

Joshua Albert

Senior Software Engineer | AI Systems

Python · LangGraph · FastAPI · Neo4j · TypeScript · React

Senior software engineer with about 20 years of professional and independent development experience across data-intensive products, graph interfaces, enterprise integration, developer tools, and tested full-stack systems. Current work includes typed, replayable agent-evaluation workflows and, separately, provenance-aware evidence and retrieval tooling, using agents and graphs only where warranted and keeping system boundaries explicit enough to inspect, evaluate, and govern.

joshua.ablert@socratesone.com
github.com/socratesone

linkedin.com/in/joshua-j-albert
socratesone.com

— SELECTED ENGINEERING EVIDENCE

Concrete systems. Evidence boundaries stated.

VISIBLE NETWORK LABS · SOURCE-BACKED PRODUCTION CLAIM · NOT PUBLICLY CORROBORATED ~10 minutes to ~10 seconds Data pipeline performance	GRAPH INTERFACES VISIBLE NETWORK LABS · SOURCE-BACKED PRODUCTION CLAIM · NOT PUBLICLY CORROBORATED Up to ~5K nodes / ~20K edges Interactive graph systems	EVALUATION & RELIABILITY INDEPENDENT CAPABILITY DEMONSTRATION · REPOSITORY-INSPECTED · PUBLIC RUN ARTIFACT PENDING Replayable agent evaluation Agent workflow evaluation
---	---	--

— PROFESSIONAL EXPERIENCE

Production product work and current independent engineering.

SocratesOne Development LLC Independent Software Engineer - AI Systems Denver, Colorado INDEPENDENT ENGINEERING · REPOSITORY-INSPECTED CURRENT WORK Current work spans separate systems: EvidEngine and VimCommander, a typed LangGraph evaluation workflow, and FastAPI-backed internal applications. - Implemented typed LangGraph workflows that separate state, model calls, scoring, policy checks, selection, and replay evaluation, with per-node traces and run metrics. - Built FastAPI-backed internal applications with validated contracts, persistent data models, audit trails, and automated test coverage. - Developed provenance-aware evidence and retrieval tooling with deterministic artifacts, stable identifiers, exact/BM25 lookup, graph traversal, optional Neo4j integration, and MCP tools.	Jul 2025 - Present
Visible Network Labs Software Engineer, Full Stack Denver, Colorado PRIOR EMPLOYMENT · SOURCE-BACKED PRODUCTION CLAIMS · PUBLIC ARTIFACTS PENDING Full-stack product engineering for data-heavy network-analysis applications. - Reduced a production CSV import from roughly 10 minutes to roughly 10 seconds by removing N+1 query patterns, batching database operations, memoizing repeated lookups, and hardening edge-case parsing. - Built Vue, TypeScript, D3.js, and Apache ECharts interfaces for relational datasets, including dynamic filtering, styling, grouping, and performance work for large network graphs. - Implemented SAML 2.0 single sign-on and granular, role-scoped authorization for complex data-access roles and permissions. - Built and maintained unit and end-to-end test infrastructure with GitHub Actions pre-merge checks for data-heavy application workflows.	Jan 2021 - Jul 2025

Current work in evidence systems and developer tools.

● AGENT SYSTEMS INDEPENDENT PROJECT · REPOSITORY-INSPECTED · EARLY-STAGE V0.1.0 EvidEngine Local-first evidence-graph tooling that turns documents, configuration, source anchors, and Git history into source-traceable artifacts, giving agent retrieval an auditing, visibility, and governance layer so teams can see and trust what agents pull in. - Implemented deterministic ingestion, extraction, linking, indexing, and query stages with stable identifiers and explicit run artifacts. - Modeled documents, segments, passages, entities, relationships, files, commits, and authors with provenance and honest source locators. - Added BM25 keyword search, exact-key lookup, graph traversal, optional Neo4j export and synchronization, and Model Context Protocol (MCP) tools. Status: early-stage, source-install, CLI-first software with no hosted service or local UI; semantic retrieval is experimental and off by default.	● DEVELOPER TOOLS INDEPENDENT PROJECT · REPOSITORY-INSPECTED · ACTIVE DEVELOPMENT VimCommander A browser space shooter built with React, TypeScript, and a full automated test suite, designed to teach Vim commands through keyboard-driven gameplay. - Built the game with TypeScript, React, Phaser, Three.js, Zustand, Zod, Vite, and Vitest. - Separated keyboard semantics from game behavior through an engine-independent mapping layer that routes high-level Vim actions. - Modeled editor modes and transitions with a pure, deterministic finite-state machine. Status: active development; the tutorial covers a deliberate subset of Vim commands.
---	--

Full-stack engineering for AI products and platforms.

AI systems Python, LangGraph, LangChain Core, OpenAI API, tool calling, structured outputs, RAG, graph-backed retrieval, agent evaluation, run metrics, policy controls, workflow orchestration, context engineering, and MCP integration.	Backend and data FastAPI, PHP, REST APIs, Neo4j, PostgreSQL, MySQL, Redis, SQL, knowledge-graph and relational modeling, caching, and data-pipeline optimization.
Frontend and visualization React, TypeScript, Vue, JavaScript, D3.js, Apache ECharts, Phaser, Three.js, responsive interfaces, network visualization, and GIS interfaces.	Quality and delivery System design, performance analysis, unit, integration, and end-to-end testing, replay and regression evaluation, observability, GitHub Actions, CI/CD, Docker, and Google Cloud Platform.

Foundation in web platforms and integration.

CommonPlaces Nov 2014 - Jan 2016 Drupal Developer Manchester, New Hampshire - Developed core components of an internal document-management intranet, including LDAP authentication, multi-tier permissions, content workflows, and Drupal application functionality. - Contributed to Drupal migration and commerce work involving custom content-management requirements and legacy integration.	Independent and Freelance Development 2005 - 2020, intermittent Full-Stack Developer - Built and maintained custom web applications and content-management implementations for small-business and consumer clients using PHP, MySQL, Drupal, WordPress, and Codelgniter. - Modernized legacy applications, implemented custom CRUD workflows, integrated third-party services, and introduced source control and deployment automation.
---	--

Discuss an AI systems or full-stack engineering role.

EMAIL joshua.ablert@socratesone.com
GITHUB github.com/socratesone

LINKEDIN linkedin.com/in/joshua-j-albert
WEBSITE socratesone.com