Pašovice 71
687 56
Czech Republic
+420 733 230 772
stavel.jan@gmail.com
janstavel
jstavel
clojars.org/jstavel



Ing. Jan Stavěl

Stories bring the power to know; success fuels the power to grow.
Trust builds joy, and shared joy is twice the joy.

Python and Perl by hand, Clojure by heart, TypeScript with agents,
Terraform with respect.
I build systems whose output can be verified, not merely demonstrated.

I cherish the fleeting moments that remind me I am still growing.

Core Expertise

Automation	Lifecycle management of large-scale test ecosystems: exploratory testing balanced with suites (Pytest, Robot Framework, Playwright) built to survive years, not quarters.
AI Verification	Evaluation harnesses for LLMs and agents — golden datasets, LLM judges, per-item provenance, pre-registered accept/regress gates — so quality is measured, not asserted; guardrail layers that fail closed.
Reliability	Long-term maintenance of integration frameworks, container-native operations (Podman), CI/CD (Jenkins, Ansible, GitHub Actions), and the monitoring that proves a system still works: Prometheus, Grafana, Nagios.
Method	Specs, ADRs, runbooks, release notes and bug reports a stranger can act on. The written artefact, not the chat, is where work gets reviewed and corrected.
Linux	Linux From Scratch (kernel compilation, toolchain, bootloader); networking, storage (RAID, LVM), hardware RAID controllers; PKI and smart card deployment (RSA tokens, USB readers); production Linux administration (Debian, RHEL); scripting in Perl, Bash and Python.

Experience

2026 **Architect & Developer**, *JanBot — spec-first AI agent*, Remote
github.com/jstavel/janbot (live demo)

- **Grounded retrieval, fail-closed by construction:** answers come only from a versioned org-mode corpus; the ingestion scope is a whitelist, so material outside the published profile is provably never indexed.
- **Evaluation harness, not vibes:** 100-item golden dataset, LLM judge, per-item provenance and pre-registered floors. A compiled optimisation was **rejected** because faithfulness dropped on four items while citations gained one — the gate decided, not the intuition.
- **Guardrails tested, including the ones that failed:** a deterministic pre-LLM relevance gate refuses out-of-corpus questions at zero cost; a persona/over-claiming test still produced a first-person commitment from tangential evidence — the exact class of defect this work exists to find. Contract-first (FastAPI/Pydantic), specs and decisions in git.

2026 **Architect & Developer**, *Reactive Testing*, Remote
github.com/jstavel/reactive_testing

- **Post-execution validation:** collectors capture real system state during a scenario; validators run as pure functions over the collected corpus afterwards — deterministic replay, no race conditions. TypeScript/Playwright with Zod schemas, CDP automation, offline reconciliation.
- **Evolution of a Red Hat insight** (RxPy/DBus monitors) onto a web stack; delivered through a disciplined agent workflow with explicit specs and ADRs.

2026 **Architect & Developer**, *Krakatoa — Matching Engine (early stage)*, Remote
github.com/jstavel/krakatoa

- **Delivered — Rust matching core:** limit order book with price–time priority, taker–maker crossing, partial fills and multi-level sweeps; 26 `cargo test` cases across three spec-first specs.
- **Scaffolded, not yet built:** Clojure/Polylith layer — ZeroMQ transport, Kafka event log, chaos orchestrator: components, ADRs and specs exist; market orders, cancellation and the distributed consistency test layer (Jepsen-style) are the next milestone.

2026 **Spikes**, *Voice notes → searchable text*, Remote

github.com/jstavel/plaud-pipeline — a TypeScript/Node pipeline that turns recordings into an append-only Org-mode ledger, decomposed into swappable capabilities each with its own spec, contracts and tests (ADR 0012): one capability at v1 with CI, the rest specified. Accompanied by Playwright/ClojureScript retrieval and a Python + Gemini transcription pipeline — exploratory by design, local only.

2015–2026-05 **Senior Quality Engineer**, *Red Hat*, Remote

Long-term reliability and scalability of enterprise automation frameworks.

- **Test architecture:** layered suites where Ansible orchestrates and Robot Framework covers the UI/DBus boundary — and every scenario also ships as a standalone artefact a developer can run without the framework.
- **Python integration testing:** maintained and extended the primary Python ecosystem behind enterprise subscription management; scaled Selenium suites over complex web workflows and UI components.
- **Clojure ownership:** hired to own the Clojure part of the subscription-manager test ecosystem — wrote Clojure libraries bridging into a Java/TestNG suite, and maintained and extended the Clojure tests as the feature set grew (public repos: test-clj.testng, verify-testng).
- **Infrastructure as code:** Ansible and Jenkins CasC for reproducible CI/CD; container-native operations with Podman and Quay.io.
- **Mentorship:** a dedicated mentor and listener — identifying individual strengths and fostering a personal path to growth rather than a generic career ladder.

2013–2015 **Systems Architect & Programmer**, *National Library of CR*, Remote

Led backend services and integration strategy for long-term digital preservation.

- **Messaging architecture:** distributed data synchronisation in Clojure and RabbitMQ; functional patterns bridging library datasets with web services (Python/Plone).
- **Quality standards:** established TDD and integration testing for the new infrastructure.

2013 **Mentor**, *Google Summer of Code*, Remote

Mentored international developers in Java/Spring, focusing on practical integration testing strategies.

2007–2013 **Automation & IT Systems**, *Baťa a.s.*, Zlín

One of the primary developers modernising and scaling retail automation: a Perl-based intranet extended on the requirements of its internal users, plus automated reporting in LaTeX.

2003–2007 **Programmer & Linux Administrator**, *CoreNet / OSVČ*, Prague

- **Enterprise Python in XML:** XMLDSig, XSLT and automated document generation at CoreNet.
- **Independent Linux administration:** Debian servers and services — mail (qmail-ldap; later Postfix + `amavisd-new` with SpamAssassin and ClamAV), **Samba PDC (domain controller — Common Name, group-based roles and access permissions)**, Nagios/SNMPv3, PKI/TLS tunnels; **TCP/IP, DNS, HTTP troubleshooting** across the full stack; intrusion detection on the same hosts — LIDS kernel-level MAC (the one I relied on), AIDE/Tripwire integrity checks, Snort network; plus Django and Perl web development.

1997–2002 **Secondary School Teacher**, *OA Emila Holuba*, Brno

Taught computer technology and physics while serving as the school's primary site administrator. Linux had already begun before this period — compiling kernels and installing Slackware, configuring fvwm2, during postgraduate study.

Technical Skills

Testing — Integration & E2E testing (Pytest, Playwright, Selenium), Robot Framework, BDD/Gherkin, formal models & invariants, property-based testing, exploratory testing, TDD; test plans, checklists and severity/triage discipline

AI Engineering — Agentic development workflows (OpenCode, Goose), DSPy, RAG (ChromaDB, embeddings), LLM & agent evaluation (golden datasets, judges, pre-registered gates), guardrails, prompt auditing, spec-driven development

Observability — Prometheus, Grafana, Nagios/SNMP, MLflow experiment tracking; reading a dashboard and noticing when a number is wrong

Languages — Python, Clojure, Perl, Bash, JavaScript, Java, C++, Groovy; TypeScript written with agents and verified (not claimed as hand-written)

Infrastructure — Linux, Ansible, Jenkins, GitHub Actions, Podman/Quay, Kafka, RabbitMQ, Git; GCP via Terraform (Cloud Run, Secret Manager, Artifact Registry) — JanBot deployed on it, repo-verifiable; **production change discipline: plan before apply, rollback planning, post-change verification;** RHCSA (2016, certId verifiable online)

Professional Development & Mindset

Method — GTD, Deep Work, Atomic Habits (Clear), Growth Mindset (Dweck), Extreme Ownership (Willink) — ownership beyond my own team: stability, and the room for someone else to grow.

Education

1990–1997 **Technical University**, *VUT Brno*, *Ing.*

Technical Cybernetics, then postgraduate study in advanced control theory and C++.

Languages

Czech	Native	
English	Intermediate	*Professional working proficiency*

Interests

Beyond code	Woodworking, farming, cheese making; trading-domain literacy (Wyckoff, P&F, BTC on Kraken).