

John Belthoff

Full-Stack Software Engineer • .NET / Angular • Security & AI Integration

Fort Lee, NJ • (860) 605-0328 • jb@johnbelthoff.com • angular.johnbelthoff.com

U.S. Citizen – Authorized to work for any U.S. employer

Professional Summary

Full-stack software engineer specializing in .NET, Angular, and secure infrastructure, with 15+ years modernizing legacy systems and building production applications from architecture through deployment. Designs and implements zero-trust security architectures including HashiCorp Vault integration (dynamic credential rotation, PKI trust injection, KV secret management), OWASP ZAP and WPScan penetration testing, and CSP nonce-based content security policies. Builds AI-integrated systems in production — RAG pipelines with PostgreSQL/pgvector, custom Model Context Protocol (MCP) servers for agentic AI tooling, and multi-section LLM prompt architectures with parallel async orchestration. Engineers for performance, achieving near-native C++ throughput in managed .NET with zero-allocation hot paths and BenchmarkDotNet-validated evaluation engines. Enforces architectural discipline through automated architecture testing, domain-driven design, and iterative AI-assisted code reviews.

Summary of Qualifications

- Full-stack .NET engineer with 15+ years of experience designing, building, and modernizing production web applications — from legacy ASP.NET Web Forms through modern .NET 10 / ASP.NET Core with Angular frontends.
- Production experience building RAG systems, MCP servers, and LLM-integrated applications in .NET/C#. Architected multi-section prompt frameworks with structured context injection, parallel async API orchestration, vector embeddings (pgvector), and encrypted AI data persistence. Built custom Model Context Protocol server enabling cross-repository AI context retrieval via semantic search.
- Designed and implemented complete HashiCorp Vault integration across four secrets engines — dynamic database credential rotation with zero-downtime lease renewal, PKI-based TLS trust injection, KV v2 static secret management, and AppRole authentication — with thread-safe credential lifecycle management and deterministic startup orchestration.
- Enforces production security through OWASP ZAP and WPScan and other penetration testing, CSP nonce architecture, SSRF prevention, timing-safe token comparison, host allowlist enforcement, and DOMPurify HTML sanitization with custom security hooks.

- Engineers for measurable performance: near-native C++ throughput in pure managed C# (zero-allocation hot paths, stackalloc/Span, AggressiveInlining), validated via BenchmarkDotNet with cross-language comparison benchmarks.
- Enforces architectural discipline through automated architecture testing (NetArchTest), domain-driven design with deterministic domain logic, multi-tier caching (in-memory L1 → Redis L2), and iterative AI-assisted code reviews (8-review progression from 72 → 91 production readiness score).
- Strong infrastructure background: Docker, NGINX reverse proxy with per-location security policies, multi-stage container builds, Cloudflare, PostgreSQL, SQL Server, and prior Secret clearance with DoD data handling across multiple security domains.

Technical Skills

Languages & Frameworks

C#, .NET 10 / ASP.NET Core, Angular 21, TypeScript, Razor Pages, MVC, JavaScript, HTML5, CSS / Tailwind CSS, Python, Node.js / Express, PHP, XML, SGML, VB.NET

AI & LLM Integration

RAG (Retrieval-Augmented Generation), pgvector, vector embeddings (OpenAI text-embedding-3-small), MCP (Model Context Protocol) server development, OpenAI Chat Completions API, OpenAI Embeddings API, Anthropic Claude API, multi-section prompt architecture, structured context injection, parallel async LLM orchestration, token usage tracking, encrypted AI data persistence, AI-assisted code review pipelines

Security & Penetration Testing

HashiCorp Vault (PKI, Database, KV v2, AppRole — dynamic credential rotation, runtime CA trust injection), OWASP ZAP (frontend + GraphQL active scanning), WPScan, CSP nonce architecture, SSRF prevention, DOMPurify (custom security hooks), TLS hardening (Mozilla Intermediate), timing-safe token comparison, AES encryption at rest, HMAC-SHA256 token hashing, X.509 custom trust stores

Databases & Data Access

SQL Server (2000–2024), T-SQL, PostgreSQL (pgvector extension, HNSW indexing, cosine similarity search), MariaDB, MySQL, Dapper, Entity Framework Core, Redis, stored procedures, multi-schema design

Performance Engineering

BenchmarkDotNet (MemoryDiagnoser), zero-allocation hot paths (stackalloc, Span<T>, ReadOnlySpan<T>), AggressiveInlining / AggressiveOptimization, Parallel.For with thread-local accumulators, cross-language benchmarking (C# vs C++), multi-tier caching (in-memory L1 → Redis L2)

Architecture & Engineering Practices

Clean architecture enforcement (NetArchTest), domain-driven design (DDD), deterministic domain logic, automated architecture testing, immutability patterns, CQRS-influenced data access (Dapper + EF Core), Syncfusion PDF report generation, Apollo Client / GraphQL (WPGGraphQL), headless CMS (WordPress as API)

Authentication & Identity

ASP.NET Core Identity, JWT (access + HttpOnly refresh tokens), HMAC-SHA256 refresh token hashing, Cloudflare Turnstile bot verification, Stripe payment integration, rate limiting, email confirmation flows, 2FA

Infrastructure & DevOps

Docker (multi-stage builds, non-root execution), NGINX (reverse proxy, per-location CSP, rate limiting, TLS termination), Docker Compose, Cloudflare, Linux (Ubuntu), Windows Server, IIS, Git / GitHub, CI/CD, Proxmox, VMware

Professional Experience

Independent Web Application Developer

2017 – Present

Selected engagements include long-term, hands-on engineering and architecture work for government, defense-adjacent, and regulated-industry clients; project and client details are redacted due to NDA and security restrictions.

- Designed and implemented complete HashiCorp Vault integration for a .NET API backend (2,650 lines across 10 classes) — dynamic PostgreSQL credential rotation with automatic lease renewal and 30-second drain for zero-downtime swaps, PKI-based TLS trust injection with custom X.509 chain validation, KV v2 secret management for JWT/SMTP/Turnstile credentials, and AppRole authentication with thread-safe token caching. Built deterministic five-phase startup with exponential-backoff retry and health check endpoints exposing Vault lease status.
- Built a RAG (Retrieval-Augmented Generation) system with PostgreSQL/pgvector providing persistent, searchable memory for AI coding assistants across multiple repositories. Developed C# embedding pipeline with hash-based change detection (SHA-256), batched OpenAI text-embedding-3-small integration, HNSW-indexed cosine similarity search, secret file detection, and

path-escape prevention. Created custom Model Context Protocol (MCP) server exposing four semantic search tools that Claude Code discovers and calls natively, enabling cross-repository context retrieval.

- Integrated OpenAI's Chat Completions API into a production .NET application (iching.rocks) as a paid premium feature, engineering a multi-section prompt architecture (8–14 concurrent requests per transaction) with domain-specific context injection, parallel async orchestration (Task.WhenAll), AES-encrypted request/response storage, and per-request token usage tracking for cost monitoring.
- Hardened an Angular 21 SSR application with CSP nonce architecture (randomBytes per request, placeholder-based cache strategy for unique nonces on cached responses), SSRF prevention (startup validation of upstream URLs against explicit allowlist), timing-safe token authentication, AsyncLocalStorage for concurrency-safe request isolation, and server-side DOMPurify sanitization with custom security hooks. Conducted OWASP ZAP and WPScan penetration testing with an 8-iteration AI-assisted code review pipeline progressing production readiness from 72 to 91.
- Achieved near-native C++ performance parity in a pure managed C# poker hand evaluator (~180–210M evaluations/sec, ~943 ns/op core). Implemented zero-allocation hot paths using stackalloc/Span, AggressiveInlining/AggressiveOptimization, and Parallel.For with thread-local accumulators. Developed a dedicated cross-language benchmark suite (BenchmarkDotNet) proving C# throughput within sub-microsecond margins of C++ MSVC /O2 AVX2.
- Enforced clean architecture across an 8-project .NET solution (~13,500 lines) using automated architecture tests (NetArchTest) validating dependency flow, domain immutability, and infrastructure isolation. Implemented domain-driven design with deterministic, side-effect-free domain logic, multi-tier caching (in-memory L1 → Redis L2 with automatic warming), BenchmarkDotNet performance profiling, and Syncfusion PDF report generation.
- Built JWT authentication system with ASP.NET Core Identity, HMAC-SHA256 refresh token hashing (raw tokens never stored), HttpOnly cookie-based refresh flow, Cloudflare Turnstile bot verification, rate limiting, and email confirmation — all secrets sourced from HashiCorp Vault at runtime via KV v2 secrets engine.
- Architected a headless WordPress/Angular 21 SSR system with WPGraphQL API, webhook-driven cache invalidation, preview authentication via custom REST endpoint, LRU HTML cache with nonce placeholder substitution, NGINX reverse proxy with per-location CSP policies, TLS 1.2/1.3, rate limiting, and Docker Compose orchestration.

Datalis Solutions Corporation

2010 – 2017

Lead Software Developer, System Administrator, Web Architect

- Led development and architecture of the Depot Tracking System (DTS) — a global logistics application tracking Electronic Warfare (EW) systems entering and leaving repair depots worldwide, built on ASP.NET / SQL Server.
- Designed and maintained a secure platform aligned with the DoD Integrated Defense Acquisition, Technology, & Logistics Life Cycle Management Framework.
- Implemented and enforced Unclassified, Confidential, and Secret data handling protocols across multiple security domains, ensuring full compliance with DoD cybersecurity and data protection standards.
- Administered Windows Server infrastructure, networking, and virtualized environments supporting high-availability deployments and global user access.
- Held and maintained an active U.S. Secret Security Clearance for the duration of the project.
- Consulted with Technical Publications on SGML and Arbortext best practices for multi-output technical documentation pipelines.

Perry Systems, Inc.

2007 – 2010

Web Architect, Full Stack .NET Developer

- Delivered end-to-end web solutions for multiple clients, integrating business workflows with secure, scalable .NET applications and SQL Server backends.
- Architected multi-user platforms supporting financial, media, and marketing use cases, from transactional systems to high-traffic consumer websites.
- Integrated secure financial data systems for a student loan consolidation client, including direct compliance alignment with Fannie Mae systems.
- Designed and deployed a stock market trading competition platform and a photo contest site, managing real-time submissions, voting logic, and scalable database infrastructure.
- Led development of FastBrowserSearch.com, a globally distributed browser toolbar platform deployed to billions of installations worldwide through large-scale marketing campaigns.

Selected Projects

Angular + ASP.NET Core Platform (2026)

Webapp: <https://angular.johnbelthoff.com/>

Full-stack Angular 21 SPA with ASP.NET Core API backend, serving as both an online portfolio and a production reference implementation. Backend integrates HashiCorp Vault across four secrets engines — dynamic PostgreSQL credential rotation (922-line credential manager with lease renewal, zero-downtime drain, and self-healing on auth errors), PKI-based TLS trust injection with custom X.509 chain validation, KV v2 secret management, and AppRole authentication. Implements JWT authentication with HMAC-SHA256 refresh token hashing, HttpOnly cookie refresh flow, ASP.NET Core Identity with email confirmation, Cloudflare Turnstile bot verification, and rate limiting. Health check endpoints expose Vault lease status, credential health, and database identity verification. Deployed behind Cloudflare and NGINX reverse proxy with Docker containerization.

Semantic Knowledge System — RAG + MCP

Blog: <https://codematters.johnbelthoff.com/semantic-knowledge-system-ai-pgvector/>

Source available upon request

RAG (Retrieval-Augmented Generation) system providing persistent, searchable memory for AI coding assistants across multiple repositories. C# embedding pipeline ingests source files and git commit history, generates OpenAI text-embedding-3-small vectors, and stores in PostgreSQL with pgvector extension and HNSW-indexed cosine similarity search. Hash-based change detection (SHA-256) minimizes API costs (~\$0.01 for 3 repos, 570 embeddings). Security-conscious ingestion with secret file detection, path-escape prevention, and database CHECK constraints blocking absolute paths. Custom Model Context Protocol (MCP) server built with the official C# SDK exposes four semantic search tools (search_context, get_interaction_history, get_file_summary, list_recent_interactions) that Claude Code discovers and calls natively. Enables cross-repository context retrieval — an AI agent working in one repo can pull decisions and code reviews from completely different repos.

CodeMatters Frontend — SSR Security + Pen Testing

Webapp: <https://codematters.johnbelthoff.com/>

Source available upon request

Headless WordPress / Angular 21 SSR architecture with production security hardening and penetration testing. Conducted OWASP ZAP (frontend + GraphQL active scans up to 120 minutes) and WPScan (aggressive plugin/theme/user enumeration, ~150K requests) with findings triaged through an 8-iteration AI-assisted code review pipeline (72 → 91 production readiness score). SSR server (979 lines) implements CSP nonce architecture with placeholder-based cache strategy, SSRF prevention via startup

URL allowlist validation, timing-safe token authentication, AsyncLocalStorage for concurrency-safe renders, SSR concurrency limiting with 503 shed, and stampede protection via in-flight render coalescing. Server-side DOMPurify pool with custom security hooks. NGINX reverse proxy with per-location CSP policies, TLS 1.2/1.3, rate limiting, and request tracing. 632-line Vitest integration test suite.

Poker Hand Evaluator

Webapp: <https://poker-calculator.johnbelthoff.com/>

GitHub: <https://github.com/JohnBelthoff/poker.net>

High-performance Texas Hold'em evaluator achieving near-native C++ performance parity in pure managed C# (~180–210M evaluations/sec, ~943 ns/op core). Complete reimplement of Cactus Kev/Suffecool algorithm using bitmask-encoded cards, precomputed lookup tables, and prime-product hashing for O(1) hand classification. Zero-allocation hot paths via stackalloc/Span, AggressiveInlining/AggressiveOptimization, ReadOnlySpan for permutation tables, and Parallel.For with thread-local accumulators. Dedicated cross-language benchmark suite (BenchmarkDotNet with MemoryDiagnoser) proves C# throughput within sub-microsecond margins of C++ MSVC /O2 AVX2. Deployed with Docker, SQL Server backend, and Razor Pages UI.

Ballistics.Systems

Webapp: <https://ballistics.systems>

Enterprise-grade .NET 10 ballistics calculator with enforced clean architecture across an 8-project solution (~13,500 lines of C#). Automated architecture tests (NetArchTest) validate dependency flow, domain immutability, and infrastructure isolation — violations fail the build. Domain-driven design with deterministic, side-effect-free domain logic (Miller gyroscopic stability model, Satterlee ladder test optimization). Multi-tier caching (in-memory L1 → Redis L2 with automatic warming), runtime-configurable cache TTLs without redeployment, BenchmarkDotNet performance profiling with MemoryDiagnoser. Production infrastructure includes SynCFusion PDF report generation, Stripe payments, MailKit email with HTML/text dual templates, ASP.NET Core Identity with 2FA, HMAC-signed API tokens, Turnstile bot protection, and multi-stage Docker builds on Alpine. Data layer uses Dapper with SQL Server (1,654-line schema, stored procedures, multiple schemas).

I Ching Oracle

Webapp: <https://iching.rocks/>

Full-stack .NET Core web application integrating OpenAI's Chat Completions API as a Stripe-gated premium feature. Custom multi-section prompt architecture generates 8–14 concurrent AI requests per reading (opening, judgment, image, individual lines, Ten Wings, synthesis, conclusion) with structured

domain data injection (hexagram, trigram, changing line context). Parallel async orchestration via Task.WhenAll, AES-encrypted request/response storage with per-encryption IV generation, per-request token usage tracking for cost monitoring, and SHA-256 key derivation. Full membership system with user authentication, Stripe payment integration, PostgreSQL via Dapper, and production deployment with caching and SEO automation.

Additional Experience

- **Broadcast & Audio Engineering:** Emmy Award–winning audio engineer with decades of experience designing audio systems for NYC television studios, supporting national television productions, live broadcasts, and post-production sound design. *(Full broadcast résumé available upon request.)*
-

Education

- **Berklee College of Music** – Professional Music, Production & Engineering (1981)
 - **The Recording Workshop** – Advanced Audio Engineering, *2nd in Class* (1984)
-

Awards

- **Emmy Award** – Outstanding Audio, 1997
- **Cable ACE Award** – Best Talk Show Audio, 1997