

BRYAN VINE

AI Engineer | MLOps | AI Infrastructure Engineering

Port Angeles, WA | bryan@bryanvine.com | showcase.vineai.tech | linkedin.com/in/bryanvine | github.com/bryanvine

PROFESSIONAL SUMMARY

AI infrastructure and MLOps engineer with 15+ years designing, automating, and operating large-scale distributed systems in production at enterprise scale, now focused on LLM platform engineering: GPU inference clusters (vLLM, CUDA, custom Triton kernels, KV-cache quantization), model serving and gateway routing with automated failover, RAG pipelines, CI/CD and infrastructure-as-code, and full-stack observability. Delivered \$15M+ in annual savings leading automation across a 200,000+ server fleet. Full-stack development of self-hosted, multi-tenant AI platforms end to end, from GPU kernels to deployed React/FastAPI product. Serial entrepreneur: cofounder, CTO, or principal architect across six technology startups and four consulting firms, with technical sales experience and patents and trademarks held. MS in Computer Science (Machine Learning) in progress at Georgia Tech.

TECHNICAL SKILLS

LLM Inference & Serving: vLLM, CUDA C/C++, Triton kernels, KV-cache quantization, GPTQ/AWQ, speculative decoding (EAGLE3), MoE expert pruning, tensor/pipeline parallelism, LiteLLM, Open WebUI, Ollama, OpenAI-compatible APIs, multi-GPU orchestration

Machine Learning: PyTorch, TensorFlow, Transformers, QLoRA fine-tuning, knowledge distillation, XGBoost, scikit-learn, MLflow, RAG, embeddings, vector search, OCR pipelines, LSTM, model backtesting & walk-forward validation

MLOps & DevOps: Docker, Docker Compose, Kubernetes, Podman, Ansible, Terraform, Puppet, HPSA, Jenkins, Bamboo, GitHub Actions, ArgoCD, Artifactory, Vagrant, CI/CD, GitOps, Prometheus, Grafana, OpenTelemetry, Nginx

Cloud & Infrastructure: AWS, GCP, Azure (Container Apps, AI Search, Entra ID, Bicep), Vast.ai GPU cloud, VMware, Hyper-V, Xen, OpenStack, Cloudflare Workers, KV & Tunnels, Linux, Windows Server & Active Directory, NetApp, F5, Cisco, distributed systems & high-availability design

Full-Stack Development: React, Next.js, FastAPI, REST & SSE APIs, multi-tenant SaaS architecture, Clerk/Entra ID auth, Playwright, PWAs & service workers, Nginx

Languages: Python, TypeScript/JavaScript, Java, Go, SQL, Bash, PowerShell, C++ (SYCL)

Data: PostgreSQL, TimescaleDB, Redis, ChromaDB, Azure AI Search, SQL Server, Neo4j, Splunk, vector databases

PROFESSIONAL EXPERIENCE

Vine AI | Port Angeles, WA | Jan 2026 – Present

Founder & Chief Architect

Build, deploy, and operate production AI infrastructure and LLM applications for clients and consumer products on self-hosted multi-GPU compute and cloud.

- Operate multi-GPU LLM inference infrastructure serving 30B–120B-parameter models with vLLM across NVIDIA (RTX PRO 6000 96GB, RTX 5090 32GB) and Intel XPU (Arc Pro B70 32GB) hardware, with hot-swappable model routing, health monitoring, and automated lifecycle management
- Ported TurboQuant KV-cache quantization (ICLR 2026) to Intel Battlemage GPUs for vLLM, writing custom Triton kernels that achieve 4–8× KV-cache compression via Walsh-Hadamard rotation and Lloyd-Max quantization; contributed upstream to vllm-xpu-kernels
- Implemented speculative decoding pipelines (suffix and EAGLE3 draft models) with fused N-spec Triton kernels, autotuning, and causal masking for 1.5–2× decode throughput
- Deployed self-hosted LLM gateway (LiteLLM, Open WebUI, PostgreSQL) unifying multiple vLLM backends behind one OpenAI-compatible API, with ordered-fallback routing, automatic failover and cooldown, per-user usage analytics, and spend tracking
- Delivered AI document intelligence platform for a private client: OCR ingestion with automated PII redaction, semantic vector search, and agentic report generation; multi-tenant FastAPI/React stack with RBAC, audit logging, OpenTelemetry tracing, and Azure deployment (Container Apps, AI Search, Entra ID) via Bicep IaC
- Architected on-premises AI credit intelligence platform for B2B lending: dual-GPU inference routed by task complexity (80B orchestrator + 14–30B subagent models), 30+ tool underwriter agent with map-reduce pipeline, and multilingual customer chatbot, with zero cloud LLM dependencies
- Rebuilt a client’s public marketing site off a hosted website builder onto a self-hosted stack: design and copy reconstructed from the live site into hand-written responsive HTML/CSS, imagery re-encoded to WebP with sized variants, served in preview by containerized Nginx behind a Cloudflare tunnel with idempotent scripted DNS and ingress provisioning, and in production as static assets on managed static hosting, the site needing no server-side logic; and embedded a retrieval-grounded site assistant, grounded in the client’s own content and provisioned on my own chatbot platform, as a drop-in widget; subsequently rebuilt against the client’s own brand brief, holding booking calls-to-action as explicit “coming soon” states until the third-party reservation page goes live rather than shipping dead links, and re-platformed once the constraints were

known rather than on preference: the client will not move registrar and the incumbent builder will not delegate nameservers, which rules out any host needing its own nameservers to answer for an apex domain, so the nameserver migration was dropped for a host accepting apex A records and a www CNAME from the existing DNS panel: two records the client edits, no support ticket. The credit-metered alternative was rejected on failure mode, not capacity: its free cap is hard and swaps the site for an error page on exhaustion, where the chosen host's quota is soft and sends mail. Since free-tier static hosting cannot publish from a private repository, and the working repo carries the client's billing documents, a scripted publish step mirrors only the built site to a public one. Handed over with a client-facing runbook, rendered to a branded PDF, covering those DNS records and retiring the website-builder subscription (sequencing the bundled mailbox transfer to direct billing first, since cancelling it instead would expire the mailbox); the apex still serves from the builder until the client makes the change

- Architected Vine AI multi-tenant on-premises platform: Docker Compose orchestration across 15+ services, Nginx SSL gateway, Prometheus/Grafana monitoring, and automated lifecycle management (weekly updates, post-boot health checks, GPU utilization alerts)
- Developed GPU-accelerated quantitative trading systems: equities engine processing 500–2,000 securities at 5-second intervals (DeepLOB/Transformer/XGBoost ensemble inference, LLM news sentiment) and crypto system on Hyperliquid (LSTM and gradient-boosted models, walk-forward validation), with TimescaleDB telemetry

Level Up Technology | Woodland Hills, CA | Jun 2020 – Present

Lead Solutions Architect

Design and deploy enterprise infrastructure and AI automation across multi-cloud environments (AWS, GCP, Azure, VMware) for MSP clients.

- Architected and deployed containerized LLM inference environments for dozens of clients using Docker and Kubernetes, with Ansible-driven provisioning and configuration management
- Built and maintained CI/CD pipelines using Jenkins, Terraform, and GitHub Actions for infrastructure-as-code deployments across cloud and on-premise environments
- Led on-premise to colocation VMware cluster migration, designing compute resource allocation, networking, and disaster recovery with OpenStack and VMware orchestration
- Modernized legacy workloads through Kubernetes containerization with automated scaling, health monitoring, and rolling deployment strategies
- Ansible consulting and technical sales engineering: scoped, demonstrated, and closed automation engagements for MSP and enterprise clients

Wells Fargo | Santa Monica, CA | Nov 2012 – Sep 2024

Lead Infrastructure Engineer | **AVP** (May 2014 – Sep 2024)

Led infrastructure automation strategy and middleware distribution across a distributed 200,000+ server enterprise environment spanning multiple data centers.

- Delivered \$15M annual savings by designing and deploying a proprietary automation framework orchestrating software distribution, configuration, and lifecycle management across the full 200,000+ server fleet
- Deployed Grafana- and Prometheus-based observability platform with custom server agents across the entire infrastructure for unified monitoring, alerting, and capacity planning
- Built automated CI/CD testing pipeline with Ansible-driven validation that reduced deployment errors and accelerated release cycles across large-scale distributed systems
- Managed resource allocation and scheduling for middleware deployments across geographically distributed compute clusters with automated rollback and health verification

Systems Architect (Nov 2012 – May 2014)

- Designed scalable datacenter compute solutions across 2,000 servers in 4 data centers using VMware clusters, HA configurations, NetApp storage, and F5 load balancers with automated provisioning
- Created infrastructure QA testing framework with automated post-maintenance validation, reducing unplanned outages across production clusters

Live Nation | Hollywood, CA | May 2011 – Nov 2012

Senior Infrastructure Engineer | **Site Reliability Engineer**

Operated the distributed systems behind a global ticketing platform across 5,000 servers handling millions of concurrent users during peak on-sale events; subject matter expert for Kerberos authentication infrastructure and PCI-DSS compliance auditing.

25 by 7, Inc. | Santa Monica, CA | Nov 2007 – May 2011

Senior Infrastructure Engineer

Lead engineer and consultant for an outsourced IT firm serving 50+ partner clients.

- Delivered projects spanning web hosting, colocated Exchange/Lync, back-office call centers (messaging, file/print, VoIP), financial transaction processing with rights management and field-laptop protection, and scientific research computing (data aggregation, warehousing, distributed compute)

- Engineered Windows AD, Exchange, VMware, Hyper-V, IIS, SQL Server, SharePoint, Rights Management Services, Forefront UAG, Cisco, SonicWall, Linux, HP, ShoreTel, EMC, F5, NetScaler, and Foundry environments

ENTREPRENEURSHIP

Cofounder, CTO, or principal architect across six technology startups and four consulting firms, in addition to Vine AI and Level Up Technology above. Hands-on ownership of product architecture, cloud infrastructure, custom hardware integration, and technical sales from first customer to exit or wind-down.

- **Vine Ventures, LLC** | Santa Monica, CA | 2015 – 2021 | Principal Architect & Cofounder: software development and cloud infrastructure consulting firm
- **BudMe** | Denver, CO | 2013 – 2022 | CTO & Cofounder: social experiences network and cannabis point-of-sale and distribution management platform
- **PicDragon** | Santa Monica, CA | 2017 – 2018 | Principal Architect & Cofounder: dynamic services estimation and marketing platform
- **Metafore** | Melbourne, Australia | 2016 – 2018 | CTO & Cofounder: entertainment and lifestyle metadata marketing
- **Blutro** | Melbourne, Australia | 2015 – 2016 | Principal Architect: event ticketing management with custom Bluetooth hardware
- **AppCopia** | Santa Monica, CA | 2012 – 2014 | Cloud Architect: phone platform migration and marketing
- **Crypto Miner** | North Hollywood, CA | 2011 – 2015 | Principal Architect: pool cryptocurrency mining operator
- **Bryan Vine Consulting** | Chatsworth, CA | 2005 – 2015 | Consultant: infrastructure and IT consulting
- **RepairYourPC** | Temple City, CA | 1998 – 2004 | Founder & Computer Technician: PC repair business

SELECTED PROJECTS

Live demos and full portfolio: showcase.vineai.tech

- **vllmstat** (open source, Python): “nvtop for vLLM”, a zero-infra terminal dashboard scraping vLLM’s /metrics for throughput/latency percentiles, GPU (NVIDIA/AMD/Intel), KV-cache, and speculative-decoding telemetry across fleets | github.com/bryanvine/vllmstat
- **llamastat** (open source, Python): the llama.cpp counterpart, “nvtop for llama.cpp”: a terminal dashboard over llama-server’s read-only endpoints showing per-slot phase, task, and context-window usage live. Because it diffs the server’s own slot token counters between polls, it reports generation and prompt tokens/s on servers started without the Prometheus endpoint, which is most of them; the server-side metrics panel appears only when that endpoint exists and hides itself otherwise. Run on the inference box it adds host telemetry the server does not expose, so work outside the server (a llama-bench run, another process on the same GPU) is still visible: CPU utilization and load average, memory, hwmon temperatures grouped per chip, and package power derived by differencing RAPL energy counters, with a documented permission fallback when those counters are root-only. Degrades to whatever the server exposes rather than crashing when it is still loading the model or has slot introspection disabled. Apache-2.0, published to PyPI | github.com/bryanvine/llamastat
- **MLA-GPT** (open source, PyTorch): controlled study of attention variants, MLA (DeepSeek Multi-head Latent Attention) vs MHA/MQA/GQA in identical from-scratch 124M GPTs (RoPE, SwiGLU, RMSNorm); a weight-absorption decode path delivers 5.6× KV-cache compression vs MHA (359 vs 258 tok/s at 16k context) at near-zero perplexity cost | github.com/bryanvine/mla-gpt
- **gpt-oss-120b expert pruning** (open weights on Hugging Face): task-calibrated pruning of OpenAI’s 117B-parameter Mixture-of-Experts model from 128 to 48 experts per layer, taking the checkpoint from 65.2 GB to ~27 GB so it serves on a single 32 GB GPU at 128k context in stock vLLM, with no custom kernels, offload policy, or runtime patches. Expert saliency (router gate weight times expert output norm) is computed one decoder layer at a time on a single 16 GB consumer card, dequantizing MXFP4 expert weights on the fly, and is calibrated not on prompts alone but on the full model’s own generation traces. On a frozen 311-prompt replay of my job-search platform’s live scoring workload, the pruned model holds Cohen’s κ 0.51 on Intel Arc (XPU) and 0.54 on an RTX 5090 (CUDA) against stored production decisions, inside the unpruned model’s own 0.49–0.53 run-to-run self-consistency range, at 100% strict-JSON parse on both backends, against 0.46 for the 20B model of the same family. The prompt-only, routing-frequency ablation collapsed to 87% parse and κ 0.15, isolating generation-aware calibration as what makes a cut this deep survivable. Saliency reproduces across two independent GPUs at Spearman 0.997 with 99.2% identical keep-sets. Variable per-layer budgets tied the uniform cut within noise at higher memory: a recorded negative result. Served with EAGLE3 speculative decoding (39–44% draft acceptance), it ran two days as a production shadow on that platform’s live scoring path, re-scoring every call behind the unpruned model, logged for agreement and never changing the primary result: over 11,653 live pairs, mean absolute delta 0.107 on the headline semantic score (median 0.06), 93% binary location-gate agreement, confidence and risk drifting further (0.18, 0.23). A distilled 8B, the next entry, has since taken the shadow slot on better numbers. Effects outside the calibrated task are deliberately unmeasured | huggingface.co/bryanvine/gpt-oss-120b-reap-48, write-up
- **Task distillation of the job-scoring tier** (open source, follow-on to the pruning work above): rather than cut the 117B teacher further, teach a model that already fits the one task, from the teacher’s own stored production outputs. Round one reconstructed 15,810 PII-scrubbed pairs from the live scoring table against the exact production prompt, and QLoRA

fine-tuned Qwen3-8B (14.3 hours on one 16 GB RTX 4080 Super) to Cohen's κ 0.64 against stored production decisions, against 0.51/0.54 for the 48-expert pruned 120B. Agreement, though, was the wrong question: a gate replayed 241 user-voted matches through the student on the production path and scored it against the votes rather than the teacher, and it failed, keeping 91% of the jobs users explicitly downvoted against the teacher's 57%, on calibration rather than ranking. Round two rebuilt the set at 22,900 pairs, added a preference stage, and benched five arms against a pre-registered gate of 57% downvote retention. The planned build passed the number and failed the intent: three-epoch DPO reached 11% by scoring everything lower, rejecting 13 of 14 upvoted jobs at an 8% positive rate against the teacher's 45%. An unplanned supervised-only arm isolated the preference dose as the cause, so the gate was tightened rather than relaxed, adding upvote retention and positive rate. Supervised training alone clones the teacher's calibration and inherits its false positives (62%, a miss); the corrective retrain, one epoch with an NLL anchor on the chosen side and upvote pairs oversampled fourfold, passes at 35%, clearing 37 of the teacher's 97 false positives with upvote losses inside noise. The same data moved an 8B control from 91% to 68%, so the preference stage, not capacity, was the lever. It stays shadow-only: on live traffic it flags under half the teacher's positive rate, and whether the matches it suppresses are the bad ones is still open | github.com/bryanvine/fine-tuned-job-score-model, write-up

- **Task distillation of the resume-writing tier:** the scoring student above distills a classifier a vote-scored gate can settle; open-ended writing has no such instrument, and the instrument is the story. Teacher capture tagged all 25,178 calls with the model that actually served them, catching a bank a substitute had quietly filled during an outage. A rank-32 LoRA on Qwen3.8-27B over 9,327 guard-accepted completions, self-quantized to W4A16. A blind judge reversed on 32 of 48 order-flipped pairs, so parity stays unmeasured; promotion rested on the groundedness guard, which accepted 28 of 30 student resumes and all 28 cover letters. The hosted fallback was then removed, making writing local-only. The next relapse showed the cost: thirteen hours emitting nothing but token zero, unalarmed, because a wedged model still answers HTTP 200; the guard kept every bad draft from users, so the only symptom was daily documents falling from 351 to 14. A watchdog now probes content rather than status every ten minutes and restarts the writer. Diagnosis was harder: moving to the other physical card reproduced the corruption anyway, 36 of 36 generations garbage at twelve-wide decode, the int4 GEMM kernel ran clean in isolation at every batch size, and a byte-level image diff killed the build-regression theory. With hardware, kernel and build exonerated, he filed it upstream as [vllm-project/vllm#53480](https://vllm-project.github.io/vllm#53480) with the isolation matrix and wrote it up publicly. Upstream triage then named the cause: a memory race in the XPU kernels for the linear-attention layers that make up 48 of the model's 64, which is why the pure-attention scoring engines never saw it. The writer moved to the runtime release carrying that fix, and the repro that had corrupted 36 of 36 generations ran 318 requests clean at the same twelve-wide concurrency before promotion | write-up, reach.jobs
- **token-saver** (open source, MIT): a coding-agent cost kit built on measurement rather than folklore: auditing a week of my own Claude Code transcripts found 92.6% of input tokens were reused context served from the prompt cache, replicating a published billion-token analysis, while the command output that popular "compressor" hooks shrink is a rounding error (a vendor A/B of one such hook measured real billing *rising* 7.6% at low reasoning effort). The kit targets the levers that actually move the bill: a skill enforcing scoped reads, filtered command output and append-only context so the expensive window stays small and cacheable; a pack of cheap-model subagents (**scout**, **reader**, **runner**) plus a model-routing rule so search, summarization and test runs never bill at the top tier; and a dependency-free audit tool that prices the usage records the agent itself logs with cache-aware rates (cache reads $0.1\times$, cache writes $1.25\text{--}2\times$) rather than counting hypothetical savings; installed into any setup by one command, with an exact uninstall and a test suite, and every rule bounded by "never cause a retry," since one failed cheap attempt costs more than it saves. Now packaged as an installable agent plugin whose repository doubles as its own distribution channel, so adoption is two commands rather than a symlink script, and the rules are injected by a session-start hook instead of waiting to be invoked, placed at the front of the prompt where it is served from cache on every request after the first, making the enforcement mechanism itself nearly free | github.com/bryanvine/token-saver, write-up
- **Alpha Research** (9-paper cross-asset series) & **JEPA-Trader**: open, reproducible quantitative research for harvestable alpha; self-supervised learning (JEPA) under a fixed rigor protocol (deflated Sharpe, PBO, purged walk-forward); net-of-cost survivors combine via risk parity into a 0.53-Sharpe book now paper-traded forward live (alpha.vineai.tech); the latest paper's signal was surfaced by an autonomous LLM search loop gated by that same harness, which then auto-killed it out-of-sample | bryanvine.github.io/alpha-research
- **Trading System** (live paper trading on Interactive Brokers): automated multi-strategy stack over a TimescaleDB market-data spine: a 40-slot mean-reversion swing book across an ~ 894 -name universe alongside a GPU-accelerated ensemble engine (DeepLOB/Transformer/XGBoost on L2 depth); hardened by a pre-registered crisis-robustness program: a replay engine mirroring live execution semantics, block-bootstrap tail estimation and crisis replays on data backfilled to 1999 measured true path risk (-44% max drawdown in a COVID replay, -61% in a GFC replay) and selected 12% annualized volatility targeting (replayed max drawdown $-44\% \rightarrow -17\%$, Sharpe $0.83 \rightarrow 1.03$) over reactive de-grossing. A defined-risk options paper sleeve pairs LLM judgment with a code-enforced policy box: a deterministic screener feeds per-candidate research briefs and one selection call, every pick is validated in code against premium, per-name, sector, expiry and liquidity limits (violations dropped, never repaired by the model), and the cycle fails to a no-op rather than to a weaker model; first fills

came in July 2026 against a mechanical LEAPS-ladder control judged at 12 weeks under criteria written in advance. Running both books in one account surfaced a co-tenancy hazard: a ticker-only position reconciler would have adopted an option written on a swing-book name as a phantom stock long on restart, then stopped it out into a short the account never held; fixed to adopt only stock contracts and fail closed. A third strand turned the same discipline on an outside claim, pre-registering gates, splits and 68 parameter variants in an append-only ledger *before* any backtest ran, then evaluating 27 years of daily data on a sealed out-of-sample window with causal execution and realistic costs: *none* of the 68 survived, the source's headline strategy replicating at 20% CAGR and 1.36 Sharpe against a claimed 34% and 1.66; the negative result is published in full with a recommendation to allocate nothing | trading.vineai.tech, published results

- **Damodaran Bot & Buffybot:** paired LLM-driven equity-valuation agents (Damodaran DCF; Buffett quality: moat gate, owner earnings, margin of safety); an open-weight GLM-5.2 tier routed through my AI Router gateway writes the qualitative read as structured JSON and deterministic Python engines return intrinsic value + BUY/HOLD/SELL over a FastAPI/SQLite pipeline | buffy.vineai.tech, dbot.vineai.tech
- **Podbay** (open source, Go): runtime contract layer for Podman stacks: declarative service contracts with preflight validation, deterministic deploys with receipts, and drift detection | github.com/1eve1Up/podbay
- **Cross-sectional deep-learning ranker** (graduate deep-learning coursework; research only, closed with a negative result written up in full): a pooled GRU and Transformer encoder that ranks all 882 names at once off the same market-data spine as the trading system above, over a 2,936-session panel of 2.03M samples, ten causal rank-normalized features on a 60-session window, training against a sector-demeaned forward return with a within-day rank-IC surrogate loss under purged annual walk-forward and three seeds per fold, on a local RTX 5060. The pass rule was pre-registered before any model ran: beat both a ridge and a Bollinger- z benchmark on information coefficient and hold a positive hedged Sharpe in at least four of six test years. Four rounds failed it and the track is now closed. Per-name encoders went 1 of 6 and 2 of 6; set attention over same-day sector peers came closest at 3 of 6 and 5 of 6, missing 2023 by an IC of 0.0016 with both t -statistics under 1, while an all-name control overfit outright (fold train IC 0.13 against validation -0.005). Re-running the best variant on 10- and 20-session labels, also pre-registered, closed horizon as a lever and showed the post-2022 weakness to be a sign inversion rather than decay. A fourth round retrained from 2019 alone, testing whether a regime-matched window was the missing piece: 2 of 6 and 3 of 6. It reproduced the 2021 fold off 233 sessions (IC 0.023 against 0.024, ridge 0.004 on the same window), so that year holds a real non-linear signal, yet still inverted in 2025 (IC -0.019) having never seen 2015–2018. Capacity, peer structure, horizon and training window are each eliminated in turn, and no validation rule can select against a flip that always postdates its own validation year.
- **Self-hosted platform operations:** the products above run on my own hardware, which makes backup and recovery my problem too. A nightly harness dumps every containerized database logically (14 Postgres and TimescaleDB clusters plus MySQL and SQLite, the last through the sqlite3 backup API rather than a file copy, since copying a live database file is how backups quietly stop working), mirrors every project tree to a JBOD array that keeps changed and deleted files in a history tree, and syncs the result to offsite object storage: 384 GB local plus cloud, seven weeks continuous, 14 of 14 dumps succeeding nightly, with a freshness watchdog that pages when a run goes missing rather than only when one errors. It replaced per-app cron jobs that had been failing silently. On top of it, a half-hourly job replicates 22 applications and 17 databases over a private mesh to a second physical machine, refusing to ship a dump that is stale or truncated and serialized under a lock so runs cannot overlap, 58 of 58 clean since it turned on. I am deliberately not calling that a warm standby: the bytes are verifiably on the far side and the restore procedure is written, but no restore has been rehearsed and the second machine runs none of the mirrored services, so recovery time is recorded as unknown rather than estimated
- **Reach:** multi-tenant auto-apply job-search SaaS that runs the whole hunt: scraped feeds and 580+ hosted ATS boards across 34 sources fill one $\sim 228\text{K}$ -posting live inventory; per user it scores every posting, writes a grounded resume and cover letter per role via my self-hosted gateway, applies over email or ATS forms, and builds interview-prep primers. Applying spans a headless Playwright filler, a resident CDP-driven Chrome, and a cross-browser extension covering eleven ATS families, CAPTCHAs human-only. A 30-day per-host audit (on identical forms, headless sent 4–18% of attempts against the resident browser's 87–95%) moved applying onto a resident queue, opt-in unattended mode capped at 30 a day. The scorer is my own model: a 14B student that replaced the 120B teacher at no marginal cost, keeping 18% of downvoted postings on a frozen 365-pair human-voted gate against that teacher's 55%, AUC 0.649 from 0.564. A deterministic requirements gate, grounded in verbatim quotes, runs shadow-only over 5,498 decisions against a pre-registered 10% bar. The ATS form mapper is now my own too: a Qwen3-8B student distilled from 5,342 Opus-labeled forms and served W4A16 on a local card. Its per-row agreement with Opus (0.913) beat the hosted model it replaced (0.857), now its failover. An Opus post-mortem then judged all 151 of my own rejections against exactly what each employer saw, exposing three production defects in tailoring and screening answers, each fixed that day with a deterministic guard. An Opus career advisor turns each active user's funnel, outcomes and interviews into three app-linked recommendations every few days, and a WCAG 2.1 AA pass (opt-in high contrast, jsx-a11y lint clean) shipped with disability-inclusive filters and Schedule A sourcing. Independent scrape, scoring and apply loops ended a starvation mode and put scoring twelve-wide, 689 to 1,710 an hour. Stripe billing with a zero-match refund; free resume checker; 31 countries; 14,000+ programmatic pages | reach.jobs
- **Reach growth agent** (internal, self-hosted social listening and engagement): a self-hosted replacement for hosted social-

listening tools, built for my own products with Reach first. Saved-search listeners pull candidate posts from Threads (Graph API, with a Playwright browser fallback), X and Instagram (browser-session search and replies) and TikTok (search from the web app's own JSON feeds), with LinkedIn (replies only as the company Page) and Facebook Page (Graph API) adapters built, an LLM tier classifies each for fit and intent, and a tiered media step skips posts that depend on unseen content and describes images with a vision model when the text scores high. Replies are drafted in the product's voice through my self-hosted LiteLLM gateway, with escalation-tier fallback, and land in a human review queue, or auto-post once an integration is cleared for it; guardrails enforce quiet hours, daily caps, jitter, per-user cooldowns, mandatory affiliation disclosure, idempotent posting and a sanitizer that truncates only at real sentence ends and never inside a URL. Replied threads are tracked for follow-ups, and tagged short links feed a conversion readback from Reach. FastAPI/Jinja portal behind a password gate, Postgres 16 and a worker loop under Docker Compose, pytest and Prometheus metrics; its first 30 reviewed replies posted within two days of the first commit, and 131 replies have posted across Threads, X and Instagram since.

- **The Ghosting Index** (monthly public data report, first edition October 2026): a recurring report on what happens to job applications after they are sent, built from Reach's anonymized outcomes, where each job seeker's dedicated inbound mailbox shows what employers actually send back. The launch edition followed 3,263 applications from 57 people, each given at least 30 days: 79.9% never drew a human response (30.7% got only an automated receipt), 17.7% were rejected and 1.6% reached an interview, broken out by applicant tracking system, seniority, workplace type, occupation and send month, with median and 75th-percentile days to a rejection. Each application counts once at the furthest outcome it reached, and mail the classifier could not place is re-read with stricter rules before it counts as a human reply. Privacy is structural: only aggregates leave the module, a bucket is shown only with 30 applications from 5 people and no single account above half of it, and each edition is an immutable snapshot the worker publishes monthly, with a CSV export, Dataset JSON-LD and an archive | reach.jobs/reports/ghosting-index
- **Off The List**: multi-tenant privacy opt-out SaaS (DeleteMe/Optery class), live at offthelist.org. It finds a subscriber's personal information across ~1,830 data-broker and people-search sites, gets it removed, and keeps it down. A worker sweeps monitored brokers (site-scoped meta-search plus direct record probes) and files each hit as an evidenced exposure with a match-confidence score (an optional open-weight LLM judges borderline matches; a lexical fallback keeps it working with no gateway), then per broker either emails a deterministic CCPA/CPRA deletion notice to the broker's privacy contact or drives a guided opt-out; a verification loop re-checks after each broker's processing window, re-files if still listed, and auto-reopens the case on reappearance. Confirmation-email brokers are closed out automatically by a managed opt-out inbox: unique per-request `optout+` aliases, inbound mail routed through Cloudflare Email Routing to a webhook, and auto-clicked confirmation links. FastAPI/PostgreSQL/Redis asyncio-worker backend, React/Vite/Tailwind front end, Clerk JIT multi-tenancy, and a Cloudflare tunnel; live CCPA delivery via Resend with append-only email and removal audit logs | offthelist.org
- **CouncilRadar**: multi-tenant civic-intelligence SaaS, live at councilradar.org. It monitors municipal council agendas and warns small businesses before the vote, not after it. Scrapers cover seven agenda platforms (Legistar, PrimeGov, CivicClerk, Granicus, CivicPlus, IQM2, eScribe) through their public JSON, OData, RSS and HTML surfaces with no headless browser and no scraping of authenticated pages, with failures isolated per municipality so one broken city never stalls the sweep; platforms that publish no structured agenda items get them derived from the compiled agenda document by a fast open-weight LLM, guarded by a sentinel row against refetch loops. Every item is classified by a fast tier against a ten-industry small-business taxonomy with an impact grade, and medium-or-higher hits fire an instant email while the free tier receives a weekly digest; each alert is claimed through a unique (subscriber, item) key *before* the send, so an alert can be lost to a crash but never duplicated, a deliberate trade because duplicate and false alarms churn subscribers faster than a rare miss. First production sweep: 57 active municipalities across 25 states, 1,786 meetings and 8,999 agenda items classified with zero scrape failures. FastAPI/PostgreSQL/Redis async worker loops with Postgres advisory-lock leader election and per-loop heartbeats, React/Vite/Tailwind front end, Clerk JIT multi-tenancy, Stripe lookup-key billing across four tiers, all model calls routed through my self-hosted LLM gateway, and a programmatic-SEO surface of per-city and per-city-per-industry ordinance-tracker pages with a dynamic sitemap, IndexNow syndication and a thin-content noindex guard that flips pages to indexed only once the classifier surfaces real matter, extended with a weekly "STR Ordinance Watch" roundup (permanent per-week edition URLs server-rendered from that week's classified high- and medium-impact short-term-rental items grouped by city, empty editions rendering noindex) as a durable link target for community distribution, plus an admin console for plan overrides and per-user activity | councilradar.org
- **Family Compass**: shared household lists and a family calendar, plus per-child device modes (school, free, bedtime) on one weekly RRULE schedule. Kotlin/Spring Boot on Postgres with a Next.js 15 parent portal behind a Cloudflare tunnel. The enforcement half assumes an unreliable network and an adversarial child. Signed ECDSA P-256 policy is version-monotonic against replay; past expiry the agent keeps enforcing the last cached copy and reports itself stale rather than opening up, and an untrusted clock can only tighten the mode. On Windows, scoped to the child's own SID, a process watcher and Image File Execution Options redirects stop named launchers, bedtime re-locks the session, and a loopback proxy enforces the school allow-list and forces SafeSearch upstream without intercepting TLS, matching apps on path and publisher, not

exe name. Six content categories (games, video, social, AI, shopping, adult) are set Allow or Block per child and per mode over a 338-host catalog extensible per family; category blocks survive free time and mirror into Chrome, Edge and Firefox enterprise URL blocklists. Screen time is metered: weekday and weekend game-time budgets with an optional pooled weekend, parent bonus grants, and a five-minute warning before any tightening. One C# policy core drives the Windows and macOS agents and an independent Kotlin one covers Android, all three answering to a frozen vector file replayed by each language's tests. Heartbeats let the portal flag an agent installed but not enforcing: two Windows PCs enforce live on a 30-second signed-policy poll and have self-updated unattended from 1.0.8 to 1.0.23, while the one enrolled Mac has been silent since 25 August, leaving the macOS port unproven. On Android, the phone app (1.0.4, installed and reporting) now carries every parental control the portal has, from per-child limits and the category grid to seven-day screen time and the site-request inbox | compass.family

- **Offword** (Android, live at offword.app): “record now, read later” conversation and voice-note transcription that never leaves the phone. Kotlin/Jetpack Compose with Hilt, Room and DataStore; a foreground recording service routes the phone mic or a Bluetooth headset into on-device sherpa-onnx speech recognition, with model management, sessions, search and an in-app update check, and no cloud transcription at any tier. The 0.5 early beta adds on-device punctuation and casing, speaker labels, TextRank summaries, and an NVIDIA Parakeet batch model beside a GigaSpeech streaming one; consented beta uploads feed a Go ingest service and a whisper large-v3 reference run on my local GPU, which cut word error rate from 45.1% to 22.9% streaming and 14.1% batch. Models download as resumable files from a Hugging Face and self-hosted mirror. Releases 0.6 and 0.7 add the Offword wearable: Seeded XIAO nRF52840 Sense boards I flashed with Omi devkit firmware stream 16 kHz Opus over Bluetooth LE to a Kotlin GATT client (Concentus decode, reconnect with packet-loss concealment, battery and RSSI readout), with zero lost packets across the first 49 minutes of device recordings; an always-on mode files each conversation as its own session using silero VAD, and speaker turns now come from post-hoc pyannote segmentation plus CAM++ diarization. The marketing site deploys as Cloudflare Worker static assets, while APKs ship from a separate nginx plus cloudflared-tunnel stack at dl.offword.app because Worker assets cap at 25 MiB | offword.app
- **VineAI Search**: self-hosted, privacy-respecting meta-search over 70+ engines with custom BM25 + domain-trust + MMR ranking and streaming AI answers from a self-hosted 120B-parameter vLLM model; now the shared retrieval backend for eight of my own products, a role that exposed three failures worth more than the feature work. The gateway had been returning nothing while every container stayed healthy: 97.9% of upstream fetches came back empty, because an engine listed without an explicit disable flag silently inherits the upstream default, so the engines that ship disabled never ran while every enabled one was hard-blocked from this host (access denied, rate limits, CAPTCHAs) after burning its full timeout; and since empty payloads are never cached, the cache sat at 198 entries against a 2,000 cap and every query needed a fresh upstream slot against a 10/minute budget, saturating the queue until searches timed out. Probing every engine individually and inverting the pool (enabling only those verified answering, more than one so a single CAPTCHA cannot repeat the outage) took results per query from 0 to 29–88, empty fetches to 0%, and repeat queries from a 3.1-second miss to a 0.5-millisecond cache hit. Multi-tenant fairness had never worked either: consumers were bucketed by client IP, but every internal caller arrives through the proxy's single bridge address, so all eight products shared one FIFO queue and interactive searches waited behind whole scraper runs (one system's ticker sweeps are 60% of traffic); each consumer now sends an application header and gets its own round-robin bucket. Load shedding was mistranslated too: the gateway sheds with 429 and Retry-After, but the API collapsed every non-2xx into a 502, so backpressure reached callers as “backend unavailable” and three in a row opened a consumer's circuit breaker for fifteen minutes; 429 now passes through intact and retryable, verified under forced backpressure: 20 of 25 concurrent queries shed with Retry-After where all 20 had previously been 502s | search.vineai.tech
- **Automated video pipeline** (on my own GPUs; twenty-six episodes are now public on YouTube): a 12-stage orchestrator (research, fact-check, script, lint, narrate, visualize, score, assemble, QA, upload, publish, promote) that takes a topic to a rendered long-form video. Postgres holds channels, topics, episodes and every per-stage run; a leader lease over a session advisory lock lets one process own the board, with episodes as bounded-concurrency async tasks. LLM stages call my own router (Claude Opus 5 to write, Haiku 4.5 for cheap passes) over my self-hosted search gateway, and a research gate refuses to advance a topic whose sourcing misses a quoted-fact, source-count and distinctive-term bar. A GPU worker service exposes local open-weight models as adapters: Chatterbox voice cloning for narration, faster-whisper large-v3 for word timings, Z-Image stills and Wan 2.2 image-to-video hero clips through ComfyUI, ACE-Step for the score, with Remotion for charts and ffmpeg for assembly; those adapters have served 4,282 real jobs on one box. Publishing is a real OAuth and resumable-upload path with an encrypted token store and a quota ledger: uploads land private with a publishAt and a poller records the flip. Twenty-six episodes have run the whole pipeline through to public videos, each flipped public by YouTube on its schedule between 09-08 and 09-17, and sixteen more are uploaded and scheduled from 09-18 to 10-09, out of 160 attempted, 163 rendered cuts and 281 GB on disk; the nightly YouTube Analytics pull covers nine of them. The wall is account-side and moved when the account did: with verification a per-channel flag, phone-verifying three of four channels lifted the 15-minute ceiling on them, cutting ceiling-held episodes from 90 to 26 (all on the fourth); 92 of the 118 deferred now wait only on the daily insert quota, against 48 uploads that landed. The promote stage drafts Shorts cuts and a community post rather than posting them.

- **AI Content Detector:** forensic tool estimating the probability that text, documents, images, or video are AI-generated. It fuses independent signals (token-level perplexity/rank via vLLM `prompt_logprobs`, cross-model entropy ratio, an LLM judge, stylometry, C2PA provenance, image/frequency forensics) in logit space, runs entirely on self-hosted open-weight models (no third-party APIs), and logs ground-truth feedback for recalibration | detector.vineai.tech
- **Portfolio Optimizer:** 8 optimization models (Mean-Variance, CAPM, Black-Litterman, Risk Parity, HRP, Monte Carlo) with GPU-accelerated covariance estimation and S&P 500 backtesting | portfolio.vineai.tech
- **AI Chatbot Platform & Study Buddy:** multitenant chatbot hosting with RAG knowledge bases, SSE streaming, an embeddable widget, and Clerk authentication; bots also answer on five messaging channels (Telegram, WhatsApp, Messenger, Instagram, Slack) through one shared RAG reply pipeline with signature-verified webhooks and platform-native reply formatting (Slack mrkdwn, WhatsApp, Telegram HTML, plain text), Stripe subscription billing (hosted Checkout and customer portal with signature-verified webhooks as the sole source of truth for plan state), metering bot count, monthly messages, messaging channels and access to the 120B model across a free and a \$15/mo Pro tier, enforced at bot creation, web chat and channel reply, with every pre-launch account grandfathered; and cross-channel “delete my data” request detection that acknowledges and flags for operator alerts; the platform also serves an external client’s public-site assistant as a drop-in widget; plus a Study Buddy AI tutor with Socratic questioning | chatbots.vineai.tech
- **Sports Analytics & Prediction Markets:** AI predictions across 6 sports with multi-sportsbook odds; aggregator over Kalshi/Polymarket/Manifold/PredictIt with arbitrage detection | sports.vineai.tech
- **MatchiMatchi** (cross-brand cosmetics shade matcher, live on Cloudflare Workers): a zero-dependency single Worker that turns public open datasets into a shade-matching PWA. Skin tone is sampled in the browser from a camera frame, an elliptical cheek patch with the darkest and brightest quartiles trimmed before averaging, then converted sRGB to CIELAB and ranked by CIEDE2000 against 6,792 foundation shades across 107 brands; a second index covers 5,275 colors over 685 products for cross-category dupes. The camera path makes exactly one network request, for the static shade file: no frame or derived measurement leaves the device, a deliberate biometric-privacy posture rather than a feature note. One color library is the single source of truth for both browser and Worker, copied at build rather than reimplemented. The SEO surface is programmatic: 6,899 generated pages, one per shade with its cross-brand twins, undertone and depth reads, plus brand indexes, served from a JSON bundle inside the Worker behind a 6,903-URL sitemap. Installable and offline-capable via a service worker caching shell and data while excluding API and shade routes. Built partly through an experimental development loop: an on-site feedback widget writes to Workers KV behind a honeypot and per-IP and global rate limits, a cron drain claims each item under a bearer secret and starts one headless coding-agent run against a prompt that treats the feedback as untrusted input and refuses anything outside the repo; one such run shipped a live UI change end to end. Deliberately unclaimed: it sits on a `workers.dev` URL with no custom domain, no analytics, and no monetization, and swatch hexes are brand marketing values rather than colorimeter readings, which the site itself discloses | matchimatchi.bryan-0fa.workers.dev
- **smog** (launcher and signed catalog for AI-made game mashups and decomp/recomp native ports, live at smog.games): a Linux-first desktop launcher and catalog built around one premise, that a mod which injects code into a game you paid for behaves exactly like malware, so nothing is listed on trust. Discovery adapters (GitHub search and releases, curated decomp indexes, a mashup index, public submissions) feed a normalize, dedupe and LLM-triage stage; every version then runs a static-only screening pipeline that never executes an artifact (AppImages are unpacked with `unsquashfs`): provenance checks on repository age, star velocity and CI attestations, ClamAV plus YARA rules for token stealers, miners and droppers, Mandiant capa capability and packer/entropy checks, VirusTotal hash lookup with no automatic upload, and Simgrep plus LLM review of the source or the diff since the last approved version. The pipeline may auto-reject but can never approve; a human sign-off in a password-protected admin console is required for every entry and every version. The publisher signs the catalog with minisign (Ed25519) and renders the static site in one atomic step; the Tauri 2 launcher pins the public key, refuses unsigned or rolled-back catalogs, honors a revocation kill switch on each six-hour refresh, verifies user-supplied game files by hash locally, and self-updates only from signed releases; no game assets are ever hosted. FastAPI public API and HTMX admin, arq/Redis worker, PostgreSQL, nginx behind a Cloudflare tunnel, all model calls through my self-hosted LLM gateway. Launcher 0.1.0 shipped as an AppImage on 16 October 2026; the first discovery runs surfaced 120 candidates and 27 scanned versions now await human review, so the public catalog is still empty | smog.games
- **Video Compressor:** target-size video compression with two-pass H.264/H.265 encoding, chunked uploads, and self-cleaning storage | converter.vineai.tech

EDUCATION

Georgia Institute of Technology – MS Computer Science, Machine Learning Specialization | Expected May 2027

California State Polytechnic University, Pomona – BS Computer Science, Java-centered curriculum (Minors: Mathematics, Business)

CERTIFICATIONS

Certified Information Systems Security Professional (CISSP) • Red Hat Certified Systems Administrator • AWS Cloud Practitioner • Azure Fundamentals • Google Cloud Foundational • Cisco Certified Network Associate • Python Certified Entry-Level Programmer •

