

Joshua Albert

Senior Software Engineer | Agentic AI Systems

Python · LangGraph · FastAPI · Neo4j · TypeScript · React

Senior software engineer with about 20 years of professional and independent development experience across data-intensive applications, graph visualization, enterprise authentication, developer tooling, testing, and Artificial Intelligence systems.

joshua.albert@socratesone.com
github.com/socratesone

linkedin.com/in/joshua-albert-9255a34
socratesone.com

SELECTED ENGINEERING EVIDENCE

Concrete systems. Traceable claims.

~10 MINUTES TO ~10 SECONDS

Data pipeline performance

Reduced processing time for CSV imports containing more than 5,000 rows by removing N+1 query patterns, batching database operations, memoizing repeated lookups, and hardening edge-case parsing.

UP TO ~5K NODES / ~20K EDGES

Interactive graph systems

Built Vue, TypeScript, D3.js, and Apache ECharts interfaces for relational datasets, with dynamic filtering, styling, grouping, and performance work for large network graphs.

9-NODE LANGGRAPH WORKFLOW

Agent workflow evaluation

Built a typed LangGraph pipeline with structured Pydantic validation, weighted model-assisted scoring, policy filtering, versioned prompts, and replay fixtures that produce per-node audit traces for regression review.

FLAGSHIP PROJECTS

Current work in evidence systems and developer tools.

ACTIVE DEVELOPMENT · V0.1.0

EvidEngine

Local-first evidence-graph tooling that turns documents, configuration, source anchors, and Git history into source-traceable artifacts for agent retrieval and inspection.

- 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.

Current limits: early-stage, source-install software with a command-line interface; no hosted service or local user interface. Semantic retrieval is experimental and off by default.

ACTIVE DEVELOPMENT · BROWSER GAME

VimCommander

A browser space shooter 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.

Current limits: the game remains in active development; its tutorial covers a deliberate subset of Vim commands, and no adoption or learning-effectiveness claims are made.

CORE TECHNICAL CAPABILITIES

Full-stack engineering for AI-native products.

Agentic Artificial Intelligence systems

Python, LangGraph, LangChain Core, OpenAI API, tool calling, structured outputs, retrieval-augmented generation, graph-backed retrieval, agent evaluation, prompt and context engineering, and Model Context Protocol integration.

Backend and data

FastAPI, PHP, REST APIs, Neo4j, PostgreSQL, MySQL, Redis, Structured Query Language, knowledge-graph and relational modeling, caching, and data-pipeline optimization.

Frontend and visualization

TypeScript, Vue, React, JavaScript, D3.js, Apache ECharts, Phaser, Three.js, responsive interfaces, network visualization, and Geographic Information System interfaces.

Quality and delivery

System design, performance analysis, unit, integration, and end-to-end testing, agent evaluation, GitHub Actions, continuous integration and delivery, Docker, and Google Cloud Platform.

PROFESSIONAL EXPERIENCE

Recent engineering work.

SocratesOne Development LLC Independent Software Engineer - Agentic AI Systems Denver, Colorado Current flagship work includes EvidEngine and VimCommander, described above, supported by focused prototypes and evaluation workflows for agentic systems. - Implemented structured LangGraph workflows that separate typed state, model calls, scoring, policy checks, selection, and replay evaluation. - Built FastAPI-backed internal applications with validated request and response contracts, persistent data models, audit trails, and automated test coverage.	Jul 2025 - Present
Visible Network Labs Software Engineer, Full Stack Denver, Colorado Full-stack product engineering for data-heavy network-analysis applications. Performance and graph-interface results are documented in Selected Engineering Evidence above. - Implemented Security Assertion Markup Language (SAML) 2.0 Single Sign-On and granular authorization for complex 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

EARLIER EXPERIENCE

Foundation in web platforms and integration.

CommonPlaces Drupal Developer Manchester, New Hampshire - Developed core components of an internal document-management intranet, including Lightweight Directory Access Protocol 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.	Nov 2014 - Jan 2016
Independent and Freelance Development 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 CodeIgniter. - Modernized legacy applications, implemented custom create-read-update-delete workflows, integrated third-party services, and introduced source control and deployment automation.	2005 - 2020, intermittent

CONTACT AND RÉSUMÉ DOWNLOAD

Discuss a senior AI systems engineering role.

EMAIL	joshua.albert@socratesone.com	LINKEDIN	linkedin.com/in/joshua-albert-9255a34
GITHUB	github.com/socratesone	WEBSITE	socratesone.com