Sadia Rahman Senior Rust Developer

Toronto, ON · sadia.rahman@gmail.com · +1 647-555-0184 · linkedin.com/in/sadiarahman

PROFILE SUMMARY

- **Senior Rust Developer** with **9 years** in **database and storage engines**, now building the **write path and compaction** of a **time series database**, after 4 years on a **C++ in-memory planning engine**.
- Builds in **Rust** with **Tokio**, **Apache Arrow**, **DataFusion** and **Parquet** on **object storage**, and reads C++ daily for the parts of the older engine still in production.
- Strong in **async and lock-free concurrency**, **heap profiling** with dhat and heaptrack, and **crash-safety testing**, and backs every storage change with **criterion benchmarks** and **fault injection**.
- Works with the **query team**, **SRE**, **support engineers** and **enterprise customers** on upgrade plans, and takes the **storage on-call rotation** one week in six.
- Leads **design reviews** for storage RFCs, mentors **3 engineers** on unsafe code review, and sends fixes upstream to **Apache Arrow and DataFusion**.

TECHNICAL SKILLS

Languages:	Rust, C++, SQL, Python, Bash
Async & Networking Crates:	Tokio, tonic, hyper, serde, object_store, futures
Systems & Interop:	unsafe Rust, cxx, bindgen, crossbeam, parking_lot, jemalloc
Wasm & Embedded Tooling:	WebAssembly, wasm-bindgen, wasm-pack
Testing & Verification:	cargo-nextest, proptest, cargo-fuzz, Miri, loom, fault injection
Profiling & Observability:	criterion, perf, dhat, heaptrack, tracing, OpenTelemetry
Build, CI & Supply Chain:	Cargo workspaces, clippy, cargo-deny, cargo-semver-checks, CircleCI, Docker

EDUCATION

University of Waterloo — **B.Math. in Computer Science** — Waterloo, ON · Sep 2012 to Apr 2017

WORK EXPERIENCE

InfluxData — **Senior Software Engineer** — Toronto, ON (remote) · Apr 2021 to Present

- Rebuilt the **write-ahead log** in Rust on **Tokio**, batching writes per table through bounded channels and borrowed buffers instead of copies, raising sustained ingest to **1.4M rows a second** per node.
- Reworked Parquet compaction after **dhat** heap profiles showed whole files held in memory, moving to a streaming merge over **Apache Arrow** batches and cutting peak compactor memory from 22 GB to **5.8 GB**.
- Built a crash-injection harness with **proptest**-generated workloads that kills the server mid-write and checks every acknowledged row, catching **11 durability bugs** before release.
- Wrote safe Rust bindings with **cxx** to the C++ decoder still used for legacy files, with every unsafe call behind a checked wrapper, closing **4 crash reports** in the first quarter.
- Split the line protocol parser into its own crate and compiled it to **WebAssembly** with wasm-bindgen so the web UI checks writes in the browser, cutting rejected writes by **38%**.

Kinaxis — **Software Developer** — Ottawa, ON · Jul 2017 to Mar 2021

- Wrote C++ query operators for the in-memory planning database, cutting scenario recalculation for the largest customers from 14 minutes to **5**.
- Moved the snapshot file reader from C++ to Rust behind a C ABI as the team's first Rust component, and crash reports from that code fell from 9 a year to **0**.
- Added benchmark suites to CI for the 20 most used operators, catching **12 slowdowns** before customer releases.