

John Scarrott

Platform Lead · Rust & Systems Engineer

john@scarrotts.uk | +44 7733 298950 | www.jscarrott.com | github.com/jscarrott | Barnstaple, Devon, UK

Profile

Platform lead and Rust engineer with over a decade of experience, from DO-178 safety-critical avionics to high-performance network security and cloud-native platforms. At RazorSecure I set technical direction across our on-train, frontend and backend platforms and led the migration of our on-train intrusion-detection agent, now on 1,000+ trains, from Python to Rust.

Experience

Senior Rust Software Engineer & Platform Lead, RazorSecure *Remote* · December 2019 – Present

- **Platform technical lead:** Set the technical direction for the product and a small engineering team, and own the cross-platform roadmap across the on-train agent, frontend and backend platforms, so an architectural decision on one doesn't break another. Recent designs include push-based configuration management, offline (air-gapped) agent deployment and an RBAC migration.
- **Rust rewrite of the on-train detection agent:** Led the agent's migration from Python to Rust, now complete and running on 1,000+ trains, as an incremental strangler-fig rewrite that shipped one stage per sprint with no big-bang cutover. Built a supervised Tokio runtime in which a faulty monitor restarts rather than taking the agent down, reached parity across the full monitor suite, replaced C-FFI dependencies with pure Rust, and productionised rail-network protocol monitoring end to end.
- **High-performance detection core:** Designed and built a zero-copy Rust deep packet inspection library with eBPF (XDP) acceleration, sustaining sub-microsecond latency at over 1 million packets per second, plus an L2–L7 firewall for an embedded rail security gateway.
- **Cloud portability, on-prem and release engineering:** Made the whole platform deployable on Azure, on-prem and in air-gapped Kubernetes clusters alongside the cloud offering. Consolidated Helm deployments for around 18 microservices into a single platform chart, moved CI onto GitHub Actions, built release automation, and stood up a GitOps-managed on-prem test rack (Talos Linux, Flux).
- **Data-store performance and observability:** Led PostgreSQL alert-store tuning (functional indexes, partition pruning, bulk inserts) that eliminated out-of-memory failures in large batch jobs, alongside MySQL, Elasticsearch and RabbitMQ upgrades. Built an authenticated Grafana proxy and instrumented the platform with Datadog APM tracing.
- **Test-evidence platform:** Designed and built, from scratch, the platform the company now uses for all of its manual test execution and evidence capture: a Rust (Loco) API and React/TypeScript app with a versioned test library, immutable test-plan snapshots, requirements traceability, role-based sign-off with SSO, an MCP server for AI agents and PDF test reports.
- **Engineering and test tooling:** Built a native Rust (GPUI) desktop configurator for agent and monitor configuration, ratatui traffic generators that produce realistic train-network and GPS data along real rail routes, and a multi-tier BDD harness that runs end to end through a real MQTT broker in CI.

Software Engineer, Helitune/Beran Instruments *Torrington, North Devon* · June 2015 – December 2019

- **Next-generation protection and condition monitoring:** Development and systems engineering of a next-generation system for large-plant monitoring. Championed the use of Rust for several of the system modules, running a dockerised signal-processing and logging system on an embedded target — being deployed at the NASA Ames Research Center.
- **Rotor Track and Balance systems:** Developed and maintained a best-in-class RTB system in both carry-on and permanent-fit configurations, written to DO-178 Level C and D.
- **Modular C++ signal-processing system:** Designed an all-new, module-based C++ signal-processing system that let existing and new products move onto a single platform. Being easily testable and replaceable, it shortened development time, made the system more robust, and fit well with agile working.
- **C#/WPF applications:** Developed and maintained several large (100,000+ line) C#/WPF projects across the whole software lifecycle, and built a configuration-driven automated test application that saved hundreds of hours and caught production faults before release.
- **Version control and CI:** Led the company's move from TFS to Git — teaching teams distributed version control and introducing code review — and built the CI workflow around it with packer/vagrant, vSphere and TeamCity.

Skills

Systems Programming	Rust (Tokio async, FFI elimination, multi-call binaries, eBPF/XDP), C++, C
Rust Applications & UI	Loco, GPUI desktop apps, ratatui TUIs, WebAssembly (ratzilla)
Backend & APIs	Python (FastAPI, Pydantic v2, Celery, uv), MCP servers, OAuth 2.0/PKCE & SSO
Frontend	TypeScript, React (Mantine, TanStack Query, React Router), Vite, Playwright
Databases	PostgreSQL (advanced indexing & partition tuning), MySQL, SQLite, QuestDB, Elasticsearch, Redis, RabbitMQ
Containerisation & Orchestration	Docker, Podman, Kubernetes, Helm, GKE, AKS, Talos Linux, Rook-Ceph, MetalLB
Infrastructure & Release Engineering	Terraform/Terragrunt, Scaffold, Pants, Replicated, Harbor, GitHub Actions, ArgoCD, FluxCD, SOPS, bors-ng merge queues, SonarCloud
Observability	Grafana, Datadog APM, statsd
Domain & Protocols	Rail network protocols, ELR/miles-and-chains, DPI/Suricata, SNMP, MQTT, ARP, GPS/NMEA/gpsd
Engineering Practice	Strangler-fig migration, TDD & BDD (Pytest, cargo test), Conventional Commits & release automation, ADRs, C4/D2 architecture modelling, secure-by-design
Leadership	Cross-platform roadmapping, technical-direction setting, migration & rollback planning, effort estimation
Project Organisation	GitHub Issues & Epics, JIRA, Confluence, Agile/SCRUM
Previously (not current)	C#/WPF, F#, Java, PowerShell, SQL Server, NUnit, GoogleTest/GoogleMock, TeamCity, vSphere, packer/vagrant

Projects

Author , STDA Sailboat Simulator	<i>GitHub</i> · 2023 – Present
<ul style="list-style-type: none">Building a physics-based simulator to design, develop and test an autonomous sailboat before deploying to real hardware.Ported the 6-degree-of-freedom (6-DOF) dynamics core to Rust from a published autonomous-sailing model (Sailing Team Darmstadt, IRSC 2018), and extended it with route-following, real nautical-chart navigation and a speed-polar calibration instrument.Rust and Python, working toward calibrating the model against real-world data for a hardware retrofit and autonomous-control trials.	
Author , Homebox Home Assistant Add-on	<i>GitHub</i> · 2025 – Present
<ul style="list-style-type: none">Home Assistant add-on that packages Homebox (a self-hosted home inventory tool) for one-click installation through the Home Assistant Supervisor.My most-starred public project, used by the wider Home Assistant community.Maintained add-on repository with a containerised build and release flow.	
Author , mdBook combiner	<i>GitHub</i> · 2023 – Present
<ul style="list-style-type: none">Small tool that combines mdBooks from multiple repositories into one site, shipped through automated releases.	
Author , jjui-pm	<i>GitHub</i> · 2026 – Present
<ul style="list-style-type: none">CLI plugin manager, written in Rust, for jjui — a TUI for the Jujutsu (jj) version-control system.Manages Lua plugins by reading and modifying the jjui config file; installable via cargo.Reflects deep day-to-day use of Jujutsu and the Rust terminal-UI ecosystem.	
Contributor , Rust nRF52 HAL	<i>GitHub</i> · 2018 – Present
<ul style="list-style-type: none">Experienced developing at the cutting edge, implementing Rust only hardware abstraction on a micro-controller.Being the first to run Rust on the nRF52840 chip proved a real problem solving challenge.	

Education & Training

BSc. in Computing and Psychology , The Open University	<i>England</i> · 2011 – 2015
<ul style="list-style-type: none">Software development and the psychology behind user interface design	
C for Real-Time Developers , Feabhas	<i>Royal Wootton Bassett</i> · 2016
<ul style="list-style-type: none">Intensive five day course on writing low-level C with and without a real time operating system.	
Advanced C++ Development , Feabhas	<i>Royal Wootton Bassett</i> · 2016
<ul style="list-style-type: none">Five-day course on C++ for micro-controllers: its pitfalls and advanced features.	