

# Blake Enloe

## Principal Engineer | Full-Stack, Realtime, AI, and Business Automation Systems

(979) 575-1540 [blake@blake-enloe.com](mailto:blake@blake-enloe.com) [blake-enloe.com](http://blake-enloe.com) [linkedin.com/in/blake-enloe](https://linkedin.com/in/blake-enloe)

---

### PROFILE

Software Development Lead with 10+ years building production business systems for ISPs, public agencies, and AI-enabled operations teams. I turn ambiguous business and operations problems into technical solutions without the fluff: automate first, ship fast, and focus on work that raises margins, reduces manual drag, and increases profit. I build automation-first business solutions: internal tools, platform integrations, workflow engines, AI assistants, computer-vision pipelines, migration workbenches, support operations, and realtime dashboards that remove manual steps, expose bottlenecks, and make teams faster.

### CORE STACK

<b>Frontend</b>	TypeScript, React, Next.js, Tailwind CSS, shadcn/ui, R3F/WebGL	<b>Backend</b>	Node.js, Fastify, Express, Python/FastAPI, REST, GraphQL, WebSockets, Pub/Sub
<b>Data</b>	PostgreSQL, MySQL, MongoDB, Redis, BigQuery, Supabase, SQLite, data modeling	<b>Cloud</b>	Google Cloud, AWS, Docker, CI/CD, Vercel, Cloud Run, MongoDB Atlas, Netlify
<b>AI/Tools</b>	OpenAI API, Claude, n8n, YOLO/computer vision, custom chat assistants, AI model integrations, platform integrations, automation-by-default workflows	<b>Product</b>	Operations-first UX, automation by default, RBAC, auth, accessibility, observability, technical leadership, mentoring

### EXPERIENCE

#### Nextlink Internet

2022 - Present

##### Software Development Lead

- Lead architecture and delivery across ISP product surfaces: internal ticketing, customer account/billing/network tools, realtime collaboration services, auth, and operational workflow automation.
- Architected **Astral**, a Node/Fastify realtime backend with WebSockets/Socket.IO, Redis/MongoDB sync, ticket rooms, presence, cursors, inter-app messaging, push notifications, and provider integration endpoints.
- Built and iterated the **Nextlink Ticketing Platform**: queue triage, shared ownership, escalation flow, internal notes, live collaborator presence, role-aware actions, linked account signals, and AI-assisted support workflows.
- Led customer-facing **Nextlink Connect** work across web and mobile: account status, billing history, network health, alerts, RBAC, legacy API abstraction, Clerk auth, and customer verification gating against billing data.
- Migrated auth patterns from AWS Cognito toward Clerk in App Router products, standardizing user metadata, claims, session boundaries, and downstream service contracts.
- Raised team delivery quality through TypeScript/Next.js standards, OpenAPI documentation workflows, CI checks, code review, implementation planning, and mentoring.

#### Brazos County Government

2019 - 2022

##### Full-Stack Developer

- Delivered 11 departmental websites and an integrated CMS in 6 months, giving non-technical staff a consistent publishing system and residents a clearer public-sector information hub.
- Led rapid development of a COVID-19 vaccine appointment scheduling system that helped facilitate 100,000+ community vaccinations after being given 7 days' notice by government stakeholders to have it deployed and functional.
- Modernized the main county website for accessibility, mobile UX, content consistency, keyboard navigation, screen reader support, and a growing, diverse population of 200k+ citizens and users.

#### ISP Supplies

2014 - 2019

##### Software Developer

- Automated NetSuite vendor-specific price-level calculations for resellers and distributors, eliminating hundreds of hours of manual entry and improving accuracy across 50k products.
- Built custom JavaScript/SuiteScript workflows to streamline ERP processes, reduce operational errors, and support day-to-day WISP business operations.
- Created React/SuiteCommerce CMS and landing-page tooling that reduced dependency on external agencies and gave internal teams faster campaign execution.

# Blake Enloe

## Selected Engineering Projects

---

### SELECTED ENGINEERING PROJECTS

---

#### Living Frame

Electron, Next.js, SQLite, OpenAI media generation, ffmpeg, local media pipeline

Scaffolded a local-first desktop art appliance that generates animated paintings, stores state locally, prepares playback media, archives generations, and keeps a display running safely from the last known good asset when network or generation calls fail.

#### Nvidia Remote Asset Tracking + AI Assistant

React, Next.js, BigQuery, Grafana, QR workflows, geospatial dashboards, AI assistant

Designed a remote lab asset dashboard with map-based tracking, live metrics, QR check-in workflows, offline-friendly field interactions, bulk admin tools, and a BigQuery-backed AI chat assistant for capacity, inventory, and operational recommendations.

#### GameCam Wildlife Monitor

Node.js ESM, Moultrie Mobile APIs, Animal Detect, Discord alerts, stateful backfills

Built a service that polls shared camera collections, runs wildlife detection, normalizes Texas-relevant species policy, posts alerts, and protects backfill integrity with isolated logs and retry-safe state handling.

#### Infara UI Registry + Consumer Bootstrap

Next.js 16, React 19, shacn registry, 125 installable items, MCP server/client, create-infara CLI

Built the installable frontend platform for AI-first ISP/WISP operations products: 64 UI components, 59 workflow blocks, shared themes, helpers, app-shell patterns, pinned release artifacts, native MCP tooling, and a bootstrap CLI that scaffolds downstream Next.js apps with starter UI, brand assets, docs, agent instructions, MCP wiring, and build verification.

#### Developer Workflow and QA Automation

GitHub Actions, OpenAPI docs, n8n, CLI workflows, validation scripts, repo audits

Regularly builds tools and guardrails around delivery: docs bots, route parity checks, security/dependency audits, test fixtures, smoke validation, platform integrations, and automation-first workflows that turn ambiguous work into requirements, implementation, QA, and review loops.

#### Duct Takeoff + HVAC Symbol Corpus

Next.js, Python/FastAPI, vector PDF parsing, ASHRAE 134 corpus, estimator workflow

Built an HVAC estimating workflow for mechanical binder PDFs and a provenance-preserving symbol corpus pipeline. The approach emphasizes vector-first parsing, confidence boundaries, reproducible manifests, metadata enrichment, and estimator reviewability.

#### Shopify Scripts Migration Workbench + script-factory

TypeScript, Shopify app workflows, migration QA, embedded app flows, cutover reports

Built migration tooling for moving legacy Shopify Scripts into reviewable Function extensions. The workbench handles official CSV/raw Ruby intake, normalized migration models, fixed Function deployment, QA guidance, and cutover-ready reporting while script-factory creates corpus evidence, prototypes, and reusable conversion documentation.

#### Starlink Atlas

Next.js 16, React, TypeScript, WebGL, MapLibre, realtime orbital data

Built a live Starlink visualization that animates satellite ground positions from public TLE data, keeps map motion stable between updates, and shifts into a local-sky workflow when geolocation and zoom context make it useful.

#### AI Scrum Assistant for Jira

Atlassian Forge, TypeScript, React, OpenAI API, prompt guardrails

Created a Jira assistant that generates user stories, acceptance criteria, BDD criteria, test cases, and refinement artifacts directly inside the issue workflow. Designed it as a suggestion-and-review tool, not a black-box authority.

### WHAT I TEND TO OWN

---

- **Business and operations problems:** support platforms, dashboards, internal tools, portals, migration flows, and admin surfaces where automation, margin, and workflow clarity matter.
- **Realtime and state-heavy UX:** presence, collaboration, field locking, notifications, live queues, event streams, and multi-user product behavior.
- **AI that stays useful:** custom chat assistants, AI model integrations, computer-vision workflows, generators, migration aids, and automations that preserve human review while removing repetitive work.
- **End-to-end delivery:** requirements, architecture, implementation, testing, build/deploy hygiene, documentation, and stakeholder-ready explanations.