Stephen "Sven" Benner

Software Engineer | AI Agent Platforms | Systems Integration

Loma Linda, CA | sven@sensiblesolutions.me | 707.968.7082

GitHub | LinkedIn | rEach source

SUMMARY

I design systems that let AI coding agents do real work safely, and I verify what they produce. At Pateras AI I have designed and directed rStack, an agentic software ecosystem. It consists of 48 software components, and it gates every agent action against a written policy, budgets LLM spend at every level of delegation, and refuses code whose tests have not been shown able to fail. Before that, I was the systems administrator for a university's estate of about 190 virtual machines, where I shipped production integrations across its student, housing, admissions and procurement systems. I have written Ruby since 2012 and built LLM tooling across the OpenAI, Anthropic, Gemini and xAI APIs since 2024.

SKILLS

- **AI engineering:** LLM API integration (OpenAI, Anthropic, Gemini, xAI); agent-workflow engineering with Claude Code and Codex (spec, test, verify and ship pipelines, lifecycle hooks, mutation-tested evidence); MCP servers; retrieval (HNSW, BM25, pgvector); local LLM serving (llama.cpp, vLLM); evaluation harnesses.
- **Platform and reliability:** process supervision and health checks, drift reconciliation, policy-as-code, budgets, rate limits and circuit breakers, secrets management, backup and restore drills, structured event logging.
- **Integration and operations:** REST, S3 and SFTP integrations; SAML SSO (WSO2); Banner, StarRez, Slate, Jaggaer, TouchNet; Microsoft 365 and PowerShell; Google Workspace; Linux, VMware and Ansible; TLS/PKI and DNS.
- **Languages (independent):** Ruby (expert), shell (Bash), JavaScript, PHP, HTML and CSS; SQL for operational use.
- **AI-assisted delivery (designed, reviewed and verified by me; code written by agents to my specs):** Rust, TypeScript with React and Svelte, Swift/SwiftUI.

EXPERIENCE

Co-founder & CTO

Pateras AI, LLC | 2026–present | remote

Designed and directed rStack, an agent-built platform for running AI coding agents safely; the code was written by AI coding agents to my specifications and verified by me.

- **Scale (measured 2026-10-20):** 63 repositories; about 1.25 million lines of code and tests (Ruby about 418,000, Rust about 365,000, TypeScript, JavaScript and Svelte about 262,000), plus about 820,000 lines of specifications and documentation.
- **Running components:** 48 components registered with its supervisor, 40 healthy: 15 platform and operations, 13 applications, 7 governance and security, and 5 AI and retrieval. Alongside them: 13 agent-harness plugins and the SDK they are built on.

Patterns the platform is built on

- **Intent, reality and integrity are owned separately.** A formation engine (rconfig) holds the desired state, the orchestrator (rstacks) owns what is actually running, on which port and whether it is healthy, and a reconciler (rhomeostasis) owns architectural integrity. Lower layers never depend on higher ones.
- **GitOps for architecture.** Component manifests and integration specs are the desired state; 42 drift checks compare them with the tree and the live registry and report every gap.
- **Decide before acting.** Every agent action is checked against a written rulebook before it runs (polispec) and answered allow, warn or deny, with the rule, the reason and the next step. A separate policy decision point (rsentinel) answers what an identity (ridentity) may do.
- **Claims owned by processes.** Every lock or lease is paired with the process that took it and released when that process dies, never while it lives (reaper).
- **Budgets that hold at every level of delegation.** A hierarchical reservation ledger (ration) meters LLM calls, never retries a charge on the caller's behalf, and gives each upstream timeouts, a concurrency cap, a token bucket, a circuit breaker and a kill switch.

- **Evidence over assertion.** An event, decision and provenance ledger (rlogs) is the durable record, and a contract-attestation ledger (rnotary) gates the orchestrator's readiness. Neither can disable the other; only the operator can override, with a two-step confirmation.

Problems it solves

- **AI-written tests that test nothing.** A "guard drill" removes an implementation and requires the new tests to fail. It caught AI-written tests that passed without testing anything, and a change gate refuses code without that evidence.
- **Concurrent agents overwriting each other.** Process-owned file and repository locks let several agent sessions work in one tree; a dead session's locks are released automatically, and a live one's never are.
- **Agents reaching production.** Branches run dev, then test, then stable. Agents may promote only to test, and only on a green gate; production promotion, rollback and secrets belong to the operator (polispec, public at github.com/SteveBenner/polispec).
- **Runaway LLM spend and failing providers.** Budgets reserved before each call, plus per-provider circuit breakers, stop a loop of agents from spending past its limit or hammering a failing upstream.
- **Documentation that drifts from what is deployed.** The drift checks named three components missing from the inventory page before a person noticed, and one within the same minute it was added. A coverage check now names anything the survey could not see, so a verdict can't be broader than its survey.

Selected components

- **rLLM** (Rust): a multi-tenant LLM inference gateway serving chat, embedding, classification, reranking and tokenization over named model slots, with throttling and budgets.
- **rMaestro** (Rust): a chat based harness that turns a request into durable, validated agent work, with a hook system of 18 event types and circuit breakers. Designed around native small local model support.
- **rHarnessFactory** (Rust): a harness composition framework that declares harness-agnostic feature modules in a pseudocode syntax, ships with core implementations in ruby. A factory for custom harnessses.
- **rAgnarok** (Rust): a retrieval service with an embedded HNSW vector store and hybrid BM25 and vector search.
- **rOuter** (Rust): the MCP gateway through which agent harnesses reach every rStack component.
- **rVault** (Ruby): secrets management with AES-256-GCM envelope encryption and time-bounded read leases.
- **rBackup** (Ruby): backups as declarative playbooks, with scheduled scrubs, restore drills and ransomware canaries.
- **rBench** (Ruby): benchmarking and evaluation of models and agents, focused on small local models.
- **rBrain** (rcorpus): an installable microdatabase and agent-memory library, with key-value, graph and retrieval planes over self-contained corpora. Built atop 78 peer-reviewed academic publications from 2026.
- **rNA** (Ruby, React with TypeScript): a command-line tool and web app that guides a person section by section through structured YAML specifications (app, product, business and more), validates them for completeness, autosaves drafts and resumes unfinished specs.
- **rSlides** (Ruby, JavaScript): turns an uploaded PowerPoint deck into a faithful, editable HTML deck, applies changes through an AI agent driven by plain sentences, with one conversation per deck and git-backed history, and exports portable HTML or PDF. Canva killer.
- **rIdentity** (Rust, Svelte): the platform's identity layer. Every human, agent and service gets one stable identity across the fleet with many sign-in methods, never keyed on email address; authorization deliberately left to policy service.
- **rAssessments** (Ruby, React, PostgreSQL with pgvector): a WSCUC academic accreditation-evidence platform prototype with cited retrieval, chain of evidence, AI-native composition and validation, and corpus management.
- **Other Pateras work:** built a paid client's iOS HealthKit app (SwiftUI), verified on real devices; hired and managed two contract developers on fixed-price agreements with NDAs (March–June 2026).

System Administrator

La Sierra University | July 2025 – August 2026 | Riverside, CA

- Built production integrations from the Banner student information system: a scheduled demographic and booking sync to StarRez housing (S3); ID-photo batches from InstantID to Slate admissions (SFTP, with bounded retries); the Jaggaer procurement connection; SAML service providers on WSO2 for vendor single sign-on; and the server and Banner link for TouchNet payments.
- Directed AI coding agents to build Sanctum, an IT-operations web application (Ruby, CouchDB, React) with remote host agents, a Minitest and Capybara test suite, CI on main and a security review. It held an inventory of about 150

servers synced from vCenter, runbooks, patch windows, change requests, role-based access and audit. Adopted by the IT team and approved by management.

- Recovered mail across about 65,000 mailboxes (about 5,000 active) with a scripted restore completed in a single 14-hour shift; detected and remediated a compromised executive account after hours within about 30 minutes, escalating first and documenting the next morning.
- Found root causes behind symptoms: a housing sync that worked by hand but failed under cron was the scheduler not passing its environment to child processes; a payroll-process outage was a Java runtime change. Led the campus wildcard SSL certificate rotation with a Java 17 upgrade.
- Piloted LLM triage of 139 help-desk tickets and won approval for an AI API budget. Wrote the university's Data Classification Policy, including rules for AI use, aligned to FERPA, HIPAA, GLBA and PCI (submitted; approval pending at departure).
- Ran a VMware estate of about 190 virtual machines; authored about 50 automation scripts and 7 Ansible playbooks.

Adjunct Professor

Zapara School of Business, La Sierra University | November 2025 – June 2026 | Riverside, CA

- Designed and taught the MBA course AI for Strategic Decision Making, and co-taught an analytics capstone whose labs I designed around local LLMs.
- Designed and directed Teach and rEach, a paired instructor and student platform for AI-assisted coursework. Teach (Ruby, Roda, PostgreSQL) enrolls students, ships encrypted, signed guardrails and assignment packages, takes submissions with signed receipts, and runs a help queue that AI helper agents staff over MCP. rEach (public: github.com/SteveBenner/rEach) is a plugin for Claude Code and Codex that turns a student's own AI agent into a bounded course partner, keeps its work inside the assignment, and checks it before submission.

Data Engineer / Analyst (subcontract)

healthcare finance | September 2023

- Wrote a Ruby tool that consolidated a senior-care provider's vendor records across its locations, replacing manual consolidation that a timed sample projected at roughly 50,000 hours.

Earlier

- Career break, 2016–2023, with part-time freelance IT and web consulting.
- Ruby on Rails developer (remote contract), Hyper, Oslo, early 2015: about three months of paid work on an open-source Rails product, followed by a full-time offer with relocation (work visa not granted).
- Software Architect, X-Factor Applications, 2013–2014: led development of MyDentalCcompanion, a mobile-health SaaS prototype, and built its Rake pipeline from build to TestFlight.
- Freelance web developer and IT consultant, 2012–2016: business websites, WordPress themes and plugins in PHP, and IT support.

PROJECTS

- **Fact Drift Sentinel** (fact_drift, 2026, github.com/SteveBenner/fact_drift): a public Drupal 10.6/11 module that flags published facts that drift from a canonical source, with an optional Drupal AI module extractor, 353 PHPUnit tests and CI on two Drupal versions. Designed and directed; built by AI coding agents and verified by hand.
- **GPTK** (2024–2025): a multi-provider LLM toolkit in Ruby across the OpenAI, Anthropic, Gemini and xAI APIs, about 75% hand-written. Injecting the outline into every prompt fixed chapter hallucinations and brought prompt caching to over 80% of input tokens.

EDUCATION

- Pacific Union College — coursework in Computer Science, 2008–2013, including a study-abroad year in Germany (2011–2012).
- La Sierra University, Zapara School of Business — undergraduate coursework in Business Administration, 2025.

REFERENCES

Robert Kurtz, rpkurtz@gmail.com · Dulce Peña, dulce4848@msn.com · Marcelo Vetter, mvetter@llu.edu