José Miguel Hernandez

Staff Software Engineer | Applied AI & Agent Platforms

info@jose.io
831-818-5505
github: jose-m-hernandez
linkedin: josemiguelhernandez
Morro Bay, CA (remote)

I make AI agents safe to run and worth using across an engineering org: platform, guardrails and adoption. At AdRoll, I classified our internal LLM agent as critical-tier and built its kill switch, its CI gate on unclassified write endpoints, and PR provenance auditable against our SOC 2 controls. 30+ colleagues, from the CTO to customer support, used it in its first month to resolve support and triage issues faster. I wrote the rules our shared agent skills repo is built to, planned migrations across 28 repositories with agents, and gave 800+ repositories a graph of what depends on what, who owns it and who to ask.

EXPERIENCE

AdRoll (NextRoll), San Francisco · Remote **Jun 2017 – Jul 2026**

Staff Software Engineer

Hired as Senior Software Engineer, promoted to Staff. My work: shortening the path from sign-up to first ad to first spend. Took PagerDuty on-call shifts and supported the systems I owned at night and on weekends.

Agent Safety and Internal Agents

- Owned the safety and governance layer of our internal LLM agent, built on the Claude Agent SDK, and became its second-largest contributor. Turned our AI agent security policy into a shared skill that classifies any agent by risk tier, and wrote classifications for four systems, including that agent, rated critical-tier under the lethal trifecta model (private data, untrusted input, ability to act). Built its mandatory kill switch in its Python broker, disabling writes while read-only investigation kept working, and a CI test that fails on unclassified write endpoints, which caught four new endpoints as the broker grew. Added requester-and-rationale provenance on agent PRs, auditable against our SOC 2 controls.
- Dozens of colleagues used the agent, from the CTO, SVP of Engineering and VP of Product to design, data, support, business systems and security. In its first month, 47 of its pull requests from Slack merged across five repositories. Closed two security-review findings and required signed Slack requests.
- Built the intelligence layer that allows our internal agent to answer who owns what from the service catalog, ownership records and org chart, promoting my prototyped skills into the shared repo.
- Largest contributor to our shared agent-skills repo. Wrote its eval-before-merge rule (each new skill's PR shows an invocation, input, output and pass/fail judgment) and its linter-enforced S/M/L sizing convention, which 125 of 128 skills carry. Audited the repo against the SVP of Engineering's nine AI principles, wrote its auto-invoke policy, and disabled auto-invoke on 54 skills.
- Evaluated skills against our real codebase before promoting them, re-running after each fix until clean.
- Audited skill-use telemetry for skills that never fired or ran on too small a model, and tested our internal agent by hand before and after every change to it or the shared skills.
- Took the Gong, Jira, Zendesk, Hex and Cube insight dashboard our Head of Product and Business Operations prototyped to deploy-ready in a week: moved its credentials to our vault, added tests and CI, purged customer data from its docs, and brought it into our org. Turned its integrations into shared agent skills, allowing support and product to run its analysis straight from Slack the next day.
- Defined the SOUL.md and wrote most of the guardrails and rapid-prototyping skills for Monet, a Slack design agent that built and deployed UX prototypes, bound to the design system's tokens, patterns and precedents.

Agent Platform and Developer Tooling

- Grew a dependency crawler into AdRoll's memory of its code, owners: metadata on ~800 repositories, refreshed daily, that coordinated migrations and releases, from version upgrades and codemods to releases behind feature flags and flag retirements, with LLM automation on top of CI. Reduced its drift points (dependency lag, code ownership, repository-to-Jira mappings) and used agents to keep the rest current. Served it over a Node.js MCP server with seven tools, so "who owns this?" took seconds instead of an afternoon, then moved the lookups into our shared Slack agent's skills for developers, designers and product people, who used them to gauge a design-system change's distribution friction.
- Pulled that system together in the React 19 Hack Week project, Overall runner-up in May 2026: added codemod capabilities to the PR pipeline, translated the dependency chains into Jira epic cohorts, and documented, sequenced and made fully transparent the entire scope of the coming migrations. Sized each task by who should make the change: a human with review, a codemod, or an LLM. Presented the plan to management as their decision whether to run it.
- Built the design system's canonical source and implementation gates so AI-assisted prototypes stayed aligned with downstream teams' components. A full-product Playwright snapshot gallery checked visual drift.
- Made design-system prototypes reviewable per pull request: a branch became a live preview URL inside its PR. Wrote the Terraform for keyless GitHub Actions deploys to AWS (OIDC federation, IAM trust policy, ECR), onboarded the repo to our Kubernetes (EKS) preview platform, and fixed the platform's cleanup job so PR activity kept an environment alive.
- Audited all 815 AdRoll production feature flags with a custom API client and corrected 300 mislabeled statuses. Gave each owning team a view of its stale flags and presented the results at AdRoll's Global Town Hall at the CEO's request. Cut our feature-flag vendor's seat count nearly in half from one Slack conversation with our internal agent, checking seats against team and ownership data I centralized.

Applied AI and Ad Products

- Built the bridge that let AdRoll move ad creation to the new ads service without a cutover: bootstrapped our shared library for managing ads, and created the service's embed view, the iframe message bus that let the legacy app host the new UI, and the old and new ad data models with the conversion layer between them.
- Revived Create Web Ads, shelved because its first version only cropped one image into six ad sizes: rebuilt it as a templated builder for our six most popular sizes, then followed advertiser feedback and usage into animation, more templates, and brand-specific AI copy and images, so advertisers could make on-brand ads in one click with no designer. By fall 2025, about 250 advertisers were spending on ads built with it.
- Prototyped an AI ad builder that generated a templated HTML5 ad from a customer's site data in one shot.
- As an early partner for OpenAI's ads API before it was public, in two days mapped OpenAI's beta Ads Manager onto our ads and assets models from screenshots, wrote backend, frontend and data-model specs checked against our codebase, and built a working prototype on a mocked API. It proved the integration feasible and unblocked planning.
- Took a colleague's Media Planner prototype to production as an internal AI media-planning assistant for sales reps preparing pitches: built its front end and tool integration and wired in its planning heuristics. Wrote and ran its test protocol, checking every prompt edit against three sample briefs and a pass/fail checklist, and wrote its rule that the agent never invents a number when benchmark data is missing. Checking the prompt against a live API response turned up eight schema mismatches and drove the next iteration.
- In 2026, in the ads service's Python backend (Flask, SQLAlchemy), wrote the data migration correcting about 14,000 misclassified ads, and triaged all 100 unresolved Sentry issues into six tickets in a day. Diagnosed an N+1 query affecting 309 users and fixed three other bugs, including a silent failure that stopped syncs for whole advertisers and a label-creation race behind 11,000+ error events.
- Enabled WebP creatives across our ad platform and its pipelines: added WebP asset handling to the ad sync pipeline, removed its WebP-to-PNG fallback, and fixed the transparency loss it exposed in our Go image service, a hardcoded background in the govips C bridge.

Technical Leadership

- Ran Front End Core from 2018 through 2026 and spearheaded web performance and observability for AdRoll's customer-facing apps: set the Core Web Vitals SLA, drove Datadog real-user monitoring across the product teams' apps, and ran the remediations that reduced page load times across many of them. Wrote the first Enzyme-to-RTL test conversions.
- Wrote the AI-Native Creatives roadmap and tied it into the 2026 company plan: mapped the team's AI work to company-level bets across all three themes, and turned product goals into specs and estimates of about 76 engineering-weeks.
- Produced NextRoll's twice-yearly Hack Week 14 times, placed in seven, winning four, and mentored five interns on the Ads and Front End teams.

TastemakerX → Rdio → Pandora, San Francisco **Mar 2013 – Mar 2017**

Senior Software Engineer

Rdio acquired TastemakerX in 2014, and Pandora acquired Rdio's assets in 2015.

- Built Pandora's integrations for smart TVs, speakers, Windows 10 and Uber for its 80 million active listeners, including account linking in Uber's app.
- Shipped on-demand UIs for the Pandora Premium launch, adding repeat, shuffle, and queue controls for Samsung, LG, Fire TV, Google TV apps, and Chromecast, profiling performance on legacy hardware.
- Architected and built the lab WiFi rig that verified cross-device communication and playstate handoff.
- Managed Rdio's Roku partnership and built account-linking and subscription-billing flows.
- Built Rdio's shutdown export, letting listeners download their collections or transfer stations to Pandora, with a personal listening-history timeline.
- Relaunched TastemakerX's music-discovery client as a Backbone single-page app, building activity-feed aggregation, a media-embedding engine, responsive UI, and Facebook onboarding. Co-built and shipped convertmyplaylist.com, a playlist converter from Spotify to Rdio.

Telly (formerly TwitVid), San Francisco **Dec 2011 – Sep 2012**

Lead Front-End Engineer

- Led TwitVid's front-end redesign into Telly, rebuilding its PHP app as a hybrid SPA with History.js navigation.

SKILLS

Applied AI · AI agents, large language models (LLMs), agentic workflows, Claude Code, Codex, Cursor, Claude API and Agent SDK, Gemini, local LLMs (Ollama), MCP, tool calling, prompt engineering, evals, guardrails, provenance

Developer Tooling · CI/CD and merge gates (Buildkite, GitHub Actions), dependency and ownership graphs, migration automation, codemods, MCP servers, agent skills, feature-flag hygiene (Split), Node.js, shell scripting

Languages · Python, TypeScript, JavaScript, Go, Java (JSP), PHP, SQL, GraphQL, Swift, Objective-C

Backend and Data · Flask, Django, SQLAlchemy, Postgres, OAuth, payments, billing

Infrastructure · AWS (S3, ECS, ECR, IAM), Terraform, Docker, observability (Datadog, Sentry)

Testing · Playwright, Jest, React Testing Library, Appium (device labs)

Frontend and Design Systems · React, Next.js, Redux, Webpack, design systems, component libraries, Figma, design tokens, accessibility

EDUCATION

University of California, Santa Cruz · BA, Film and Digital Media