Omega: The Agent Layer

Turning AI agents into super agents.

What

Omega is agent infrastructure built on the Continuity stack.

- **Fleek**: "Turning AI models into supermodels" (inference)
- **Omega**: "Turning AI agents into super agents" (agents)

From the spec: *"I am stochastic omega. You are the oracle."* The droids propose, the oracle disposes. Obligations must resolve to omega.

Why

Current AI agents are brilliant with amnesia. Every conversation starts from zero. Every session is isolated. Models get smarter but never wiser.

The numbers:

Problem	Current State
Tool calling success (complex tasks)	30-40%
Multi-step workflow completion	20-30%
Knowledge retained across sessions	0%

OpenCode: 30K stars. Cline: 65K stars. OpenHands: 65K. OpenClaw: 100K+. The demand is real—developers want coding agents that work. The missing piece is persistent memory and reliable execution.

How

Three systems that make agents actually work:

1. Memory (CAS)

Content-addressed storage. Unlimited. Permanent. "*The result is saved*."

2. Obligations (Ledger)

Track every resource. Force resolution. "*If it doesn't resolve, it doesn't attest*."

3. Receipts (Attestation)

Cryptographic proof of execution. "*The attestation is machine-checkable*."

The Lean4 proofs slot in directly—the obligation algebra, the coeffect system, the prevention thesis. The proofs become the foundation, not marketing.

The Wedge: Flagship Forks

One-line installs. No config. Immediate value.

```bash
npx omega-code    # OpenCode + full stack
npx omega-claw    # OpenClaw + full stack
npx omega-sdk     # OpenAI Agent SDK + full stack
pipx run omega-aider  # Aider + memory
```

Three Flagships — Full Continuity Stack

We don't just add memory. We fork the hottest tools and rebuild them on the complete stack.

Flagship	Fork Of	Stars	Category
Omega Code	OpenCode	30K	CLI coding agent
Omega Claw	OpenClaw	100K+	CLI coding agent
Omega SDK	OpenAI Agent SDK	—	Agent framework

Each flagship gets the complete system:

- **Persistent Memory (CAS)** — Codebase understanding compounds. Day 1: learns your architecture. Day 30: knows patterns you forgot you wrote.

- **Obligation Tracking** — Every file opened gets closed. Every process spawned gets joined. Tool calls don't silently fail.
- **Attested Builds** — Cryptographic receipts for every action. Know exactly what the agent did and why.
- **Zeitschrift Integration** — Scope graphs for your codebase. References resolve or the build fails. No more hallucinated imports.
- **Verified Correctness** — Built on Lean4 proofs. 24 theorems, 9 axioms, 1 intentional sorry.

Then we expand with adapters for Cline, Aider, OpenHands, LangChain, n8n, and the rest.

Target ecosystem: 600K+ developers

- Coding agents: OpenClaw (100K+), Cline (65K), OpenHands (65K), OpenCode (30K), Aider (30K)
- Frameworks: LangChain (100K), n8n (60K), OpenAI SDK, CrewAI (25K)

The Fleek Integration

Omega works with any LLM. Omega + Fleek is native:

Capability	Any Provider	Omega + Fleek
Tool calling	~50-60%	~90%+
Cost	Baseline	-70%
Memory	Manual	Native
Setup	Config	Zero-config

Why Fleek?

- Enhanced tool calling API
- NVFP4 quantization (7× compression, same quality)
- S4 codegen (optimized CUDA kernels)
- Native Omega integration

The Name

Omega (ω) — The end state. Completion. Resolution.

From the spec. Mathematical weight. Where everything converges. Obligations resolve to omega or the take fails.

What We Need

1. **Ship the flagships** — Omega Code + Omega Claw weeks 1-2, Omega SDK week 3-4
2. **Integrate the proofs** — The Lean4 theorems are the foundation
3. **Side-by-side demos** — Same task: vanilla forgets context and leaves files open, Omega remembers and resolves
4. **Tell the story** — "the result is saved" as the throughline

---

The Bet

Everyone's building smarter coding agents. Cline, OpenCode, Aider—200K+ combined GitHub stars.

None of them have persistent memory. Every session starts from zero.

One command changes that:

```bash
npx omega-code
```

Your coding agent. But it never forgets your codebase—and it actually finishes what it starts.

That's the gap. That's Omega.

---

the result is saved

omega · fleek · 2026