

Brody Kretz

AI Agent Engineering | Multi-Agent Systems | LLM Tool Use & MCP
brodyeckretz@gmail.com | +1 561-818-8486 | Boulder, CO | brodykretz.com | [LinkedIn](#) | [GitHub](#)

Technical Skills

AI Agents & Multi-Agent Systems

- **Claude API** (Sonnet, Haiku, tool use), **LangGraph**, LangChain — typed-state pipelines, conditional routing, feedback loops
- **Model Context Protocol (MCP)** — server design, JSON Schema, tool annotations, permission-callback patterns
- Adversarial quality gating, LLM-vs-deterministic split, cost-aware model routing, eval harnesses, event-bus design

Backend & Data

- **Python** 3.10+, **FastAPI**, **asyncio** + WebSockets, **pytest**, **APScheduler**; JWT auth, row-level security, REST API integration
- **Postgres** / **pgvector**, Supabase, **SQLite**, **SQL** + **BigQuery**; hybrid **RAG** retrieval, **pandas**, **scikit-learn**, **Matplotlib**, **Tableau**

Web & Delivery

- **TypeScript**, Next.js, React, Tailwind; PHP/WordPress (Abilities API), vanilla-JS WebSocket dashboards; Vercel, Fly.io, Stripe, Git

Experience

AI Engineer — Contract (Private Client)

2026

Sole engineer — Python, FastAPI, Supabase/Postgres, pgvector, Claude API

- Took a production AI backend from an empty repository to shipped: FastAPI service with JWT auth and row-level security over Supabase/Postgres, hybrid RAG retrieval on pgvector, and a three-tier persistent-memory layer that carries context across sessions.
- Hand-rolled the agent orchestration loop rather than adopting a framework, keeping control flow inspectable; added cost-aware model routing so classification runs on a cheap model and only generation escalates.
- Built an eval harness to prove that the cost routing did not degrade answer quality — measured rather than assumed.

Intern

May – Aug 2022

VIDSIG — Walnut Creek, CA

- Tested and troubleshooted the company's digital platform, documenting performance issues and reproduction steps for the engineering team.
- Supported creator outreach and onboarding, contributing to the platform's early roster and reach.

Projects

Pipeline Guard — Multi-Agent Self-Healing Data Pipeline

2026

Python, Claude tool use, FastAPI, APScheduler, pytest — open source

- Architected a six-agent self-healing ETL system: a Chaos agent fires one of seven adversarial sabotages while a Monitor → Diagnosis → Patch → Validator → Reporter chain detects, diagnoses, repairs, and logs each break autonomously.
- Made a deliberate LLM-vs-deterministic split — only Chaos and Diagnosis call Claude, where judgment is genuinely needed; the rest are deterministic, which is faster, cheaper, and reproducible at no cost to quality.
- Wrote a 79-test pytest harness with a scripted fake client covering the full healing chain, so CI runs with zero live API calls.

ChessReflex — Production Opening Trainer

2026

TypeScript, Next.js, Supabase/Postgres, Stripe, Stockfish WASM — live at chessreflex.com

- Designed and shipped a full training product: a spaced-repetition scheduler, an engine-backed puzzle generator that mines a player's own games for the first real blunder, and a 15-course opening repertoire.
- Cut a 16-second main-thread freeze to imperceptible by finding an $O(n^2)$ parse in the move-tree builder; verified the fix against 1,383 real moves with zero output mismatches.

Chess Coach AI — Three-Agent LangGraph System with Quality Gate

2026

Python, LangGraph, Claude (Sonnet + Haiku), python-chess, SQLite

- Built an Analyst → Coach → Critic pipeline with adversarial quality gating: the Critic scores Coach output on four weighted criteria and anything below 70/100 regenerates, with the Critic's objection injected into the retry prompt.
- Ran two Claude Haiku calls per move concurrently via `asyncio.gather` — one evaluating the player, one generating the reply — to hold live-play latency down.

wp-claude-mcp — Production MCP Server for WordPress

2026

PHP, WordPress 6.9 Abilities API, MCP, JSON Schema — MIT licensed

- Designed and shipped a 20-ability MCP plugin giving any MCP client permission-checked editorial control of posts, pages, media, CSS, and footer JS; ran it in production on my own site.
- Worked around WordPress's `wp_kses_post` script stripping with a footer-JS option-storage pattern, allowing site-wide interactive behavior without weakening the post sanitizer.

Education

B.S., Information Science (Minor: Business)

Aug 2022 – May 2026

University of Colorado Boulder — Boulder, CO

Relevant coursework: Applied Machine Learning, Fundamental Concepts in Data Science, Statistics for Information Science, Programming for Information Science I–II

Certification: **Google Data Analytics Professional Certificate** (2026)