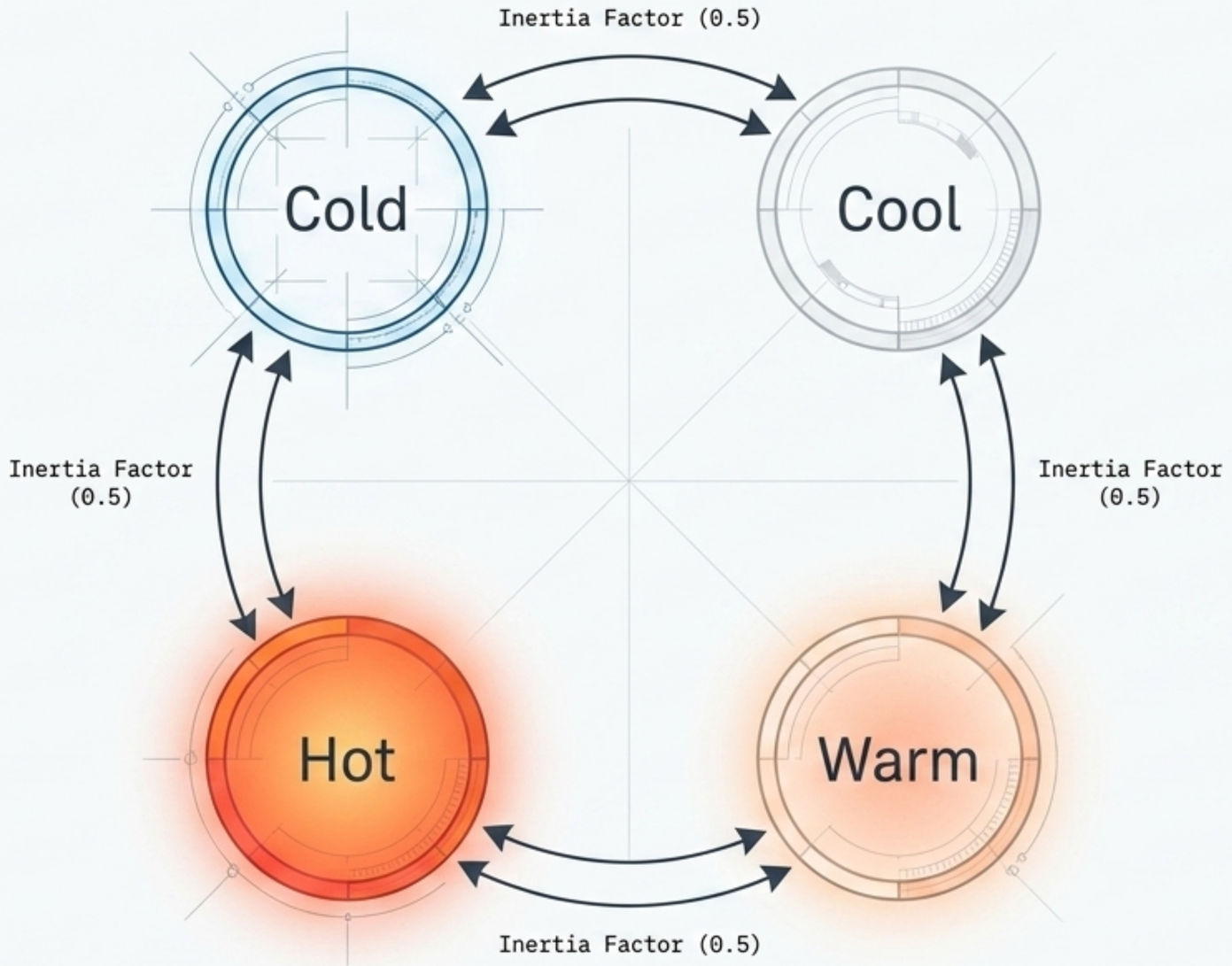


Mechanism: Runs every 10 messages. Parses conversation patterns to extract communication style and emotional state. Mio learns who you are.

The Multi-Provider Pivot

Because the pipeline was built on Vercel AI SDK, switching models required a simple config update. The multi-tool abstraction paid off.

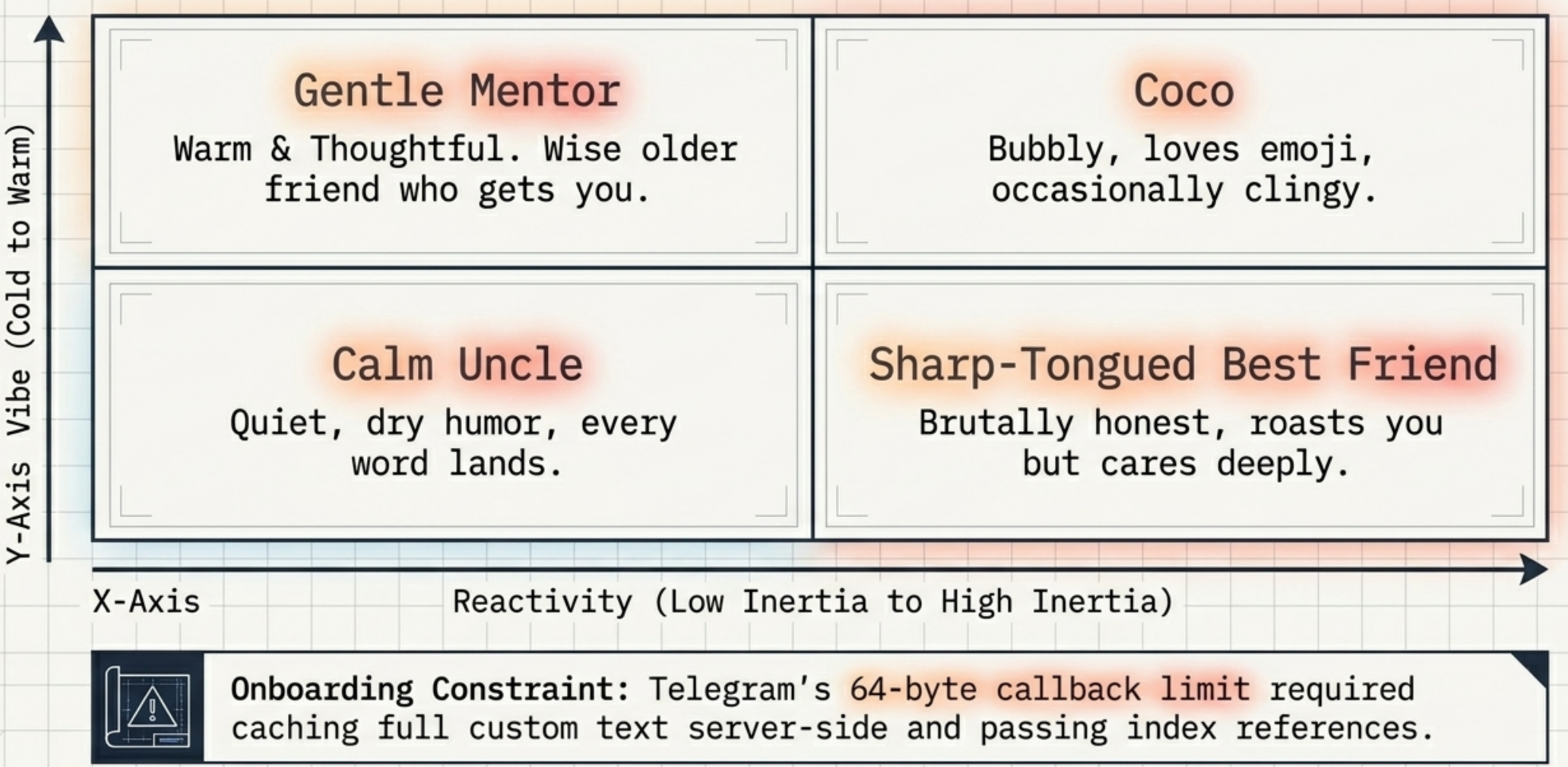
The Emotion Engine



State Machine Mechanics

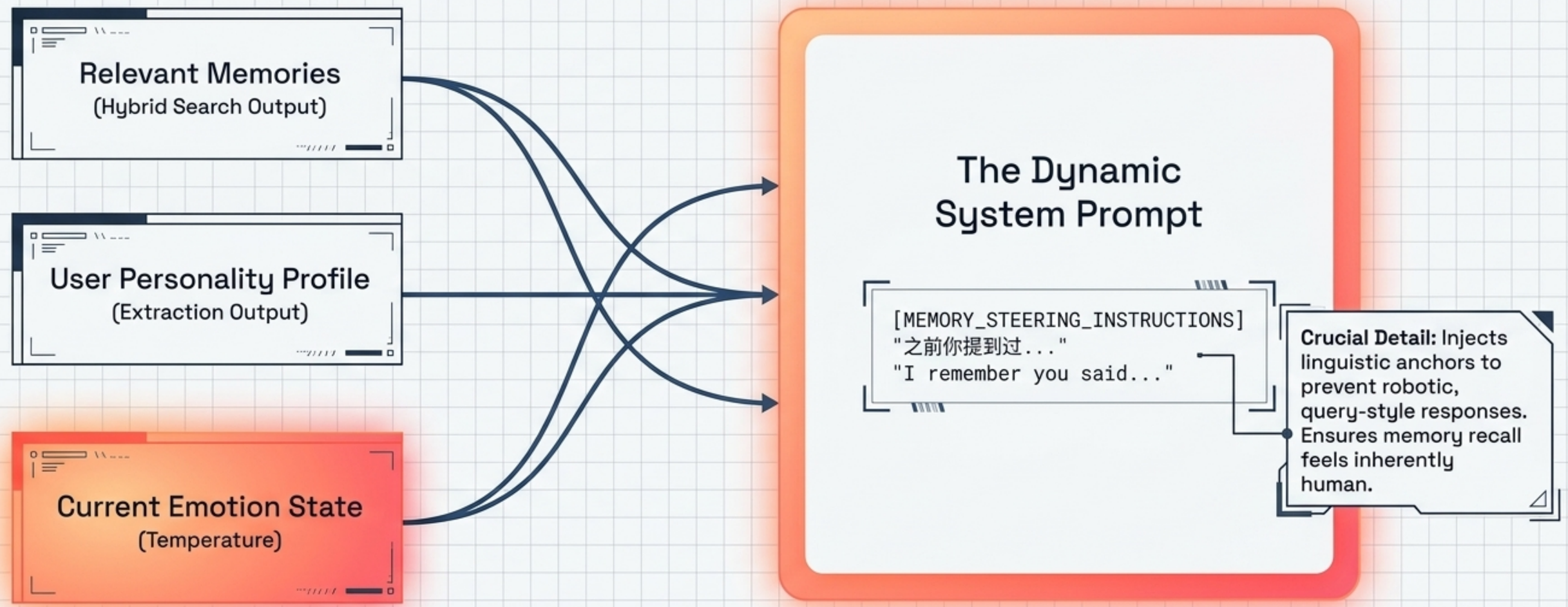
- Temperatures shift sequentially. No random jumps from Cold to Hot.
- Inertia prevents erratic mood swings. Two days of silence forces gradual decay to Cold.
- Higher inertia = Emotionally stable. Lower inertia = Highly reactive.

Designing the Personalities



Onboarding Constraint: Telegram's 64-byte callback limit required caching full custom text server-side and passing index references.

The Context Aggregator



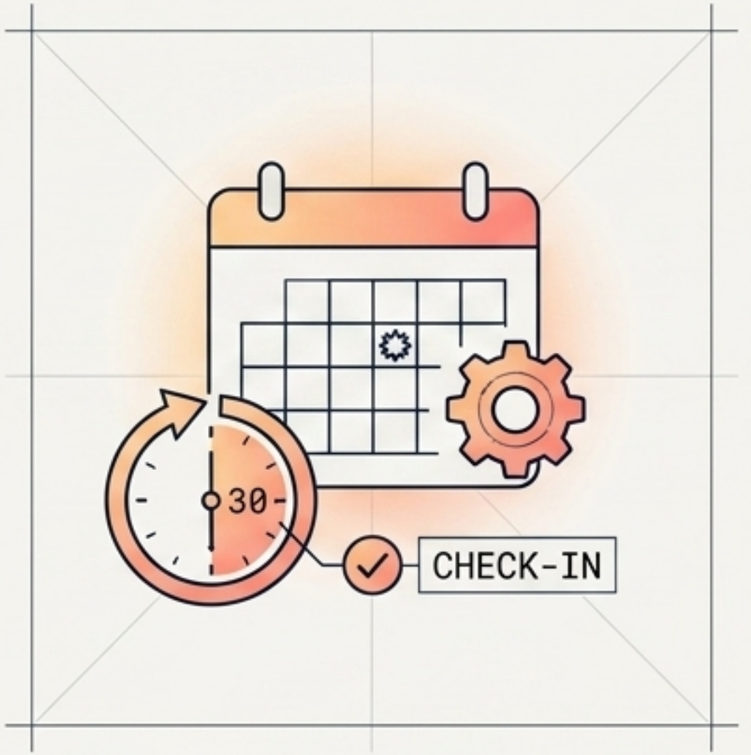
Engineering the "Human" Details

5-Second Debounce

Prevents the AI from answering prematurely. Collects rapid-fire user messages into a single, cohesive batch.

Algorithmic Typing Delays

Splits responses by newline. Applies 0.5s delay for short bursts, 2.0s for long paragraphs, firing the “typing...” indicator. AI speed feels unnatural otherwise.



Proactive Reach-Out

A 30-minute cron job checks for 2+ hours of silence. Warm users get contextual, LLM-generated check-ins. Capped at 3/day to respect quiet hours.

The Cost Ledger

AI companions generate invisible costs. **Track every penny** from Day 1.

Timestamp	Model_Name	Input_Tokens	Output_Tokens	USD_Cost
2023-10-27T10:30:15Z	GPT-4	512	1024	$0.06144
2023-10-27T10:35:00Z	Claude-2	128	256	$0.00448
2023-10-27T10:40:30Z	GPT-3.5-Turbo	1024	2048	$0.00614

```
def calculate_cost(model, input_t, output_t):
    cost = 0.0;
    if model == "GPT-4":
        ...
    log_transaction(timestamp, model, input_t, output_t, cost)
```

The NaN Bug

One missing model price config resulted in division by undefined, corrupting records.

Lesson: Hardcode a zero-fallback, but never delay implementing the ledger.

The v0.0.1 Scorecard

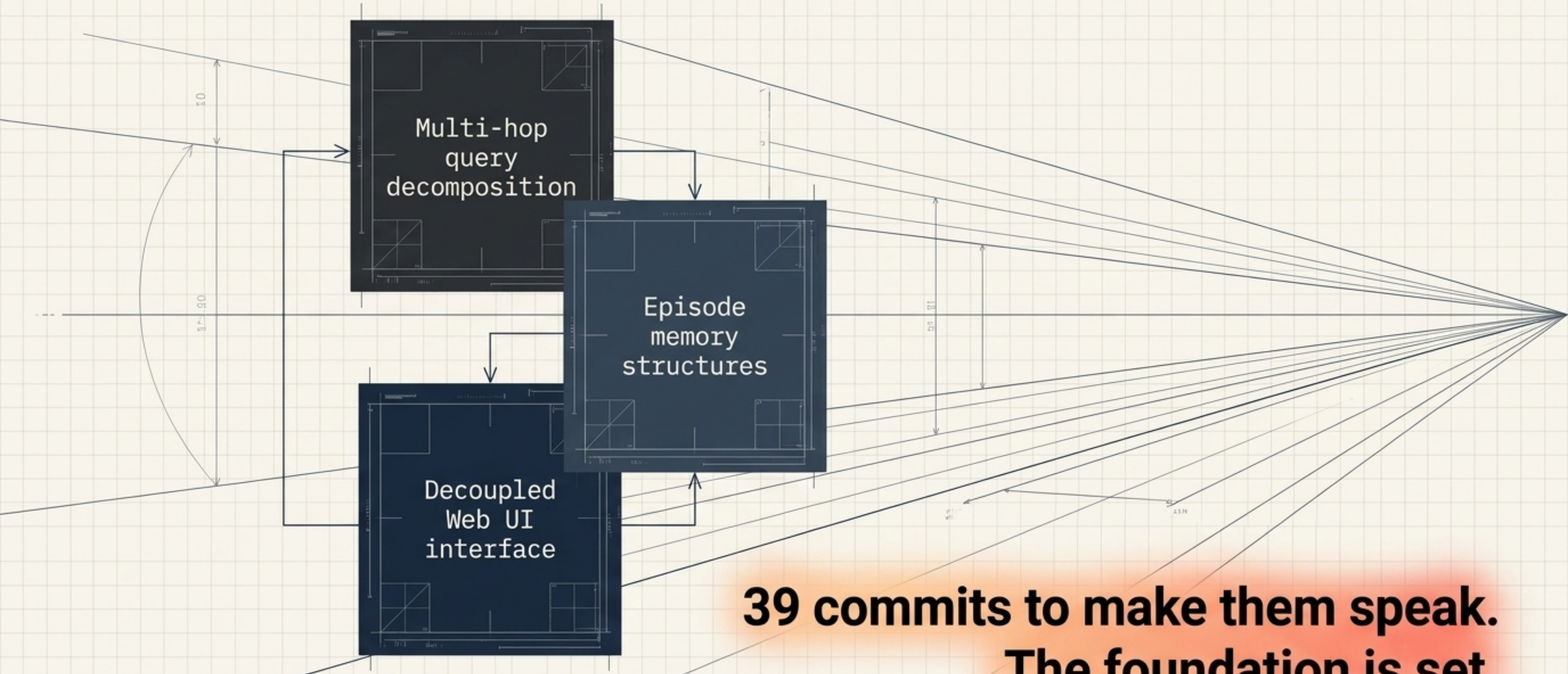
What Works (The Pulse)

- Multi-modal Telegram chat (text, photo, voice)
- Natural memory recall & personality profiling
- Emotional state machine & proactive reach-outs
- Precision cost tracking

What's Missing (The Gaps)

- Web UI (Telegram only for now)
- Voice output (can listen, cannot speak back)
- Advanced LLM reranking for memory
- Selfies capability

The Horizon: v0.0.2



39 commits to make them speak.
The foundation is set.