

Sylvan Bloch

Senior Rust Engineer | Ultra-Low-Latency Trading Systems & Distributed Infrastructure

UTC+2 +27722056229 sylvanbloch@gmail.com Portfolio Website

dVeon-loch Sylvan Bloch

Summary

Systems Engineer with 5+ years of experience, including 3+ years of production Rust focused on low-latency trading infrastructure and exchange connectivity systems. Architected multi-chain crypto arbitrage platforms integrating major CEXs and DEXs using Tokio, custom WebSocket infrastructure, and real-time market data pipelines for latency-sensitive execution. Currently developing a high-performance FIX protocol session layer as part of a larger multi-exchange market data collection and analysis platform, including session management, sequencing, and optimized message encoding. Experienced operating production Rust systems across AWS/Linux environments with Docker and CI/CD workflows. Proven ability to independently drive the full engineering lifecycle including research, architecture, infrastructure, deployment, observability, optimization, and operational reliability within high-ownership startup environments.

Experience

Stealth Startup

Rust Software Engineer

Remote

July 2024 - Present

- Low-Latency Trading Bot Architecture:** Co-architected and optimized a multi-chain crypto arbitrage trading bot using basis arbitrage strategy across Ethereum and Aptos, integrating CEXs (Binance, Bybit, OKX perpetuals) and DEXs (Hyperliquid, Decibel). Leverages Tokio async runtime for concurrent order execution and resource efficiency.
- Custom Low-Latency WebSocket Layer:** Built and maintained custom WebSocket client and server implementations using tokio-tungstenite, leveraging lock-free architectures and zero-copy deserialization techniques to optimize real-time market data streaming and order execution paths. Reduced message processing latency and improved connection stability under high-volume market conditions.
- Organization Library Architecture:** Led migration from fragmented duplicated dependency layers across projects into a unified organization library using feature-gating, crate splitting, and dependency isolation. Reduced build times by 60% while significantly improving maintainability, modularity, and development velocity across services.
- Blockchain Threat Detection System:** Led end-to-end development of Ethereum hack detection tool in Rust, integrating real-time mempool streaming via Reth, trace analysis, opcode interpretation, and calldata decoding.
- Technologies:** Rust, Tokio, tokio-tungstenite, ethers-rs, alloy, Reth, Ethereum, Aptos, CEX/DEX arbitrage, Binance, Bybit, OKX, Hyperliquid, Decibel, Mempool Analysis, WebSocket, High-performance Systems

Dataswyft

Lead Rust Engineer and Architect

Remote

August 2025 - Present

- Lead Architect of Platform Team:** Leading replacement of Scala-based microservices with strategically planned Rust components, architecting for performance, reliability, and correctness as the core backend powering all Dataswyft technology.
- Observability-Driven Architecture Decisions:** Built out structured logging and metrics to guide migration: identified which services were real bottlenecks, measured latency impact of replacements, and distinguished high-value rewrites from low-impact cleanup.
- Technologies:** Rust, Microservices, Scala, Metrics, Observability, Distributed systems

<https://dataswyft.com/>

Civil Designer

Rust & C++ Software Engineer

Cape Town, South Africa (Hybrid)

August 2023 - April 2025

- Multi-threaded Algorithm Optimization:** Designed and implemented multi-threaded algorithms that reduced rendering backend frame-time by over 400%, demonstrating deep systems optimization and low-level performance engineering.
- High-Performance Rust Analytics Pipeline:** Solo designed and deployed Actix-web, SQLx, Postgres backend on AWS EC2, ingesting analytics data from thousands of concurrent CAD users, fully dockerized with custom images.
- Rust Pioneer and Infrastructure Owner:** Pioneered Rust adoption at the company from zero, architected entire CI/CD and version control infrastructure on self-managed Ubuntu Linux with Docker and GitLab, including custom images for Rust cross-compilation.
- Advanced WebView2 and WASM Integration:** Integrated Rust into legacy MFC application via Microsoft WebView2, creating a custom JavaScript compatibility layer linking C++ code with WASM-based reactive Rust components.
- Technologies:** Rust, C++, Actix-web, SQLx, SeaORM, Postgres, AWS EC2, Docker, GitLab CI/CD, Vulkan, DirectX12, Leptos, WASM, OAuth 2.0

<https://civildesigner.com/>

Rocketseed

Rust Developer

Cape Town, South Africa (Hybrid)

April 2025 - August 2025

- Core Rust Software Lead:** Responsible for maintenance and improvement of core Rust application for high-performance email data processing where resource efficiency and performance are paramount.
- Legacy C++ Maintenance:** Maintained core tech stack during transition to Rust, working with unmaintainable codebase lacking company C++ expertise.
- Technologies:** Rust, C++, Email Data Processing, Performance Optimization, Legacy Migration

<https://rocketseed.com>

Peralex

Software Development Engineer

Cape Town, South Africa

April 2021 - August 2023

- Enhanced OpenGL Rendering Engine for Visualization:** Maintained and optimized an in-house OpenGL rendering engine, improving graphing and data visualization capabilities.
- Developed High-Performance GUI Applications:** Led C++ and Java GUI development for commercial spectrum monitoring tools, utilizing OOP and functional programming principles to ensure efficiency and responsiveness.
- Established Team Coding Standards:** Introduced new coding standards emphasizing simplicity, reduced cognitive complexity, and clean code practices, enhancing team productivity and code quality.
- Database Integration Expertise:** Worked with PostgreSQL and H2 databases to support data-intensive applications and seamless backend functionality.

<https://www.peralex.com/>

Peralex

Vacation Work Intern

Cape Town, South Africa

December 2019 - August 2020

- Developed Advanced GIS and Radar Management Interfaces:** Contributed to a sophisticated TypeScript-based GIS for real-time aircraft tracking through a passive radar system, alongside a C++/Qt GUI for direct hardware management of the radar.
- Implemented Elastic Stack for System Metrics:** Pioneered the use of the Elastic Stack within the company to generate and analyze detailed performance metrics for the radar system, enhancing system monitoring and data-driven decision-making.

<https://peralex.com>

Projects

Exchange-API (Rust Crate)

Personal exchange connectivity library in Rust

April 2026 - Present

- FIX 4.4 Protocol from Scratch:** Built complete FIX 4.4 protocol layer in Rust including full message parsing and encoding with SOH-delimited tag-value fields, checksum computation and validation, session state machine (Disconnected, LogonSent, Active, LogoutSent, LogoutReceived), and full round-trip parse/encode tests.
- Order Book Management:** Implemented Binance adapter with local order book management using incremental depth updates and snapshot sync, parsing spot and futures WebSocket streams for trades, order books, and tickers.
- Abstract Exchange Trait and Runtime:** Designed provider-agnostic Exchange trait for WebSocket URL resolution, subscription message generation, and stream parsing. Runtime manages connection lifecycle and output routing per exchange.
- WebSocket Auto-Reconnect:** Built ws-proto crate with automatic reconnection using exponential backoff with jitter, configurable max retries, and connection state tracking.
- Multi-Sink Output Pipeline:** Feature-gated output routing supporting Kafka (rdkafka), Redis (fred), file, and broadcast channel outputs. Each exchange streams to all configured sinks concurrently via Tokio broadcast channels.
- Technologies:** Rust, Tokio, tokio-tungstenite, FIX protocol, Binance REST/WebSocket API, Kafka, Redis, async-trait, exponential backoff, order book management

<https://github.com/dVeon-loch/exchange-api>

Exchange-Tools

Exchange dashboard and data-analysis platform

April 2026 - Present

Full-stack exchange data streaming platform (Leptos SSR + Axum) consuming exchange-api for live market data. Stores to TimescaleDB via SQLx, streams to browser over WebSocket, deployed on Fly.io with Docker.

Distributed architecture:

- Ultra low-latency data ingestion server using custom WS protocol streaming and data normalization (FIX protocol in-progress)
- Data split into two main consumers: Redis with pub/sub to horizontally-scaled Web backends + Kafka for offline data collection and storing to TimescaleDB for later analysis

Currently in-progress, focusing on major exchanges initially but architecting for the maximum amount of data-throughput as possible.

<https://exchange-tools.sylvan-b.com>

Education

University of Cape Town

Bachelor of Science in Engineering, Mechatronics Specialization

Honours

Bachelor of Science in Engineering

December 2020

References

Tyler Weir

Dataswyft CTO & Direct Manager

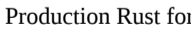
Email: tyler.weir@gmail.com

Linkedin: <https://www.linkedin.com/in/tylerweir/>

Skills

Rust

Production Rust for ultra-low-latency systems: Tokio async runtime, custom networking via tokio-tungstenite, trading infrastructure, blockchain systems, and large-scale library architecture with feature-gating and crate splitting.



Rust, Tokio, Low-Latency, Concurrency, Async, Performance Optimization, Systems Programming, tokio-tungstenite

Low-Level Systems & Performance

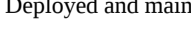
Low-level systems and performance engineering focused on low-latency networking, concurrent processing, and high-throughput runtime optimization. Experience designing multi-threaded and lock-free architectures using sync and async runtimes (Rayon, Tokio, Smol), custom WebSocket networking, and zero-copy deserialization techniques for latency-sensitive workloads. Background in optimization-critical C++ systems development with Vulkan and DirectX12, optimizing frame-time stability, memory efficiency, and parallel execution across large-scale production codebases.



Rust, C++, Multithreading, Low-Latency Networking, Performance Optimization, Systems Optimization, Smol, Rayon, Tokio, Lock-Free

Infrastructure / DevOps

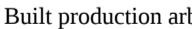
Deployed and maintained production Rust services across Cloud (AWS, GCR, Azure) and bare-metal Linux environments using Docker-based infrastructure and automated deployment workflows. Managed scalable backend infrastructure, observability, and runtime reliability for latency-sensitive trading systems operating in real-time execution environments.



Docker, Kubernetes, CI/CD, AWS/GCP, Linux Administration, Monitoring/Observability, Distributed Deployment Systems

Trading Systems & Exchange Connectivity

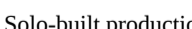
Built production arbitrage trading bot with multi-exchange integration, implemented FIX 4.4 protocol from scratch, local order book management with incremental depth sync, and WebSocket auto-reconnect with exponential backoff.



FIX protocol, WebSocket, CEX/DEX, Binance, Bybit, OKX, Hyperliquid, Order Books, Trading Systems, Exchange API, Market Data

Backend & Distributed Systems

Solo-built production Rust web services (Actix-web, Axum), distributed microservices architecture, cloud deployment on AWS/Azure, Docker containerization, CI/CD from scratch.



Rust, Axum, Actix-web, PostgreSQL, SQLx, Docker, AWS, Azure, Distributed Systems, Microservices

English

Fluent/Mother Tongue

