

Justin Sublette

AI Systems Engineer | Agent Infrastructure and Persistent Execution

j.sublette@vmosaic.com · 812-374-4565

Naples, Florida, USA · Willing to relocate

Open source: [Mathematical Research](#) · [Protocol Runner](#)

Profile

AI systems builder engineering durable state, execution and recovery for autonomous research and coding agents. Builds systems that preserve useful work across agent replacement, interrupted operations and changing context. Connects storage and process boundaries to research continuity, dependable development workflows and deployed applications.

Experience

Founder and AI Systems Engineer | Vector Mosaic Inc. | 2025-present

Founded a software company spanning autonomous research, agent infrastructure and practical applications. Own problem definition, architecture, integration, evaluation and release acceptance, collaborating extensively with Codex on implementation, investigation and testing. Began practical software development in June 2025 through self-directed, AI-assisted learning. Selected systems below are projects within this work.

Mathematical Research / RH Mission | Durable infrastructure for autonomous research | 2026-present

- Built a research system pursuing a checkable proof or disproof of the Riemann Hypothesis with an AI executive. Designed the Mission to survive replacement of that executive: the agent chooses scientific strategy, while the host controls process lifecycle and containment.
- Separated durable objective, strategy, cumulative scientific context, branches, evidence and proposed results. Implemented one SQLite writer, immutable revision identities, content-addressed source material and exact checkpoints so successors reconstruct the research position from owned records rather than an immortal conversation.
- Integrated authenticated indexed retrieval and atomic publication of one evidence result with its authored scientific continuation. Exact-request replay recovers committed effects without rerunning completed research; preserving a proposed result stays separate from review and mathematical admission.
- Released the complete standalone core, Host and adapters under MIT. [Real Linux qualification](#) established initialization, tool interaction and orderly stop/restart with retained state; a separate live successor retrieved exact records and tightened a bound under its retained assumptions.

RH remains open. Public results cover the standalone release with fresh state and the selected examples.

Protocol Runner | Stateful orchestration for long agent workflows | 2026-present

- Designed persistent plans, execution state, attempts and evidence outside conversations. Kept dependent steps in an ongoing Codex conversation and launched separate workers for independent assignments; the coordinating agent retains quality judgment and final acceptance under my direction.
- Used Runner hundreds of times in repository development and investigation; a retained census identified more than 4,400 worker tasks matching its prompt signature. Repaired a wrong-checkout execution failure by binding workers to the launching Git worktree, then verified the assignment in a fresh run.
- Carried an engineering-history update through 25 investigation assignments over 595 commits, separate review, correction of three of 40 accounts and fresh full-draft review before internal acceptance. Preserved completed investigations instead of restarting the whole workflow.
- Open-sourced [Protocol Runner](#) under MIT, with a real API/store recovery demo using simulated workers and separate live Codex CLI completion and cancellation checks. [Evidence](#).

Workstation Control | Machine and application access for AI agents | 2026-present

- Built authenticated worker services and explicit host profiles for configuration delivery, credential installation and remote file inspection. Used atomic replacement, identical-content replay, explicit overwrite for conflicts and registered credential destinations; returned fingerprints rather than secret values.
- Bound browser workflows to a browser, debugging port and profile. Required exactly one matching managed process before endpoint reuse and rejected competing ownership; added tests for mismatched processes and sessions.
- Separated boot-time services from interactive desktop automation and coordinated reboot preparation while keeping health inspection available. Isolated stateful smoke tests after early tests created synthetic conversations in normal working state.
- Developed Windows UI Automation prompt delivery after experiments showed persisted conversation state did not guarantee desktop visibility. Live Protocol Runner tests selected exact visible conversations and received worker start/completion reports.
- Extended related repository-wide tool-interface work with non-mutating discovery, structured results and consistent option placement. A closing audit covered 81 stable tools with zero interface findings or probe failures, checking command discovery and interface conformance.

Life Links | Deployed application boundaries for general-purpose AI clients | 2026-present

- Created a private-context application for physical places, objects, routines and calendars. Shared application commands, persistence and ownership rules across a React UI, page-bound WebMCP and OAuth-protected remote MCP. Bound destructive confirmation to the exact target, revision and previewed effects; used stable command identities and guarded recovery after uncertain provider responses.
- Qualified deployed Codex and ChatGPT create/read/edit workflows across conversations with the site closed, plus deletion confirmation, replay and revocation checks. Bounded live Google and Outlook calendar tests exposed metadata, response-interpretation and time-zone defects that were repaired before affected checks passed.
- Reduced startup API requests from 55 to 7 on the same 48-member Collection. Median first-content rendering improved 43-57% across four matched desktop/mobile-width conditions, three observations each. Released sanitized MIT-licensed source and deployed exact checked revisions on Railway.

Technical foundation

TypeScript, Node.js, Python, PowerShell, React; SQLite, PostgreSQL, atomic persistence, immutable revisions and content-addressed storage; MCP, HTTP/JSON-RPC, REST/OpenAPI, OAuth/PKCE; Git worktrees, Windows services/scheduled tasks, UI Automation, browser debugging protocols, Linux/systemd, SSH, authenticated worker APIs, containers, Hetzner and Railway. Experience centers on existing-model integration and agent systems; model construction and training remain a development area.

Earlier experience and education

Manufacturing and pipeline construction, 2012-2018: shift-team leadership and field coordination. Independent market research and trading, 2018-2025.

CCNA: certified April 7, 2025; valid through April 7, 2028. **Indiana University, SPEA:** policy analysis studies, 2013-Spring 2015; no degree completed. **Columbus North High School:** Academic Honors diploma.