Tyler Brown

Applied AI & Systems Engineer

Henderson, Nevada · Tyler@innovativebiosci.com · (303) 941-9657 · linkedin.com/in/atylerbrown · atylerbrown.pages.dev

I build AI systems that run in production, and I measure whether they actually worked.

I do applied AI inside a commercial operation, which means the systems have to work on Monday: local model inference, retrieval, agent orchestration, and the evaluation harnesses that decide whether any of it ships. **614,000 lines** of JavaScript, Python and shell across twelve production systems, **824 commits** over seven repositories, and **553 test files carrying 5,836 test cases**. Fixes outnumber features 224 to 206. Where a result did not survive re-measurement I withdrew it in writing, and two of the entries below exist because of that. Full case studies at atylerbrown.pages.dev.

AI SYSTEMS, BUILT AND MEASURED

Literature extraction on local GPU inference. Reads the Methods section of published papers and extracts structured attributions, on a 14-billion-parameter model running locally through Ollama. The model must return a verbatim source span that is checked in code as a literal substring, so anything not present in the text is discarded rather than reported. A structural hallucination filter, not a prompt instruction.

6,537 papers in one 11-hour unattended run, 0 failures · throughput engineered 1.27 → 9.43 papers/min, ~3.3× of it from a quantization that fits in 12 GB VRAM · P 1.000 / R 0.813 / F1 0.897 on a 24-paper held-out set labelled from source before any model output was seen · truncation repair recovered 80 lost papers, then 0 failures across the next 8,000 · $0 marginal cost

An evaluation-contamination catch on my own work. A prompt revision measured as a large quality gain. I found the eval set had been assembled out of the previous prompt's own failures, with worked examples lifted verbatim from papers inside it. Re-measured on randomly sampled blind-labelled papers, reported the tie, kept the revision on other grounds, and wrote the rule down: an eval set built from a model's failures will always flatter the fix aimed at them.

claimed F1 0.927 vs 0.818 · on random blind labels 21/24 · 21/24, no measurable gap · withdrawn

Retrieval platform, and the benchmark that caught my own inflated number. Company-wide semantic search over contracts, product data and operational history, used by people and agents. Pinecone, Gemini embeddings, MongoDB semantic cache and query audit log, contextual chunking. I then built a falsifiable benchmark across every knowledge layer and found my own earlier headline figure was inflated by answer leakage in the test harness, and voided it.

~27,000 vectors / 40 namespaces · warm latency 2.7–4.3 s → 280–400 ms · corrected result: cross-layer fan-out answers 0.81 of questions against 0.42 for the vector store alone

Nightly evaluation harness and threshold recalibration. Automated retrieval-quality regression testing with a model as grader. Investigating a long-running red state established that none of 100 stored runs had ever passed all three gates: the thresholds were stock values never calibrated to this corpus. Recalibrated against a full replay. Separately traced a reported 22-run failure streak to an artifact rather than a regression, because the alert log only recorded violations, so a passing run wrote nothing and a streak could only grow.

0 of 100 runs passing → 67 of 100, every remaining failure a genuine collapse · four stacked grader defects fixed, error cases 30 → 0

An agent fleet, and a written protocol for letting agents do risky work. 39 specialised agent definitions over a seven-stage protocol: issue specification, contract negotiation, review by an evaluator running in a separate context, an isolated worktree, continuous scope-drift monitoring during the build, gates, and a trace review able to amend the protocol itself. Dogfooded honestly: on first run the protocol's own evaluator returned amendment-required against its designer, and an adversarial meta-review found ten blocking design flaws.

protocol 1,005 lines, 279 tests · 21 integrated tool servers · 80 scheduled jobs

Tool servers where the risk model is the product. A Model Context Protocol server exposing a CRM to agents across 35 tools, in which every call is risk-classified and the classification is the control: outbound sending and billing refused outright, deletes and writes gated behind a filesystem stop sentinel, social tools forced to draft regardless of caller input. A public catalogue server built alongside it was hardened across eight adversarial review cycles, including a Cyrillic homoglyph bypass found in the outbound content filter.

35 tools, send and billing refused outright · catalogue server 299/299 tests across 12 suites including bypass-attack and red-team suites

Generative media with a control layer. The hard part is the hundredth frame matching the first, and the label still saying what the label says. A measured quality gate (border colour and texture carried no signal; floor-band darkness did) and a label-integrity guard that warps original pixels onto the generated object and fails closed rather than shipping something plausible.

discriminator 52 against 90–159 at the 5th percentile · guard fails closed, 2 of 3 generations rejected · documented limits, including a one-digit phone-number drift at temperature zero

THE ENGINEERING AROUND THE AI

cXML and OCI e-procurement connector. Institutional buyers purchase from inside their own purchasing platform. Seven message types plus SAP OCI against vendored DTDs, durable outbound queue with at-least-once delivery, full-jitter backoff, strict document ordering and dead-lettering, atomic idempotency on order intake, and two deliberately split health endpoints.

26,396 lines · 153 files · 94 test files · 9 adversarial security reviews closed 40+ findings, including an unauthenticated NoSQL operator injection to session hijack on the one service that creates real orders · live

Marketplace order connector. Signed CloudEvents over HMAC-SHA256. Verifies fail-closed, persists the raw signed event before acting on it, acknowledges inside a three-second budget, handles money as integer cents through BigInt, and runs idempotency at both event and effect level so at-least-once redelivery can never create a second order.

244 tests across 25 suites · two silent order-loss defects caught pre-launch, one of which would have rejected 100% of real orders while returning a plausible response

A seventeen-layer pre-send guard. A gate answering whether an address may be contacted right now, called by every drafting pipeline. Fail-open and fail-closed posture assigned per layer as a deliberate asymmetry, with two independent halts.

172 sends, 0 bounces · the controlling finding came from audit, not code: all 25 historical hard bounces were unverified addresses

Extraction at scale. Browser automation with fingerprint rotation, per-domain adaptive backoff behind a circuit breaker, and eight confidence-ranked extraction strategies with de-obfuscation for HTML entities, reversed strings, ROT13 and JavaScript concatenation.

48,810 lines · 1,014 addresses from 1,198 pages, 0 duplicates, 0 provenance gaps · found 92% of guessed addresses wrong where the institution issues opaque IDs

> **What this is not.** I build applied AI systems and the production engineering around them. I have not trained foundation models, written CUDA kernels, built distributed training infrastructure, or done ML research. If a role needs that, I am the wrong candidate and would rather say so here than in the third interview. Much of the implementation volume above was produced with AI coding agents under a protocol I wrote for that purpose; the engineering that matters is deciding what they may touch, what has to fail closed, and which of their results survive an independent measurement.

EXPERIENCE

Sales Manager, Innovative Bioscience

JANUARY 2024 – PRESENT · LAS VEGAS VALLEY, NEVADA

Carry the commercial number for cell culture reagents and laboratory consumables into research universities, cancer centres and biotechnology companies, and built every system above inside that role. Designed and deployed the e-procurement integration, consolidated CRM, quoting, pricing and accounting into one reporting layer, built the quoting and price sheet engines the team uses daily, and administer the Microsoft 365 tenant and identity estate. The systems have a user who complains directly to me, which is a useful constraint.

President, SunNSee.com

JUNE 2019 – NOVEMBER 2023 · UNITED STATES

Founded and ran an e-commerce and equipment business serving ophthalmic practices nationwide. Owned the storefront, supplier relationships and fulfilment, produced all marketing content including product photography, video and aerial drone work, and built an inventory tracking system against my own profit and loss. That was the first commercial software I wrote, and the reason I build systems rather than buy them.

Regional Sales Manager, Lombart Instrument (Advancing Eyecare)

OCTOBER 2018 – AUGUST 2020 · NEVADA AND UTAH

Managed a $1.8 million annual pipeline of capital diagnostic and surgical ophthalmic systems. Ran the buying process end to end: return-on-investment modelling, leasing, and multi-stakeholder approval across surgeons, operating-room staff, administrators and purchasing. Named Top Regional Sales Performer in 2019. Earlier: ophthalmic technician and dispensing optician in Las Vegas, 2011 to 2017.

TECHNICAL

Models	Qwen3-14B · Qwen3-8B · Gemini 2.5 Flash · Gemini 3 Pro Image · Claude · WhisperX large-v3 · Veo 3.1
AI stack	Ollama · GGUF quantization · Pinecone · RAG · MCP servers · LLM-as-judge · span grounding · agent orchestration · eval harness design
Build	Node.js · Python · Bash · SQL · Docker · Jest · node:test · Playwright · Puppeteer
Data	MongoDB · SQLite · PostgreSQL · Litestream · Cloudflare R2 · Common Crawl · NCBI E-utilities
Infra	Cloudflare Workers, Access and Pages · Railway · Tailscale · OAuth 2.0 · JWT and JWKS · queues and dead-lettering
Protocol	cXML 1.2 · SAP OCI · CloudEvents · HMAC-SHA256 · GraphQL · REST · G-code
Practice	Test-driven development · adversarial and red-team review · measurement and evaluation design · fail-closed safety architecture

EDUCATION

College of Southern Nevada · Associate of Science, Ophthalmic Laboratory Technology · 2011 – 2013

Front Range Community College · Applied Science, Mechanical Engineering · 2009 – 2011