

MARKO NOGIC

markonogic1@gmail.com • linkedin.com/in/markonogic • markonogic.com

SOFTWARE ENGINEER — FULL STACK

Software engineer with a professional full-stack foundation shipping production Java/.NET systems for Southwest Airlines operations, followed by six years leading engineering delivery inside Toyota, Paramount, and Goldman Sachs — writing code throughout. Currently building: a native Swift 6/SwiftUI streaming client shipped to TestFlight beta and a full-stack React/TypeScript + FastAPI/PostgreSQL collection-management platform, with a companion Go media server in early development. Full-stack profile with strengths in client architecture, API and contract design, media playback, Linux infrastructure, and AI-native development workflows.

TECHNICAL SKILLS

Languages: Python, SQL, Java, C#/.NET, TypeScript/JavaScript, Swift, Go, Bash

Backend & APIs: REST API design and versioned OpenAPI contracts, FastAPI (Pydantic, SQLAlchemy), generated TypeScript API clients with CI drift checks, microservices, session/auth design, streaming media pipelines (HLS/fMP4, FFmpeg/ffprobe orchestration, HDR/Dolby Vision)

Data: PostgreSQL, SQLite, MS SQL Server, Oracle; schema design, reporting/analytics pipelines

Frontend: React 19 + TypeScript (TanStack Query/Virtual, React Router, Tailwind, Vite), SwiftUI (MVVM/Observation), hls.js

Infrastructure: Linux, Docker/Compose, CI/CD (Jenkins, Azure DevOps), Git, Tailscale/Cloudflare, local LLM inference (NVIDIA RTX 3090)

Testing & Quality: Golden-fixture corpora, contract/route-parity tests, process-supervision fakes, output-conformance verification, Playwright, unit/integration testing

AI-Native Development: Spec-first, test-verified agentic workflow — product requirements and architecture docs plus BDD test plans gate implementation; parallel Git worktrees and tmux sessions; multi-model implementation and skeptical review (frontier + local Qwen models); every change compiled, tested, and human-reviewed before merge

ENGINEERING PROJECTS

Moonberry Client — Creator — Swift 6, SwiftUI, AVFoundation (tvOS)

2025 – Present

- Designed and built a native Apple TV streaming client as a thin app target over modular Swift packages (DesignSystem, MediaCore, JellyfinKit, Features) — MVVM on the Observation framework with per-screen @Observable view models, typed value-based navigation carrying lightweight IDs only, and an immutable composition-root dependency graph.
- Built a fully stateless, previewable design system: components consume plain view-data structs and have zero network or service knowledge, keeping every component reusable across future backend services; a single MediaCard owns all focus and overlay behavior behind thin episode/person/collection wrappers.
- Implemented the complete auth and session lifecycle: username/password and poll-to-authorize Quick Connect sign-in, Keychain token storage, GRDB-persisted sessions auto-restored on launch, and a SessionMonitor that catches mid-session 401/403s and returns the app to sign-in.
- Built playback on AVFoundation/AVPlayerViewController with server-side PlaybackInfo negotiation (direct play vs HLS transcode), resume-from-position, and periodic progress reporting; plus home shelves (Continue Watching, Next Up), debounced server-side search, and shared detail scaffolds across all media types.
- Shipped a TestFlight beta (v0.5.0) with custom branding and design palette, a release runbook, and a BDD beta test-plan catalog; currently reworking playback for HLS variant selection, Dolby Vision device profiles, and chapter/skip-intro markers.

MTG Vault — Creator — React 19/TypeScript, FastAPI, PostgreSQL

2026 – Present

- Built a full-stack Magic: The Gathering collection manager and deck builder: React 19/TypeScript frontend (TanStack Query/Virtual, React Router, Tailwind, Vite) over a Python FastAPI + Pydantic + async SQLAlchemy + PostgreSQL backend — 44 versioned API operations, ~73 Pydantic models, and generated OpenAPI→TypeScript contracts enforced by CI drift checks.
- Ingested Scryfall's bulk dataset into a locally searchable catalog of 534k+ printings (38k+ distinct cards) through a scheduled sync worker with leasing, heartbeats, retries, streamed downloads, and staged validation before catalog activation.
- Engineered server-side collection browsing at scale: composable filters across name/rules text, sets, types, mana value, and color inclusion/exclusion/exact matching; keyset cursor pagination; URL-owned filter state; and virtualized grid/table rendering.
- Built a customizable analytics dashboard — collection value with historical reconstruction from timestamped price observations, price movers, and recent activity — with drag-and-drop, keyboard-reorderable widgets, saved layouts with optimistic versioning, and explicit partial/unavailable-price states; plus a Ctrl-K command palette searching cards, binders, and decks.
- Modeled inventory as append-only transactions with idempotency keys and compensating undo; deck building uses explicit desired/owned/assigned/free/missing semantics with exact-printing assignment, completion validation, and double-count-safe collection valuation.
- Implemented first-party authentication and session management — hashed credentials, opaque server-side sessions, CSRF protection, and per-owner data scoping — plus 32 Playwright scenarios across 14 specs: full Docker Compose-backed E2E lifecycles, WCAG A/AA accessibility checks, reduced-motion behavior, and visual-regression baselines.

Moonberry Media Server — *Creator* — *Go, SQLite, FFmpeg (early development)*

2026 – Present

- Designing and building a self-hosted media server in Go: a REST API under a versioned OpenAPI contract, signed expiring session-scoped URLs for header-less HLS player auth, and a playback-decision engine that negotiates direct-play, remix, or transcode plans from device-capability documents and fprobe media descriptions — FFmpeg-orchestrated HLS/fMP4 delivery with HDR/Dolby Vision handling, backed by golden-fixture and route-parity tests.

Homelab & Local AI Infrastructure — *Linux, Docker, NVIDIA*

2016 – Present

- Operate a Debian homelab running containerized media services and local LLM inference on an RTX 3090, with Tailscale-based network segmentation separating exposed and internal services; currently standing up a local AI code-review flow in which self-hosted Qwen models perform independent, skeptical review passes on agent-generated diffs before human review and merge.

PROFESSIONAL EXPERIENCE

Toyota (via K&B Global Services) — *Agile Delivery Lead / Technical Program Manager*

Jan 2025 – Present

- Built internal delivery-analytics tooling end to end — Python and SQL over the Jira REST API with a React frontend — restoring sprint, quarterly, and scope-change visibility for VP leadership after prior tooling was lost; now used daily by teammates.
- Developed eazyBI/MDX reporting and Jira automations with advanced branching, including debugging report timeouts and automation edge cases; sole delivery lead for 3 engineering teams (~21 people) building microservice-based dealership platform applications.

Paramount Global / Pluto TV — *Technical Program Manager II*

Jul 2022 – Oct 2024

- Owned delivery of the Pluto TV web platform across 3 squads (24 engineers); partnered with engineering leadership on the Kubernetes migration, an AWS Personalize recommendation engine, and raising code coverage from 45% to 75%.
- Drove the release-cadence transformation from monthly to roughly biweekly (7 → 24 production releases per year) and improved objective delivery from 63% to 88%.

Southwest Airlines — *Senior Scrum Master / Technical Program Manager*

Oct 2021 – Jul 2022

- Authored API contracts and system designs for features on the cloud migration of the mission-critical crew-scheduling platform, bridging product and engineering gaps and influencing stack choices for long-term maintainability; built Python/SQL/Jira REST reporting that replaced manual initiative tracking.

Goldman Sachs / Marcus — *Scrum Master / Technical Program Manager*

Feb 2020 – Oct 2021

- Defined the technical requirements for programmatic integration with FEMA's API for the disaster-eligibility email program, shipped under a compliance deadline in under one month and averting significant regulatory exposure; served as the primary technical liaison with Apple across multiple programs, presenting directly to the VP of Apple Wallet.
- Built daily Python/SQL program reporting across 3 teams; partnered with an AI/ML team on requirements and integration planning for an in-house fraud-detection model replacing a third-party vendor.

Fenway Group @ Southwest Airlines — *Software Engineer → Tech Lead / Scrum Master*

Jan 2016 – Feb 2020

- Developed and maintained full-stack Java (Struts, Maven) and C#/.NET applications supporting Southwest operational systems classified Critical-to-Business, Required-to-Operate, and Required-to-Fly.
- Built REST API services supporting a new COTS platform, collaborating with Solution Architects to define API contracts and shared development standards across teams.
- Extended a database-agnostic scheduled reporting system (YAML-configured SQL, cron scheduling, pluggable DB connectors) bringing ~20 automated reports into production, many of which remained in production for years after rollout.
- Replaced manual drag-and-drop .NET releases with a Python-based deployment pipeline, cutting release errors and promotion time; migrated source control from ClearCase to Git and modernized a legacy Java framework, reducing merge conflicts.
- Built Jira REST API metrics tooling tracking velocity and contributor throughput; as Tech Lead, led a 7-person dev team, coordinated a 5-person DevOps team, and cut cycle time ~30%; mentored associate engineers and delivered an Agile immersion workshop that helped 50+ participants earn PSD certification.

EDUCATION & CERTIFICATIONS

Louisiana Tech University — B.S., Computer Science

2013 – 2018

MIT Sloan School of Management — Managing Technical Professionals and Organizations (Executive Education)

2019

Professional Scrum Developer (PSD, 2017) • Professional Scrum Master (PSM, 2018)