Antawari de la Torre

SENIOR AI ENGINEER · MULTI-AGENT & LLM INFRASTRUCTURE

Puerto Vallarta, México · Fully remote

antawari@gmail.com · github.com/BonfireAI/bonfire · pypi.org/project/bonfire-ai · linkedin.com/in/antawari-de-la-torre-a65a0843

I ship **production AI against real operational bottlenecks — not prototypes.** 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), and I keep that edge sharp building agent-orchestration and memory-graph infrastructure in the open. I partner directly with non-technical business owners and drive a multi-model agent cadre from Claude Code / Codex every day.

A 38-year **polyglot** — I read the compilation and execution model of a stack and operate in it natively; right now **~47,000 lines of production TypeScript / React**, alongside Python and a Node.js / Java / C# backend history. **The work has changed shape:** 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. The proof is the output: **~7,000 source files across 30+ repositories** (Python · TypeScript · Rust · Go · Java · SQL), one engineer — governed for correctness via structural TDD, an LLM-as-judge review gate, and typed-observable failure.

/ METHOD — ENFORCEMENT OVER PROMPTING

Control lives in the system, not the prompt string. I've moved **beyond prompt-tuning** to **model enforcement:** quality **laws** and **ADRs** as durable, versioned constraints, structural **TDD** and an **LLM-as-judge** review gate, and **graph-grounded context** over context-stuffing — behavior is enforced at the boundary, *typed and observable, not coaxed.*

/ PRODUCTION AI

Dealership Performance 360 (DP360 CRM) · US Automotive SaaS

AI Tech Lead / Senior AI Engineer · Remote (México) · 2019–2026

- **Autonomous back-office automation:** built an autonomous lead auto-response engine (15 parallel workers + smart follow-up scheduling) that cut lead-response **30+ min → under 2 min** — a **23% first-response conversion lift** across **50+ multi-tenant dealerships.**
- **6 production AI features** shipped to real users — autonomous auto-response, SMS chatbox, email composition, lead intelligence, lead nurturing, dealer-customizable prompts — together they **doubled the product's pricing** and drove the **2024 acquisition.**
- **LLM tool-calling architecture:** built the company's first standalone **MCP service** (FastMCP HTTP) — Claude tool-calling against dealer data over an authenticated API (no-direct-DB) — from zero to **deployed on AWS in six weeks** with complete Terraform IaC and OIDC keyless CI/CD.
- **Production guardrails:** hardened AI output to **96% clean,** empirically validated (N=202, p<0.001) — eliminated user-facing prompt-injection-style failures.
- **Scale & reliability:** scaled a core service **26.7 → 936 RPS (~30×)** via query optimization (DB CPU 85% → 12%), cut AI-related support tickets **~90%.** Multi-agent orchestration with Claude model-routing (Opus / Sonnet / Haiku), LLM-as-judge review, self-healing infra. Stack: Node.js / NestJS · TypeScript · Python / FastAPI · MySQL · PostgreSQL · AWS.

/ INDEPENDENT AI ENGINEERING & OPEN SOURCE

Solo R&D — CandyFactory

Remote (México) · 2025–Present

- **Bonfire / Forge — agent-orchestration framework** · `pip install bonfire-ai` · Apache-2.0 · public on GitHub. Role-bound agents through quality gates, an **LLM-as-judge** review gate, structural TDD (tester writes failing tests; implementer can't edit them), provider-agnostic via runtime-checkable protocols — *beyond prompting, a real autonomous multi-agent workflow.*
- **Arachne — agent memory infrastructure** *(experimental):* a three-layer memory system combining **vector embeddings (nomic) with a graph recall layer over SQLite,** live-wired into a Claude Code session — plus experimental cross-language context-graphs (CuartaStratta, Seam Graph).
- **Chunk — autonomous sales agent** *(experimental):* LLM **tool-calling writes to a bi-temporal interconnection graph** (SQLite → Apache AGE) via a typed `move_lead` tool with an act→learn loop — autonomous back-office automation against a real CRM seam.
- **MEXX·AI — self-serve payment-pipeline agent:** a **MercadoPago** integration compliant with Mexico's 2025 recurring-billing reform and data-protection law — automates merchant onboarding/billing end-to-end, built straight from business + legal research, **no PM in the loop.**
- **TypeScript production — current, daily: SweetCRM,** a **~47,000-line TypeScript / React** CRM platform (442 modules) on Supabase /

Postgres — plus a `cf-gate-ts` quality kit (typed file-budget + `tsc` ratchets) and typed-failure guardrails.

/ STACK & EXPERIENCE

TypeScript / Node · 6+ yrs production — NestJS · React · DP360 + SweetCRM (~47k LOC)

Python / FastAPI · 6 yrs professional — *using since v1.2 (1995)* · AI microservice + agent infra

AI · multi-agent · LLM · 3+ yrs production — tool-calling & structured outputs · RAG · LLM-as-judge evals · MCP / FastMCP · Claude API · OpenAI API

Model enforcement — laws · ADRs · graph-grounded context · quality gates · typed-observable failure (my control layer, beyond prompt-tuning)

SQL · 6+ yrs — PostgreSQL · MySQL (80–95% query-time cuts) · SQLite · Apache AGE (graph) · Supabase

AWS · Terraform · Docker · 1 yr — ECS Fargate · IaC · OIDC keyless CI/CD · **Azure** (AD / SSO integration) · Vercel · New Relic APM

PHP / CakePHP · 4+ yrs — DP360 2019–22

Angular · production · Certified · **TypeScript** Certified

Java · C# · C++ · 1996–2019 (legacy) — financial services · healthcare · SaaS

Rust · Go · Bash · via orchestration — materialized & reviewed under my quality gates

Coding since **1988 (38 yrs).** Production hands-on in **Python & TypeScript;** the rest materialized and reviewed under my quality gates.

/ 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.**

/ EDUCATION & VERIFY

Self-directed engineering education — continuous since 1988 · Angular (Certified) · TypeScript (Certified). **Verify the open-source work in one command:** `pip install bonfire-ai` · github.com/BonfireAI/bonfire

/ PRACTICAL

Mexican · Spanish (EU) · Venezuelan citizen · fully remote · English (native/bilingual) · Spanish (native) · Portuguese (conversational)