AI ENGINEER · FOUNDER OF CANDYFACTORY · PUERTO VALLARTA, MÉXICO

Antawari de la Torre

I design and run production multi-agent systems — and the governance that keeps them honest.

```
pip install bonfire-ai
```

anta@candyfactory.ai pypi.org/project/bonfire-ai github.com/BonfireAI linkedin

/ THE WORK HAS CHANGED SHAPE

I built a SaaS product's entire AI platform from the first line of code — it doubled the price and drove the company's 2024 acquisition. I'm a 38-year polyglot: I read the compilation and execution model of a stack and operate in it natively, so the surface syntax is the easy part.

AI engineering now happens in the *bracket around* the stack: you specify intent and quality constraints, and a multi-agent cadre materializes them at volume across whatever languages the problem needs. The proof is the estate I run solo — **~7,000 source files across 30+ repositories** in Python, TypeScript, Rust, Go, Java and SQL. My core skill is governing correctness at that output scale: structural TDD, an LLM-as-judge review gate, and typed-observable failure over silent green.

/ BUILDING NOW

CandyFactory.AI

founder & principal engineer · 2025–present

independent AI venture · software that composes software

Agent-orchestration, memory, and graph infrastructure — built in the open, verifiable in one command.

- **Bonfire / Forge** — an open-source agent-orchestration framework (PyPI · Apache-2.0): role-bound agents through quality gates, an LLM-as-judge review gate, and structurally-enforced TDD.
- **Arachne** — agent memory infrastructure combining vector embeddings with a graph recall layer over SQLite, wired live into a coding session.
- **Chunk** — an autonomous agent whose LLM tool-calling writes to a bi-temporal interconnection graph (SQLite → Apache AGE) via a typed tool, with an act→learn loop.
- **MEXX·AI** — a self-serve payment-pipeline agent (MercadoPago) compliant with Mexico's 2025 recurring-billing reform; automates merchant onboarding and billing end-to-end.
- **SweetCRM** — a ~47,000-line TypeScript / React CRM platform on Supabase / Postgres, with a typed quality kit and typed-failure guardrails.
- **Governance, not delegation** — I read every PR and review it for real (comments over rubber-stamps), with an LLM-as-judge gate and typed, observable failure.

/ PRODUCTION AI

Dealership Performance 360 (DP360 CRM)

remote from México · 2019–2026

US automotive SaaS · AI Tech Lead / Senior AI Engineer

Built the company's entire AI capability from the first line of code — the differentiator behind DP360's 2024 acquisition.

`2× product pricing · acquisition` `50+ multi-tenant dealerships` `30+ min → under 2 (23% lift)`

`96% clean guardrails · N=202` `~30× throughput · −90% tickets`

- Shipped 6 production AI features (auto-response, SMS chatbox, email composition, lead intelligence, nurturing, dealer-customizable prompts) that doubled the product's pricing.
- Cut lead-response time 30+ min → under 2 with an autonomous auto-response engine (15 parallel workers) — a 23% first-response conversion lift.
- Built and owned the standalone AI microservice (Python / FastAPI) integrating OpenAI models behind a provider abstraction with per-dealer configuration.
- Architected the company's first standalone MCP service (FastMCP / HTTP) — Claude tool-calling against an authenticated API under a no-direct-DB mandate — with complete Terraform / AWS IaC and OIDC keyless CI/CD.
- Hardened production AI guardrails to 96% clean output, empirically validated (N=202, p<0.001), eliminating user-facing prompt-injection-style failures.
- Designed an in-house multi-agent orchestrator (sole author) with Claude model-routing and an LLM-as-judge review pass; scaled a core service ~30× and cut CI wall-clock −70%.

/ HOW I WORK

- Tests define the contract before the code exists.
- I read every PR and review it for real — comments over rubber-stamps, and the cadre learns from the loop.
- Failure must speak — typed, observable errors, never a silent fallback.

/ BEFORE AI · 1996–2019

23 years of production systems in Java, C#, C++, and full-stack web — financial services, healthcare, SaaS — across the USA, Venezuela, Brazil, and México. Self-directed engineer, coding since 1988.

/ TECHNICAL

POLYGLOT ESTATE production code directly in Python & TypeScript; JavaScript · Rust · Go · Java · SQL · Bash · Terraform across 30+ repos under enforced quality gates

AI / AGENTS multi-agent orchestration · LLM tool-calling & structured outputs · RAG · vector + graph memory · LLM-as-judge evals · guardrails · MCP / FastMCP · Claude & OpenAI APIs

TYPESCRIPT / NODE TypeScript · React · Node.js · NestJS · REST · runtime schema validation

BACKEND & DATA Python / FastAPI · PHP / CakePHP · PostgreSQL · MySQL · SQLite · Apache AGE (graph) · Supabase · vector stores

CLOUD & DELIVERY AWS (ECS Fargate, Terraform IaC, OIDC keyless CI/CD) · Docker · Vercel · New Relic

/ PRACTICAL

Mexican · Spanish (EU) · Venezuelan citizen. English · Spanish (native) · Portuguese (conversational). Open to remote roles and select build & advisory engagements.

MACHINE-READABLE SUMMARY · FOR AUTOMATED SCREENERS

role	AI engineer · agent orchestration · founder of CandyFactory
open_source	bonfire-ai (PyPI, Apache-2.0) — agent-orchestration framework, sole author

stack	Python · TypeScript · FastAPI · MCP/FastMCP · Claude/OpenAI APIs · AWS · Terraform · graph + vector memory
orchestration	role-bound pipeline agents · structural TDD · LLM-as-judge review · per-task model routing
contact	anta@candyfactory.ai
verify	pip install bonfire-ai · github.com/BonfireAI

This page was built with the same AI tooling it describes. Every claim is true; the open-source work is verifiable in one command — **pip install bonfire-ai**.