Mark Davis

Backend and platform engineer. Six years in logistics and payments. Go, Postgres, AWS. Austin, or remote in US timezones.

mark@markdavis.dev markdavis.dev Austin, Texas

SHIPPED

2025

Dispatch matching rewrite, Freightline

p99 latency 1.4s to 180ms, database spend halved.

- Matching ran as a single Postgres query with eleven joins and had grown a 1.4 second p99 at morning peak; staff refreshing to force retries doubled the load that was already the problem.
- Split the read path out, precomputed driver eligibility on a five second cycle, and moved final ranking into the application so it could short-circuit. Correctness kept: ineligibility is filtered at assignment, not at ranking.
- Result: p99 180ms, timeouts down 94 percent, database spend from $6,200 to $2,800 a month, rolled out behind a flag over three weeks.

2024

Idempotent payouts pipeline, Freightline

Zero duplicate payments in 18 months across $41M a year.

- Carrier payouts were triggered by a webhook with no idempotency and a retry policy that assumed success. Two redeliveries in one quarter paid carriers twice.
- Every payout now carries a deterministic key derived from the load and settlement period, written in the same transaction as the ledger entry, so a replay is a lookup rather than a payment.
- Result: no duplicate payouts since, and nightly reconciliation down from four hours to six minutes.

2023

On-call rehabilitation, Freightline

Paging volume from eleven a week to two, with no missed incidents.

- Rewrote alerts around symptoms customers notice, attached runbook links to the page body, and deleted alerts that had never corresponded to an outage.
- Result: eleven pages a week down to two, full runbook coverage on everything that still pages, and no incident missed since the change.

EXPERIENCE AND EDUCATION

2022 to now

Senior Backend Engineer, Freightline

Logistics SaaS. Matching, payouts, platform reliability.

2020 to 2022

Backend Engineer, Corsair Payments

Ledger services and processor integrations.

2020

BS Computer Science, Texas A&M University

STACK

Daily	Go, Postgres including query plans and partitioning, AWS with Terraform.
Comfortable	Python for data and tooling, SQS and Lambda, Prometheus and Grafana.
Can operate	Kubernetes. I would not claim to run a cluster alone.
Honest limit	Enough React for an internal tool. Frontend is not my strength.
Open source	ratelim, a Go token bucket limiter with a Redis backend, used by around forty projects.