Cade Snyder

AI Systems and Integration Engineer | Agent Harnesses, Retrieval, LLM Cost Governance

LeClaire, IA | 563-320-0235 | snydercade123@gmail.com | csnyder256.github.io | github.com/csnyder256

SUMMARY

AI integration engineer who builds the harness around the model: what it may touch, what it remembers, what it retrieves, and what it costs. Owns the AI surface of a live four-product B2B platform, routing GPT-4o, Anthropic Claude, and a self-hosted olmOCR model on vLLM behind per-tenant key routing, token quotas, and PII anonymization. Author of two published architecture specifications for long-running agent systems, one for a single operator and one for an organization, covering permission-enforced tool gating, memory authority, hybrid retrieval, context compaction, and budget governance. Full-stack delivery underneath: 10 backend services, 1,019 HTTP endpoints, 263 PostgreSQL tables, and an in-house Kafka-wire-compatible broker in Go. Public code carries 1,037 automated tests green in CI.

TECHNICAL SKILLS

AI and LLM Systems: OpenAI API, Anthropic Claude API, self-hosted LLM serving (vLLM), local ONNX embedding inference, retrieval-augmented generation, embeddings and semantic search, hybrid retrieval with reciprocal rank fusion, structured-output enforcement, prompt engineering, Model Context Protocol (MCP), per-tenant key routing, token quotas, PII anonymization (Presidio)

Agent Harness and Evaluation: Agent harness design, AI agents and agentic workflows, orchestrator and worker dispatch, permission hooks and tool gating, path fencing, context compaction, memory authority and supersedence, token budgeting and cost governance, evaluation harnesses, audit logging

Languages: Python, TypeScript, JavaScript, Go, SQL, HTML/CSS

Backend and Distributed Systems: FastAPI, Flask, Node.js, REST APIs, async SQLAlchemy 2.0, Pydantic v2, Alembic migrations, distributed systems, transactional outbox, Kafka-wire messaging, WebSockets, authentication, encryption

Data: PostgreSQL 16, row-level security, pgvector, SQLite, Prisma ORM, Redis, AWS S3, vector search, data pipelines, ETL, numpy, pandas

Frontend: React 19, Next.js App Router, Tailwind CSS, TipTap/ProseMirror, Yjs real-time collaboration, Zustand

Testing and Infrastructure: pytest, vitest, Playwright, Hypothesis property-based fuzzing, chaos testing, Docker, GitHub Actions CI/CD, Helm, Vercel, Railway

PROFESSIONAL EXPERIENCE

Clarus Platform (CLM, CFO, HR, Nexus) Mar 2026 to Present

Software Engineer, AI integration and platform | clarusclm.com

- Own the model integration layer across four live B2B products: GPT-4o, Anthropic Claude, and a self-hosted olmOCR model served on vLLM for document extraction and classification, behind per-tenant API key routing, per-tenant token quotas, and Presidio-based PII anonymization.
- Built contract intelligence that parses raw OOXML and PDF into a ProseMirror representation, types clauses against a 42-type taxonomy across 11 categories, scores per-clause risk, and overlays amendments as new versions rather than overwriting, with hybrid semantic and keyword search over pgvector embeddings and real-time co-editing over Yjs.
- Architected the production monorepo: 10 backend services, 1,019 HTTP endpoints, 263 tables, 203 Alembic migrations, and per-request tenant isolation enforced by PostgreSQL row-level security.
- Wrote ClarusStream, an in-house Kafka-wire-compatible broker in Go with consumer groups, SASL, ACLs, and S3 segment tiering. Producers publish through a transactional outbox drained by a relay service with retries and exponential backoff.
- Shipped OAuth integrations with nine third-party platforms plus Stripe billing, verified by an adversarial chaos-testing harness running nightly and weekly burn-ins in CI.

Per Mar Security Services Davenport, IA | Mar 2023 to Nov 2025

Security Officer, campus operations for Palmer College of Chiropractic

- Access control, incident response, and activity reporting. Earlier roles in auto detailing and car wash supervision, 2019 to 2023.

PUBLISHED ARCHITECTURE (github.com/csnyder256)

RAG-OS MIT

Blueprint and runnable starter for a self-hosted, always-on personal AI operating system

- A zero-context supervisor kernel is the only writer of the operational database while all intelligence runs in short-lived spawned sessions, so a crash, restart, or model downgrade loses no durable state. An orchestrator decides and delegates; ephemeral headless workers do the repository edits.
- Security is enforced in code rather than in prompts, re-checked on every tool call, failing closed and audited. Path fences normalize against traversal, symlinks, junctions, home aliases, and UNC prefixes. Dangerous actions sit behind an out-of-band single-use approval code.
- Capability-based model registry keeps model names in exactly one config file, enforced by a check that fails if a name appears anywhere else, so switching vendors is a config edit. A budget governor meters spend before a session spawns and degrades, then parks, in stages.
- Compiled knowledge as git-tracked Markdown with index-first retrieval and a rebuildable BM25 plus vector index fused by reciprocal rank fusion, embedded locally on CPU through ONNX. Superseded facts are invalidated rather than deleted so the base does not rot.

org-memory-os MIT

Organization-wide shared, permission-aware, auditable AI memory with one compaction discipline

- 13 architectural pillars, 70 decision forks each carrying options and a recommended default, 38 documented failure modes, and a 16-milestone build order where every milestone ends in a definition of done proven by a real command and its output.
- Names which single-operator invariants break at organizational scale and what replaces each: single-writer state becomes one writer per aggregate with a fencing token; identity as a boolean becomes an identity graph with fresh permission checks on the candidate set; invalidate-never-delete becomes wrong for personal data once erasure applies.
- Permission-aware retrieval filters the candidate set as the asking human through relationship-based access control with forced-fresh consistency, never filtering the generated answer. A nightly cross-tenant isolation canary is a hard release gate.
- Governance in the schema: four retention classes, per-subject crypto-shredding, a legal-hold table consulted before any deletion, and an immutable audit tier with citation-level logging.

SELECTED PROJECTS

ux-struggle-detector 165 tests, CI green

TypeScript, Next.js 15, React, Prisma, PostgreSQL, Playwright

- Multi-tenant SaaS that maps a customer web app by Babel AST parsing of its GitHub repo, then runs 40 struggle-detection rules server-side over hydrated cross-request session history and returns interventions in the same HTTP response the events arrived in. Dependency-free browser SDK, client-side PII scrubbing, AES-GCM encrypted key storage.

shadow-options-trading-lab and option-contract-grader 726 and 49 tests, CI green

Python, numpy, pandas, FastAPI, pytest, Hypothesis

- Runs 20 options strategies against live market data in shadow mode placing zero orders, graded with anytime-valid e-processes (test martingales, Ville's inequality) so nightly evaluation stays statistically valid. A subprocess-isolated test proves no order-placement code is reachable from the live path. The companion grader sweeps a 6,000-name universe, solving implied volatility and Greeks from Black-Scholes-Merton locally rather than trusting vendor values.

CERTIFICATIONS AND AFFILIATIONS

Anthropic Skilljar Certified

Technical advisor (informal, unpaid) to Varde Labs, an AI-integration firm