

Andy Rusterholz

Longmont, CO • andyrusterholz@gmail.com • andy.rusterholz.org • github.com/rusterholz

SUMMARY

Software engineer since 2007, in production Ruby on Rails since 2011, working at the seam between platform and product. One of three primary architects of a workflow orchestration platform that processes 500,000+ events a day for a ~\$900M marketplace, with zero customer-facing incidents in five years, and the author of much of the customer-facing automation built on top of it. Backend specialized (domain modeling, event-driven architecture, API and SDK design, and the instrumentation that keeps it honest), and fluent enough across the stack to build the internal admin UI or write the transactional email template. Architecture at staff scope has never been a reason to stop writing code. Happiest on the problem every other engineer keeps tripping over, or on the tooling that lets a team move fast and sleep well afterward.

SKILLS

Core Expertise: Ruby (2.6–3.3), Rails (6.1–7.2), PostgreSQL, Redis, Sidekiq, Kafka, Zeebe (BPMN), workflow orchestration, event-driven architecture, distributed systems, domain and schema modeling, background processing, retries and idempotency, API design (REST, GraphQL), third-party integrations (REST, SOAP/XML, JSON, FTP/SFTP), SDK and gem development, RSpec and testability-first design, legacy modernization and incremental migration, server-rendered Rails views, ActiveAdmin and custom Arbore components, JavaScript, HTML/CSS

Leadership & Process: technical leadership and architecture ownership, multi-quarter cross-team initiatives, RFC-driven design, cross-functional collaboration with product and design, mentoring and onboarding, technical interviewing, pairing and code review, developer evangelism

Infrastructure & Reliability: AWS, Docker, Kubernetes (EKS), CI/CD (GitHub Actions, Helm, ArgoCD), Datadog (metrics, dashboards, SLOs, monitors), PagerDuty and on-call response, Four Golden Signals instrumentation, incident response, Linux server administration, AI-assisted development (Claude Code as day-to-day driver). Also Elixir, Python, Nomad and Consul, MySQL, and earlier work in PHP and Java EE

OPEN SOURCE

Gems I write and maintain:

busybee (github.com/rusterholz/busybee): Workflow orchestration for Ruby, built on the ISO-standard BPMN format: a worker framework, an idiomatic Zeebe client, a CLI, and RSpec testing integration. Tackles the same class of worker-framework design problems I owned at production scale, for the broader Ruby ecosystem. v0.3 available now; v1.0 targets 2027.

weft (github.com/rusterholz/weft): Component-oriented hypermedia for Ruby, built on Arbore and htmx. Components declare their parameters, derived data, and actions; the framework derives the routing, request handling, and client-side wiring from those declarations. v0.2 available now with a full tutorial and worked-example set; production-ready release targeted for late 2026, v1.0 in 2027.

PROFESSIONAL EXPERIENCE

ezCater: Senior Software Engineer

Boston, MA (Remote) · October 2019 – November 2025

Senior Software Engineer across platform and core services squads over six years, progressing from feature development to architectural leadership while staying hands-on. One of three primary architects of a workflow orchestration platform, and co-owner of order management in the Rails monolith behind the largest online business catering marketplace in the US (~100,000 restaurant partners, ~\$900M in annual bookings, ~2,500 orders/day).

Workflow Orchestration Platform & SDK (2020–2025): One of three primary architects of a horizontally scalable platform built on Zeebe, an open-source engine for the ISO-standard BPMN process format, replacing Kafka choreography that had made distributed business processes hard to track or reason about.

- Designed and built the rules engine service, the bridge between the event-driven world and the orchestration engine: it harvests state-machine and timer events off Kafka and starts or stops workflow instances by configurable rule
- Co-architected the engineer-facing SDK, four gems giving teams different ways to participate: state-machine and timer gems that put events on the message bus, the worker framework gem (near-sole authorship) that defines and runs workflow tasks through a centralized DSL with org-wide cataloging and auditability, and a subscription gem that collapses the single-task case, cutting platform onboarding from one month to one week
- Treated testability as a first-class feature of the SDK: custom RSpec matchers, data factories, and public helpers that let teams unit-test their workflow definitions against a real engine in CI
- Built the engineer-facing ActiveAdmin UIs across the platform's three internal apps with no designer in the loop, extending the stock component library with custom Arbore components and jQuery: collapsible panels, color-coded badges for business event types, and an embedded zoomable bpmn.js viewer rendering live workflow diagrams
- Platform processes ~500,000 events/day, runs ~2,000 workflows, executes ~10,000 tasks daily, horizontally scalable with no identified upper bound; cut feature development from one quarter to four weeks and iteration from two weeks to one-to-two days, with **zero customer-facing incidents** over five years

Product Features & Automation (2020–2023): Built customer-facing features on the platform, owning each from idea through deployment and measurement, alongside product and design.

- Built Direct Connect, a Rails application automating vendor-initiated order changes through a GraphQL interface federated into the main public API graph; it replaced vendor → CS agent → customer phone calls with immediate resolution and cut CS involvement ~20%
- Built Auto-Swap selection logic that reassigned rejected or canceled orders to nearby caterers, rescuing ~30% of eligible orders, and the recommendation engine ranking alternates by distance, menu compatibility, and availability, delivered end-to-end down to the customer-facing email template and its tracking links
- Built a risk-scored order outreach system end-to-end, scoring each order for customer-dissatisfaction risk and routing flagged orders to CS agents for proactive outreach, which lifted reliability on those orders ~35%; later helped make the call to decommission it when measurement showed CS labor costs outweighed the benefit

Legacy Modernization & Domain Modeling (2020–2025): Co-owned order management in the Rails monolith: data modeling, performance, incident response, and the migration path to the new platform, without big-bang rewrites.

- Co-designed a "manifest" storage model (with another Senior SE and a Staff SE) to replace a 100+ column Order god-object thick with mixins and callbacks; led the design of order versioning and revision-proposal as first-class state (one party proposes a change, the other agrees, full revision history retained), in a model that also modernized how orders and items were represented; cutover by dual-write under feature flag
- Migrated legacy order operations (Submit, Cancel, Accept) from a metaprogramming-heavy task system to isolated, testable service classes; because static analysis could not trace the metaprogramming, injected ActiveRecord-level tracking in production to map every attribute mutation before consolidating into single documented code paths
- Led Datadog instrumentation with Four Golden Signals metrics, dashboards, and SLOs; surfaced row-level locking on the orders table as a system-wide bottleneck and produced the data that justified platform investment; held a <5s p99 SLO for legacy order operations through the migration

Technical Leadership & Operations (2019–2025): Owned technical direction well beyond my own squad while staying in the code.

- Go-to consultant across teams on architecture and system design; division representative on the internal Standards and Best Practices working group for ~2 years; authored numerous RFCs and design docs reviewed alongside Staff and Principal engineers
- Evangelized the platform org-wide: reference docs, onboarding guides, lunch-and-learns on internals, and direct pairing with adopting squads, whom we treated as our customers
- On-call throughout tenure across order management and platform components; advocated for instrumentation as part of the V0 definition of done; mentored a junior engineer on another squad, helped stand up a company-wide mentoring program, and ran technical interview rounds against rubrics

Blinker, Inc.: Senior Software Engineer, API

Denver, CO · October 2014 – March 2019 (promoted August 2018)

4th permanent engineer and first non-founder backend developer (~employee #12) at a consumer fintech startup building a mobile app for car valuation, P2P vehicle sales, and auto loan origination and refinancing. Helped take the API from proof-of-concept to production V1, then drove the rearchitecture from REST to GraphQL, personally building large fractions of the V1.x backend as the company grew from 12 to 50 people.

Lending Lifecycle & Third-Party Data (2014–2019): Jointly responsible, with three other API engineers, for the entire backend behind the product's launch.

- Modeled the full auto-loan lifecycle as explicit state machines, building regulatory approval gates and audit trails into the domain model rather than bolting them on; worked side by side with the company's lending and compliance specialists, translating their requirements into schema and state transitions and their edge cases into tests; delivered loan origination and P2P vehicle sales across five regulated state markets
- Integrated Plaid for identity verification and ACH money movement, DocuSign for e-signatures, and vehicle data APIs for VIN lookups and valuations, normalizing inconsistent and frequently wrong third-party data into a model the product could trust
- With three months to launch, co-led an emergency migration off a brittle screen-scraping loan-decisioning integration to a direct vendor API, eliminating a recurring source of outages

API Rearchitecture & Developer Tooling (2015–2018): Co-designed the GraphQL schema and API conventions with iOS, Android, and React engineers, and wrapped the then-immature graphql-ruby gem in Rails-like conventions so engineers new to GraphQL could contribute correctly. Earlier, co-designed an in-house per-branch deployment system on Nomad and Consul, before off-the-shelf Kubernetes tooling had matured, teaching myself Python to write its developer-facing CLI.

Brain Hurricane, LLC: Senior Developer

Chicago, IL · October 2011 – October 2014 (promoted June 2014)

Sole developer at a bootstrapped K-8 tutoring company, inheriting the legacy PHP application that ran the entire operation three days after joining, when the only other developer quit without notice. Led a ground-up Rails rewrite while keeping the legacy system in production, reimplementing every core business function: tutor scheduling and time tracking, payroll generation, student records, and progress reporting.

- Worked directly with the non-technical admin team to automate their daily pain points, including a Twilio click-to-call scheduling UI that replaced sequential manual phone calls
- Ran both systems simultaneously through an incremental cutover, each proactively exchanging data with the other: an early lesson in choreographed data consistency across systems

Anime Central (MAPS): Junior Systems Administrator / Programmer

Rosemont, IL · May 2007 – May 2011 (volunteer)

IT infrastructure and web development for a 23,000-attendee annual convention. After an outage damaged the registration system's reputation, built fault-tolerant on-site infrastructure from donated hardware on a zero budget, meeting a ten-second total-downtime target across the four-day event through manual hot-standby failover (matched MAC addresses, MySQL replication, a cable swap for invisible cutover). Also co-built the registration and payment application (PHP, Zend, MySQL).

EARLIER EXPERIENCE

Apple, Inc., Genius (*Northbrook, IL · 2009 – 2011*): component-level Mac and iPhone diagnosis and repair, warranty coverage and exception decisions. **TekSystems, Contractor** (*Schaumburg, IL · 2009*): maintained a large custom web application (PHP 4, JavaScript, Oracle 11) for an international pharmaceutical manufacturer. **Accenture, Analyst** (*Chicago, IL · 2007 – 2009*): enterprise web application projects for Fortune 500 clients, building UI components, business services, and batch processing in Java EE 5, Spring, Hibernate, JSP, and SAP environments.

EDUCATION

University of Wisconsin – Madison: B.S. Computer Engineering · December 2006