Jason Michael Wergin

Birmingham, AL, USA
205.643.8580 | jmw@jasonwergin.com
https://jasonwergin.com/ | https://linkedin.com/in/jason-wergin/

SUMMARY

Principal-level software architect specializing in Framework Development and Codified Architecture — 23 years building custom application frameworks whose structural discipline produces measurable durability (frameworks I've built are still in production 5-23 years after deployment). Currently applying the same discipline to AI-native development, shipping production code with structural correctness enforced by design rather than by discipline. Deep expertise in C#/.NET and modern TypeScript/Node/React on cloud-native platforms (GCP, Azure). Specialties: distributed and event-driven architecture, microservices, API and platform design, domain-driven design, and AI-native development. Hands-on architect who still ships production code, with a long record of mentoring engineers and setting engineering standards across teams.

SIGNATURE PRACTICES

Framework Development (23 years) — NCP inheritance framework (2003), EBSCO distributed-compute framework (2004), White Plume workflow framework (2007), ibml application framework (2018), Ticket Dawg platform (2021–present), among others.

Codified Architecture (23 years) — Building the architecture into the structure of the code so its preservation doesn't depend on any individual developer's understanding. Prevents drift (small compromises by developers who don't fully see the whole pattern) while enabling deliberate architectural evolution when real requirements exceed the current pattern's scope.

SKILLS

Languages & Frameworks: C#, .NET / .NET Core, TypeScript, Node.js, Next.js, React, JavaScript, SQL, Java, C++

Architecture: Distributed Systems, Microservices, Event-Driven Architecture, Pub/Sub, Domain-Driven Design (DDD), CQRS, API / Platform Design, Design Patterns, SOLID, Multi-tenancy, Anti-Corruption Layer, E-commerce, Catalog & Inventory Modeling

Cloud & DevOps: Google Cloud Platform (GCP), Cloud Run, Azure, Docker, Kubernetes concepts, CI/CD, GitLab, GitHub, Terraform concepts

Data: PostgreSQL, SQL Server, MongoDB, MySQL, MariaDB, Neo4j, Entity Framework, Dapper

AI: AI-native development (intensive production since November 2025), constrained-AI code generation via codified architecture, Claude Code, Anthropic API, prompt engineering, retrieval-augmented generation (RAG), agentic retrieval, structured outputs, LLM application architecture.

Practices: Scrum / Agile, code review, mentoring, technical leadership, testing automation, observability (custom instrumentation, metrics, alerting, health monitoring)

PROFESSIONAL EXPERIENCE

Founder & CTO, Ticket Dawg, LLC — Birmingham, AL
April 2021 to Present

- Own and run a revenue-generating software company with a business partner — owning architecture, product direction, client delivery, and budget across a growing suite of production applications and shared microservices serving multiple paying clients.
- The codified architecture across the entire product suite — 12 applications on the same framework, service, repository, and CTE layers — has held with zero drift. Cross-cutting concerns are updated once in the blanket layer and applied everywhere. When patterns need to evolve, we redesign, refit, and rebake the whole ecosystem — a sweeping change AI-native development now makes achievable in days rather than months — I haven't hand-written code since November 2025; the architecture documentation is the prompt.
- Build a fully decoupled, event-driven Pub/Sub architecture with recursive provider routing and health-based trust scoring (hot/warm failover mesh, "fast to distrust, slow to re-trust") — swap email providers mid-peak-season in 10 minutes via ACL-based interfaces with zero downtime.
- Create the Blanket Pattern — a composition pattern that cuts API endpoint boilerplate from 40 lines to 8 and enforces consistent handler architecture across the entire codebase.
- Architect a multi-tenant e-commerce platform (TypeScript/Node, Next.js, React, PostgreSQL, GCP Cloud Run) — 130,000+ tickets sold and redeemed, sustaining ~5 orders/minute at seasonal peak with checkout and live redemption running concurrently.
- Design the catalog and inventory data model — product types, tiered pricing, time-windowed availability, and a "ticket drop" allocation model using expiring database-level cart locks with stage-aware TTLs, reclaimed lazily by the next writer with no reaper service — structurally preventing oversell under concurrent checkout while keeping abandoned carts from stranding inventory.
- Design the un/trusted coding strategy and trust-line CTE architecture — treats user-supplied values as untrusted expectations until they cross a trust boundary at the database layer, where CTE-embedded permission checks structurally prevent unauthorized operations. Reduce permission-check overhead from ~800ms across multiple queries to ~30ms in a single query and hold every query under 80ms at peak, allowing the platform to run on the smallest database tier through peak season.
- Build a multi-tenant billing microservice with a hash-chained, tamper-evident audit ledger — invoicing, metered usage, subscriptions, refunds, and a QuickBooks accounting mirror with durable retry and reconciliation.
- Build a production observability layer from scratch — structured event instrumentation, real-time error-rate and throughput metrics (orders/minute), severity-tiered alerting, and provider health monitoring driving automated failover — surfacing system and business health in real time without third-party APM.

Founder, jasonwergin.com — Birmingham, AL
August 2022 to Present

- Ship a production LLM feature end-to-end: ACL-isolated provider, structured outputs, prompt-cached context, and an honesty-calibrated scoring rubric tuned against real job postings. Publicly available at jasonwergin.com/jobmatch — used by hiring managers and recruiters to evaluate open roles against my background.

Lead Software Engineer, Metrc, LLC — Lakeland, FL
July 2023 to Present

- Lead engineer on a regulated seed-to-sale cannabis-tracking platform (C#/.NET, React, Angular, SQL Server, MongoDB, Neo4j) used for government compliance and supply-chain traceability — driving architecture and code quality across my team.
- Built an AI-native requirements engine on a retrieval-augmented architecture — model-generated search terms over a typed context model, relevancy-scored code retrieval, and a codified format spec as structural constraint. Produces Jira tickets consistent enough to drive downstream automated test generation.
- Re-model a 3TB package-traceability dataset — multi-layered harvest and repackage lineage — from Azure SQL Server into Neo4j, replacing slow recursive SQL traversal with graph queries better suited to the hierarchical relationships.
- Own high-level software design, the code-release strategy, and code-quality standards across the team, and manage the Scrum process and task assignment.

Senior Engineer, Metrc, LLC — Lakeland, FL
March 2023 to July 2023

- Integrated Tableau reporting into the platform, giving state regulators deeper visibility into the businesses they oversee.

Senior Engineer, BLOX, Inc. — Bessemer, AL
July 2021 to March 2023

- Designed and built, with a team, the design, manufacture, and construct ecosystem serving BLOX and its customers, suppliers, and wholesalers — a React front end on an Express backend — working with the product manager to shape product direction.
- Modeled design, manufacture, and construct bills of materials (BOMs) as a complex Neo4j graph.
- Set up the initial CI/CD pipeline in GitLab and the containerization process for the application suite.

Architect, ibml — Birmingham, AL
June 2018 to July 2021

- Designed and led development of a codified application framework at ibml — enforced consistent architectural patterns across all subsequent applications and dramatically accelerated new-application delivery.
- Led a weekly meeting allowing developers to learn new design strategies. This allowed for consistent implementation of our architecture and software guidelines.
- Applied Domain-Driven Design to ibml's modular document scanners — a C# domain model that composed itself at runtime from the modules the C++ control layer reported, then drove and diagnosed the physical machine through repository-backed drivers with a loose CQRS split between reads and writes.

SELECTED EARLIER EXPERIENCE

A full historical résumé can be found at jasonwergin.com

Senior Engineer, DST, Inc. — Birmingham, AL
July 2016 to May 2018

- Modernized a physician-facing healthcare application (legacy ASP → Backbone/Web API) that let providers monitor their MIPS and Medicare Star Ratings — the quality scores that drive

reimbursement.

- Built an application enabling users to register for insurance.
- Mentored the development team in C#, JavaScript, Backbone, and .NET.

Founder & CTO, Nuyella, Inc. — Birmingham, AL
May 2011 to March 2013

- Set technical goals and direction for the company; ran sales, payroll, and billing.
- Designed a production method for rapid website development and release management.
- Built a back-office system tracking customer payments, publishing, and automated payment processing through Authorize.net.
- Managed a team building SEO-targeted websites.

Architect, White Plume Technologies — Birmingham, AL
October 2007 to March 2011

- Rebuilt the entire product base on a custom codified framework I designed — new applications shipped in 45 days instead of the previous 6-9 months. 18 years later, the framework is still in production. Where the architecture has evolved, it evolved through deliberate redirection, not drift.
- Could have adopted the first release of Windows Workflow Foundation (WWF), but it lacked the pluggability we needed — so I hand-wrote our own event-driven workflow engine in about three months, using the same fan-in/fan-out pipeline and activity model WWF would only mature into roughly a year later.
- Designed and built a working HL7 interoperability hub on a canonical intermediate format. HL7 is a standard in name - every vendor populates it differently - so the hub learned each vendor's dialect and mapped it to ours, letting any two systems exchange clinical data without either needing to understand the other's conventions. Mapped a leading PMIS with a second PMIS exchanging live data through it.
- Started and led Cornerstone, a weekly architecture meeting that gave the team cross-product knowledge White Plume had not had before. Combined with bringing in the Microsoft Solutions Framework, it earned the President’s Award — an annual award for work above and beyond expectations, presented with the remark that I had "single handedly changed how White Plume does business."

Senior Engineer, EBSCO Industries, Inc. — Birmingham, AL
April 2004 to October 2005

- Built a distributed-compute framework that harvested idle workstations — a client service detected machine lock, registered capacity with a coordinator, and received farmed-out work units. Utilities were written against a common framework contract, decoupled from the distribution layer; a second utility on the same framework drove idle machines into power-saving mode. Predated mainstream commercial cycle-scavenging.
- Replaced a content-loading system that imported fewer than five records a day with one handling 18,000+.
- Trained and mentored my peers in VB.Net and the .Net framework.

Architect, National Computer Print, Inc. — Birmingham, AL
April 2003 to April 2004

- Designed the migration of 800+ applications off the mainframe — projected at nine months, delivered in eight.
- Built a self-updating, web-deployed client that pulled down only the application packages a user's database security grants allowed — so staff ran the same apps identically from home or office. This beat mainstream ClickOnce-style deployment by three to four years.
- Architected a network-aware data layer on .NET 1.1: it queried SQL Server directly when local, or transparently proxied through a web service returning XML when remote — calling code never knew which path it took. Predated WCF and occasionally-connected smart-client patterns by years.
- Designed a dynamic, role-driven application browser covering the entire enterprise (shop floor, HR, customer service, mail room) — composing each user's available components at runtime from their permissions, using MVC, DTO, MSMQ, dynamic host discovery, and workflow.
- Drove all identity and authorization from the database with role- and column-level security — adding a SQL Server user provisioned them into the system, no separate user administration.
- Built the company's codified architecture framework — an abstract-base-class inheritance system that structurally enforced consistency across every application built on it. 23 years later the framework is still in production. Where the architecture has evolved, it evolved through deliberate redirection — never through drift.

EDUCATION

M.Eng. in Information Engineering and Management, University of Alabama at Birmingham, May 2013

B.S. in Computer Science (Minor: Mathematics), Jacksonville State University, December 1994

CERTIFICATIONS

Design Guidelines and Patterns for Microservices — SEI, March 2024

Documenting Software Architectures — SEI, March 2024

Software Architecture Principles and Practices — SEI, December 2023