Dave Bradford

Boise, Idaho (MST) · linkedin.com/in/zerobearing · github.com/zerobearing2 · zerobearing.com

Professional summary

AI-native engineer and engineering leader. Give me a concept and I'll ship the product, then build the team that keeps shipping it. I was the founding engineer at Daasity and on Gierd's product team, the first engineer at Odoclub, and one of the first at PEAR Health Labs and RaceHQ. As the Gierd team grew, I moved from writing the code myself to code review, technical direction, and mentoring: I hired engineers and quickly took a front-end developer to a mid-level full-stack Rails engineer. Most of my code now comes through AI agents, inside a process I design: requirements pressure-tested up front, a spec and tickets with numbered acceptance criteria, agents building test-first in isolation, and CI, browser QA, and independent review on every change before a draft PR an engineer approves. I'm looking for a full-time role as a hands-on engineering leader or a senior, staff, or principal engineer, in Rails or product engineering.

Work experience

Gierd, Inc. Boise, ID

Sr. Principal Engineer / VP of Engineering Feb 2024 to Aug 2026

- Founding engineer on the product team. Joined when Gierd ran on a collection of third-party products with no unified customer-facing application, and made nearly all the early technical decisions. Year one was the foundation: framework, platform, and pipelines, doing the lead engineer's job and the VP's at the same time.
- Grew the engineering team, onshore and offshore, hiring full-time: seniors plus intermediate and junior engineers. As the team grew, stepped back from day-to-day Rails code to code review, team direction, and mentoring, and oversaw engineering while major features after the MVP were built from product-team specs. Mentored through pair work and code review, and led from inside the codebase: architecture, production DevOps, coding standards. Mentored a front-end developer into a mid-level full-stack Rails engineer.
- Introduced coding standards and an SDLC in place of ad hoc deploys to cloud functions. Put disaster recovery in place for code and customer data.
- As VP, rejected a feature built on a service-based architecture the codebase had never used. Rewrote a use case in plain Rails and brought pros and cons research. After shipping, the team agreed the simpler version was easier to debug.
- Helped build the initial Rails MVP in a matter of weeks: a multi-tenant SaaS app with embedded Metabase dashboards. It replaced a patchwork of bespoke tools (a Metabase dashboard hand-rebuilt per customer, a separate reconciliation app, email-driven processes) with one login covering dashboards, settings, and notifications, and became the product's foundation.
- Rebuilt the data extraction pipeline system from the ground up and owned its tech selection: the pipeline tools (Kestra, Airbyte, dbt), the process, and the data structure. Standardized data schemas across customer accounts and unified ingestion from about half a dozen major marketplaces and vendor platforms into one structure, so a dashboard or feature built for one customer shipped to all. Cut turnaround on dashboards and ad hoc queries, and cut data-extraction error rates well below the old scripts. Custom scripts handled API extractions, and Rails background jobs coordinated with the UI and triggered events. The hardest and longest part was marketplace extraction: rate limits, data validation, and orchestrating it all.
- Did all the initial transform work while the data analyst team was being hired, built all the orchestration, and migrated the existing workflows off GCP onto the new system. Chose Kestra to orchestrate extraction, transform, and data-movement workflows, replacing GCP cloud functions that had sparse observability. Failures became visible in the Kestra UI with Slack alerts, support could go from error to logs to fix without a GCP dig, workflow changes went through GitHub PRs, and visible failures pushed the team to fix recurring bugs like rate limiting. Self-hosted some of the infrastructure to stay cloud-agnostic, which kept a run-in-customer-infrastructure option open.
- Designed multi-cloud infrastructure (GCP and AWS) spanning internal and customer-facing services, and built data pipeline systems from raw ingestion to the UI layer. Key stakeholder in the data platform alongside the CTO. Kept BigQuery as the warehouse: legacy systems had too many bespoke dependents, and experiments showed a switch would set the company back.
- Replaced manually provisioned, copy-pasted per-customer scripts with an automated data platform that provisions and manages infrastructure when an account is created. Setup went from a full day to 10 to 15 minutes of provisioning, with first data flowing within a couple of hours of account creation, fewer provisioning errors, and more code reuse.

- As Sr. Principal Engineer, simplified how analytics got from the warehouse to the Rails UI: removed as much of the orchestration and as many moving parts as possible, and reduced the latency between data being calculated in the warehouse and showing up in the Rails app.
- Part of Gierd's applied AI team, a dedicated team stood up to deploy agentic workflows, agent skills, verification loops, and automation (Claude Code, Cursor, Kestra). AI agent automations took over manual cross-system work in customer support, finance, and returns. Built some of them, including the chargeback agent: it assembled evidence for marketplace disputes by cross-referencing about half a dozen systems, taking a job from hours to a day or two by hand down to minutes.
- Stood up the AI workloads in the SDLC: code review, diagnosing and repairing CI breakages, and ticket triage. Established AI-augmented development practices and built internal AI skills and automations for teams across the org.

Daasity San Diego, CA

Director of Platform Engineering / Lead Software Engineer May 2017 to Feb 2024

- Founding engineer. Joined before the product existed, when the founders ran the work by hand as manual processes for a couple of customers. Grew a couple of bespoke scripts into the full platform and was there for the whole build. Replaced hours of manual per-customer data compilation and weekly founder-led review meetings with a self-serve Rails platform running nightly extraction into a warehouse behind a unified data schema across marketplaces. Chose Rails, since the CTO wasn't a Rails specialist, and was the platform's main architect until I hired Rails coworkers. Early on, built every piece of the product.
- As Director of Platform Engineering, stayed hands-on. Later focused on the Rails side of engineering while another team owned the dashboards and SQL, kept owning infrastructure, production environments, deployments, and day-to-day ops, and still took coding tasks and optimizations. Handled the Director work with the CTO and worked with the C-suite to align engineering output with product strategy.
- Helped build and train the engineering team as it grew to 25+ across product and engineering, hiring through an agency with short trial periods. Mentored and did code reviews. Established coding standards, development workflows, and infrastructure foundations the team scaled on.
- Designed and built the extraction pipelines in Ruby on Sidekiq Enterprise, feeding Snowflake and BigQuery and processing billions of rows per day.
- Architected multi-warehouse support (Snowflake, BigQuery, Redshift), including SQL-dialect differences across them, and built it with the team while staying hands-on. Designed with the CTO a model where customers could own their warehouse or have the platform host and manage it, sharded per customer with no cross-tenant mixing, fed by custom pipelines that cleaned and normalized incoming data.
- Architected a workflow engine that triggered reports and notifications when data became available, which enabled intraday sales reporting. Customers previously had only nightly updates. Built it with the team.
- Owned the stack end to end from the initial Heroku MVP through the migration to an EKS cluster on AWS. Made the case to the CTO for the spend, found an AWS partner for the cutover, then took ownership and tuned the environment so the team kept a Heroku-like workflow. Built custom tooling to replace the Heroku conveniences the team relied on, so the migration stayed out of the way of feature development. The company grew from 1 to 20+ brands over that span.

TaylorMade Golf Company Carlsbad, CA

Senior Software Engineer Oct 2015 to Mar 2017

- Wrote internal mapping software (Rails with a React UX on PostgreSQL) for MyRoundPro, TaylorMade's golf GPS and shot-tracking product. MyRoundPro was built on Microsoft Band as part of an exclusive TaylorMade and Microsoft partnership. The tool loaded satellite imagery so people could draw around a course's major features and outer bounds, calculated geocoordinates for every feature, and stored them in a database.
- Worked alongside the lead engineer on the app side while Microsoft built the Band hardware. Shots uploaded from the Band to the backend, and the app combined the device coordinates with the mapped course data to place each ball on the course. Golfers viewed their rounds in the web app our team built, with shots plotted on satellite course imagery. It shipped to real customers, and another team later ported it to iOS and Android.

PEAR Health Labs Solana Beach, CA

Senior Software Engineer Jun 2014 to Oct 2015

- Rejoined after PEAR moved to hardware-agnostic mobile software. Worked with the CTO to port the Sinatra backend I'd originally written onto Rails, and built whatever the new platform needed.

Lead Ruby Developer Dec 2010 to Jan 2014

- Founding engineer on the backend. Nothing existed when I started: the firmware engineer was just starting the hardware for Square One, a heart rate monitor and foot pod that gave real-time audio coaching. I chose the stack and built the server backend and API in Ruby on Sinatra. The product was on sale by April 2012, with a web portal for uploading workout data and creating plans.
- Built the API bridge between device firmware and the web app: biometric workout data up for dashboards and history, coaching plans down to the device. I was the first engineer on it and worked through launch.
- When PEAR moved to an iPhone training app in Oct 2012, kept the API and server backend running so coaching plans still loaded from PEAR's servers.

Odoclub, Inc. Carlsbad, CA

Lead Rails Developer Sep 2007 to Jul 2014

- Original and only engineer on Odoclub, a white-labeled B2B SaaS loyalty platform for auto dealerships built on the frequent-flyer model: members earned reward miles by driving and redeemed them at their dealership, where an authorized person checked the odometer to validate starting miles. Each dealership got its own branding, and the goal was customer lifetime value for the dealership.
- Built all of it in Rails and MySQL with one designer, from the client's idea through launch, and made every technical call. The client owned the business concept and I owned the architecture and application development. Held the build to a pixel-perfect design bar through a lot of iteration ahead of a soft launch in dealerships and a slow rollout.
- Shipped it through go-to-market. After launch it mostly ran on its own while I handled occasional features, improvements, and fixes through 2014.

TACTIC Race Solutions, LLC (RaceHQ) Solana Beach, CA

Lead Rails Developer Oct 2011 to Jul 2014

- One of the founding engineers taking RaceHQ from concept to a shipped white-label SaaS for managing race events end to end: vendor relations, registration, participants, event-planning templates, calendar sync, team permissions, and purchasing.
- Worked directly with the company's co-founder and president, who brought the requirements and reviewed and tested the product to make sure we built what he envisioned, and cross-functionally with a designer and the head of engineering.
- A lead on the small team that built all of RaceHQ, on Rails and MongoDB, and made the stack calls. Helped with the initial launch and rollout to a couple of customers. Customers using it before I left included Ruckus Sports and VAVi Sport & Social.

Earlier career

- **mSnap, Inc.** · Sr. Programmer · Solana Beach, CA · Jan 2007 to Nov 2007. First engineer hire. Coded the entire SMS ad platform in Ruby with the VP of Engineering, working with a designer and the founder to spec and build it. Broadcasters and advertisers created and ran SMS ad campaigns on it; paying customers were on it and a large volume of SMS messages ran through the platform.
- **Sharp Healthcare** · Sr. Programmer/Analyst · San Diego, CA · May 2005 to Dec 2006. Used Ruby on Rails and Linux, and did system administration and hardware support.
- **Affinity Development Group** · Sr. Programmer · San Diego, CA · Aug 2002 to May 2005. Active Server Pages and Microsoft SQL Server.
- **Advanced Reservation Systems, Inc** · PHP Programmer · Jan 2002 to Aug 2002. New features, and ported existing functionality to a new platform.
- **BDC Technologies** · Sr. Programmer & Partner · Vista, CA · Feb 1999 to Oct 2002. CGI/Perl on Solaris. Clients included Sony, Universal Music, and Sykes Enterprises.
- **Cyberworks, Inc.** · Sr. Programmer · Vista, CA · Jan 1997 to Jan 1999.
- **PennySaver USA** · Software Developer · Vista, CA · Jan 1996 to Dec 1996. C/C++ pre-press pagination software and ad-update automation. The automation saved 80+ man-hours per week.

Projects

- **The Feedback Agent** · thefeedbackagent.com · Solo. Anonymous feedback: recipients never see the sender, AI rewrites each message for clarity and constructiveness while keeping the sender's voice, and the recipient can reply once (also AI-improved). OTP email verification and rate limiting. Rails 8.1, SQLite, Hotwire, Tailwind + DaisyUI. MVP shipped June 2026; free.

- **Rails-AI** · rails-ai.dev · Solo creator. Open-source (MIT) Claude Code plugin for Rails, built on Superpowers: workflow commands (setup, feature, refactor, debug, review), domain skills (models, controllers, Hotwire, Solid Stack, security, Minitest, performance), and team rules. Early codegen models over-engineered Rails code, so the skills act as guardrails for convention and simplicity.
- **Omatalk** · omatalk.zerobearing.com · Local, on-device text-to-speech plugin for Omarchy Linux: select text, hit a hotkey, and hear it in a neural voice, with no cloud calls or API keys. A background daemon keeps the model warm for instant playback. Submitted to the Omarchy plugin marketplace.

Education

- California State University San Marcos · Computer Science

Certifications

- FAA Private Pilot, Instrument Rating

Skills

- **Languages:** Ruby, JavaScript, PHP, Perl, C/C++, SQL, HTML/CSS
- **Frameworks and libraries:** Ruby on Rails, Hotwire (Turbo, Stimulus), Tailwind CSS, Sidekiq, Solid Stack, Minitest, RSpec
- **Data:** PostgreSQL, MySQL, MongoDB, SQLite, Microsoft SQL Server, BigQuery, Snowflake, Redshift, Kestra, Airbyte, dbt, Metabase
- **Cloud and infrastructure:** AWS, EKS/Kubernetes, GCP, Heroku, Docker, Kamal, CI/CD, Linux, Git, GitHub
- **Architecture:** Multi-tenant SaaS, REST APIs, data pipelines, multi-warehouse analytics
- **AI:** Claude Code, Cursor, OpenAI and Anthropic APIs, LLM integration, AI agents, AI automation, agentic workflows, agent skills, verification loops
- **Leadership and process:** Remote team leadership, technical leadership, engineering team building, hiring, mentoring, code review, coding standards, SDLC design, Agile, Shape Up, TDD, BDD