

Travis Carter

Staff Software Engineer / Architect

Las Vegas, NV · Remote · phatjam98@gmail.com · 702.572.7734 · phatjam98.com · github.com/phatjam98 · linkedin.com/in/travis-carter

SUMMARY

Platform and distributed-systems architect. I change production systems that cannot stop: a 34-module Java monolith through five simultaneous major-version upgrades, a strangler decomposition behind a gRPC coexistence layer, a production AWS estate migrated without losing the data plane. A decade of civil engineering design before software taught me to size the structure first. AI pairing is deliberate practice, not a shortcut: I built the tooling my team ships through, and the trailers are in the history.

EXPERIENCE

Sr. Software Engineer, ECFX Oct 2023 – Present · Remote

- Designed and built the four-service, config-driven DAG platform that turns onboarding a new court from an engineering project into a configuration change, collapsing a per-court processor hierarchy of several hundred Java classes into seven patterns over one pipeline and a config document. 430 commits across four repositories, all mine, 718 files, April to September 2026. The first court pattern runs end to end on dev; staging and production stay frozen by direction until every in-scope court proves out.
- Drove five simultaneous major-version upgrades across the 34-module production monolith with no big-bang cutover: Java 11 to 25, Micronaut 1 to 5, Hibernate 7, Gradle 9.5, Groovy 5, sequenced as a ticket chain that walked modules to green one at a time. Root-caused the regressions myself, including a Jackson double-decode that silently broke SAML login. Sole driver of the modernization program, and it all ships on master with no Micronaut 1 pins left.
- Sequenced the monolith's strangler decomposition into independently shippable steps: an 85-class JPA entity library extracted from it (130 of 137 non-merge commits), the seven-service gRPC bridge that lets Micronaut 1 and Micronaut 5 read one PostgreSQL source of truth (111 of 112), and Micronaut 5 proved on the extracted receipt-ingest service alongside another engineer (80 of 118).
- Led the production migration off self-managed EKS and RDS onto DuploCloud as the only engineer on it: wrote the cutover runbook, took AWS DMS as far as three blocking defects on the roughly 5 TB primary database before standardizing on native PostgreSQL logical replication, ran the per-firm KMS re-encryption as a gated step of the rollover, and executed the teardown as one commit removing 15,412 lines across 74 files while the live data plane kept serving. Post-run verification found all 207 firm keys on the new CMK, none on the legacy alias.
- Owned the platform under all of it: Kubernetes manifests for every service (112 of 680 non-merge commits), a resource-optimization sprint standardizing JVM tuning and health probes across 20+ services, and a Karpenter major-version upgrade (v0.37 to 1.0.2, CRD v1beta1 to v1) plus an EC2 instance-generation modernization across three EKS clusters.
- Cut median master CI time about 65%, from 38 minutes (September 2024) to 11 minutes (July 2025), across two changes: moving the pipeline off GitLab shared runners onto self-hosted runners, then replacing a serial Docker build loop over 17 services with parallel per-app Gradle Jib builds and path-based change detection with shared-library dependency fan-out. Full rebuilds of all 17 services, the same unit of work either way, went from 27 minutes to about 12.
- Built the OAuth 2.1 authorization server and a fail-closed SAML replay defense, rejecting Redis for the replay store because the incumbent Redis auth path fails open and using an atomic Postgres insert-on-conflict instead. Put the firm-scoped MCP server into production behind it: 13 analytics tools, RS256/JWKS with RFC 8707 audience binding and a host allow-list, two replicas on the production tenant, a first live validation of roughly 697 tool calls (73 of 74 non-merge commits mine). Reopened it in September for third-party LLM clients: dynamic client registration under a deny-list policy, RFC 8252 loopback redirects, RFC 9207 issuer, automatic key rotation.
- Shipped a case-requeue subsystem for the document-management integration across four repositories in three weeks: a dispatch lane with per-intent attempt budgets, a queue-age gauge and staleness alert, provider-exception classification, and operator surfaces in the customer dashboard and the internal admin app.

- Built the two platforms the team now ships through. Observability as code: two live Grafana environments off click-ops and into Terraform in 24 days, 21 dashboards adopted with zero imports, four other engineers now shipping dashboards and alerts through it (69 of 86 mine). AI engineering: a Claude Code component library of 65 skills and 33 agents with a closed-loop TDD memory system, now seven authors including four other engineers (33 of 79 mine).

Director of Platform Engineering, The Public Health Company Jan 2023 – Mar 2024 · Remote

- Led platform engineering at a Series A health-tech company, owning architecture, system design and the development roadmap for a global biosecurity platform.
- Built the real-time risk anomaly-detection and notification platform from scratch on Flink, Elasticsearch, Protocol Buffers and Micronaut, with a gRPC API layer over Elasticsearch behind the product's queries.
- Designed a mixed-language Flink StateFun framework so detection processors could be written in Java or Python, letting the research team work in the language it already used.

Sr. Software Engineer, The Public Health Company Oct 2021 – Dec 2022 · Remote

- Built the Flink ingestion, anomaly-detection and notification-generation pipelines at the core of the platform, with Micronaut gRPC services in front of them.
- Owned data-schema management and wrote the shared Java libraries the services built on.

Sr. Software Engineer, Globality Oct 2020 – Oct 2021 · Remote

- Architected the Kubernetes cluster layout and deployed vCluster to give a 20-person engineering org sandboxed per-developer environments instead of contention over shared staging.
- Established the company's CI/CD process and Kubernetes practice, and built a new authentication service.

Sr. Software Engineer, Banjo Aug 2014 – Aug 2020 · Remote and Hybrid

- Built real-time event processing and anomaly detection over live public-signal data, including the automation path that took machine-learning detections straight to breaking-news notification, cutting discovery to notification from seven minutes to under thirty seconds and removing the manual review step.
- Created the inter-agency event-code coordination framework that let law enforcement, fire and public-safety agencies match incidents across jurisdictions. The patent below covers it.
- Led the core systems team through architecture and the conversion of the Ruby platform to Java across multiple verticals.

Earlier, 1998 – 2014. Frontend Developer, 2011 – 2014: JavaScript single-page applications and UX work. Civil engineering draftsman and project designer, 1998 – 2009: plan creation, design and engineering approvals on golf course, commercial and international land-development projects.

OPEN SOURCE & PROJECTS

- **proto-faker.** Java library generating field-name-aware fake data for arbitrary Protocol Buffer messages. Sole-authored, published to Maven Central (0.1.0, 0.2.0) via a GPG-signed JReleaser pipeline, Java 11/17/21 CI matrix.
- **EventHorizon.** Real-time anomaly-detection platform: five chained Flink jobs over Kafka fusing statistical and ML scores into one ranked result per event, with Elasticsearch and pgvector behind a GraphQL/SSE gateway. 13 modules, six ADRs, Apache 2.0, sole-authored across all 70 commits on main. Runs locally, not scale-tested.
- **phatjam98.com.** Astro portfolio: architecture case studies and essays on context engineering and agent-driven development.

SKILLS

Languages. Java, TypeScript, Python, Ruby, Groovy, SQL

Streaming & Data. Flink, Kafka, RabbitMQ, gRPC, Protocol Buffers, Elasticsearch, PostgreSQL

Frameworks & Cloud. Micronaut, Hibernate, Spock, Vue, AWS, EKS, Kubernetes, Terraform, Docker, GitLab CI

Observability & AI. Grafana, Prometheus, Loki, observability as code, Claude Code, Model Context Protocol, context engineering, TDD, event-driven architecture

EDUCATION & PATENT

Kaplan University · AS, Computer Science · 2010 – 2011. Patented a method for matching event codes between various agencies, enhancing situational awareness.