Elijah Johnson

Bethesda, MD
johnsonelijah2626@gmail.com
linkedin.com/in/edjohnson26
github.com/elijahjohnson | lumicreative.org

Technical Product Lead | Distributed Systems & Platform Architecture

Product-focused systems engineer designing scalable platforms for complex financial and collaboration workflows. Designs event-driven systems, modular domain architectures, and distributed platforms while translating business requirements into high-leverage technical systems.

Technical Skills

Languages
C#, TypeScript, JavaScript, Node.js, SQL, PHP

Frameworks
.NET / ASP.NET Core, Angular, React, NestJS, Laravel, Prisma

Architecture
Distributed Systems • Event-Driven Architecture • CQRS • Modular Monoliths • Microservices • Multi-Tenant Platforms

Infrastructure
PostgreSQL • Cosmos DB • Redis • Docker • GraphQL • Hangfire • API Gateways

Observability
OpenTelemetry • Distributed Tracing • Performance Monitoring

Experience

Lead Senior Software Engineer (Technical Product Lead)

Pioneering Evolution — Arlington, VA
2024–Present

Lead engineer driving architecture and product design across legacy modernization and the SyncPoint distributed platform, a multi-tenant financial workflow system supporting federal program operations.

Platform Architecture & Modernization (Lead)

- Led architecture redesign of a contract management platform, enabling full digital lifecycle tracking and unlocking ~$10M in previously unallocated funding
- Reduced system latency from ~15s to <700ms through caching strategy, query optimization, and data access redesign
- Defined backend architecture including API design, relational data modeling, and workflow orchestration for contract lifecycle systems
- Conducted 30+ stakeholder interviews to translate operational workflows into scalable system architecture and product flows

SyncPoint Distributed Platform (Backend Architecture & Systems Ownership)

- Designed and implemented CQRS-based backend services using MediatR with validation, authorization, idempotency, and audit pipelines
- Implemented event-driven synchronization patterns ensuring consistency across distributed services
- Architected asynchronous processing pipelines using the transactional outbox pattern and background workers (Hangfire), enabling reliable cross-service workflows
- Contributed to polyglot persistence architecture using Cosmos DB (domain data), PostgreSQL (platform data), and Redis caching
- Improved system reliability and observability through OpenTelemetry-based tracing and service performance monitoring

---

Software Engineer III

Pioneering Evolution — Arlington, VA
2023–2024

- Led development of financial workflow APIs and domain models supporting budgeting, approvals, and reporting systems
- Reduced application load times by ~35% through frontend optimization and improved backend query patterns

- Refactored legacy architecture into modular components, improving maintainability and development velocity

---

Software Engineer

Pioneering Evolution — Arlington, VA
2019–2023

- Built APIs and UI components for financial planning and budgeting systems used by government program offices
- Implemented workflow automation, approval pipelines, and operational dashboards
- Collaborated closely with stakeholders to translate business processes into scalable product features

---

Founder-Led Platform Architecture Initiative

Lumi — Networked Creative Studio Platform

Architecting a mobile-first collaboration platform enabling creators to discover collaborators, form project teams, and execute structured creative work.

- Defined modular domain architecture across core platform modules (Auth, Identity, Match, Network, Community, Studio, Notifications, Platform)
- Designed event-driven system using transactional outbox pattern and background workers, enabling asynchronous coordination across platform workflows
- Architected collaboration systems including creator discovery, messaging, project workspaces, and activity-driven feed
- Designed scalable data model supporting 100k+ creators with PostgreSQL and Redis-backed caching strategies
- Defined domain event catalog, worker runtime architecture, and module boundary enforcement to ensure long-term system scalability
- Designed growth-oriented system mechanics integrating discovery, feed ranking, and collaboration demand signals

---

System Design & Architecture Highlights

Event-Driven Workflow Architecture

- Designed reliable event pipelines using transactional outbox pattern and asynchronous workers
- Implemented idempotent processing and retry-safe background job systems
- Enabled cross-service consistency through event-driven synchronization

Distributed Data & Performance Optimization

- Diagnosed and resolved critical performance bottlenecks in production systems
- Reduced response times from 15s → <700ms through caching and query optimization strategies
- Designed scalable data access patterns across relational and NoSQL systems

Modular Domain Architecture

- Designed domain-driven module boundaries enforcing ownership, isolation, and clean integration via events
- Established architectural patterns supporting future service extraction without system rewrites

---

Engineering Writing & Thought Work

- Event-Driven Architecture for Financial Systems
- Designing Escrow-Based Payment Systems for Creator Platforms
- Performance Optimization in Distributed Applications
