

S. M. Faisal Rahman

London, UK • smfaisal1648@gmail.com • +44 7377 872865 • linkedin.com/in/faisal-rahman-88174512a • github.com/Frank-en-stein

Executive Summary

Software engineer (7+ years) specialising in high-throughput, low-latency systems and data pipelines. Experienced in designing and shipping production indexers with sub-second read paths, multi-service ingestion systems with bounded tail behaviour (p99/p999), and organising focused optimisation workstreams that materially reduce cost, latency or increase throughput. I enjoy designing microbenchmarks, hunting hot-path allocations and syscalls and ingestion systems where milliseconds matter.

Core Technical Skills

Languages: Rust (jstz runtime, indexer), Go (L1 blockchain indexer, SDK), C/C++ (rendering optimisation, toolchain), Java (distributed event systems), Python (async pipelines).

Systems & Performance: lock-free concurrency (atomic fetch_update, RAIL slot management), multi-tier caching (2Q-LRU, watermark eviction, TTL deduplication), checkpoint-based recovery with idempotent upserts, binary protocol parsing (Blake2b verification, nested rollup messages).

Distributed Systems: event-driven architectures (tokio async channels, RabbitMQ), batching with adaptive backpressure (1000 msg/batch, 100ms polling), exponential backoff retry with circuit breakers, protocol state machine migrations (4 Tezos upgrades: Paris, Quebec, Rio, Seoul).

Financial Systems: mutez arithmetic (10^{-6}), proportional reward distribution with rounding correction, double-entry staking flows, hexadecimal rights encoding (128x compression) and Nested Smart function trace with graph reconstruction.

Infrastructure: GCP/Kubernetes (StatefulSets, multi-tier node pools), Terraform/Terragrunt, Prometheus/GCloud observability, Docker multi-arch builds (AMD64/ARM64).

Professional Experience

Trilitech — Software Engineer (jstz and blockchain indexers)

London, UK. Nov 2023 – Present

Engineering the jstz verifiable compute runtime (Rust/RISC-V kernel), production infrastructure, and two blockchain indexers: tzstats.com (Tezos L1, Go) and jstz indexer (Rust). Maintain tzgo SDK powering exchange integrations.

- **[Indexer performance — Go]:** re-architected multi-tier cache system (account: **1k→210k** entries, operation: **98k→10M** entries, bigmap: **8k→839k** entries) with watermark-based eviction (75%/85% thresholds) — reduced disk I/O by **~90%**, stabilised intermittent indexer lag to subsecond latencies.
- **[Protocol migrations — Go]:** owned indexer migration for 4 major Tezos financial protocols (Paris, Quebec, Rio, Seoul) including Paris consensus migration requiring massive overhaul of Tezos's staking, snapshot, baking rights and rewards calculations accurately.
- **[Indexer — Rust]:** designed and shipped jstz-indexer from scratch: event-driven architecture with tokio async channels, semaphore-based concurrency limiting, checkpoint-based recovery with atomic status tracking, LRU cache with TTL deduplication — **80ms p95/250ms p99** at **100k RPS** on GCP, **30ms p95/120ms p99** at **1M RPS** ceiling on bare-metal, 100% consistency guarantees.
- **[Lock-free concurrency — Rust]:** implemented generic const-parameterised concurrency limiter using atomic fetch_update with RAIL slot management — zero-overhead call depth limiting preventing resource exhaustion without mutex contention.
- **[Binary parsing — Rust]:** authored deserializers for nested Smart Rollup messages with Blake2b hash verification, signature validation, and ticketer-aware inbox parsing; integrated with jstz_kernel library for deterministic execution.
- **[Middleware with finsys precision — Go/.NET]:** implemented proportional staking reward distribution with mutez-level rounding correction (floor + remainder redistribution), hexadecimal rights encoding achieving **128x** compression (8192 blocks→2048 hex chars), and luck/performance metrics calculation.
- **[SDK / Integrations — Go]:** maintained tzgo SDK powering exchange integrations (Coinbase, Binance); implemented deterministic transaction flows and protocol-upgrade handling for finance-grade accuracy.
- **[Infrastructure — GCP/Terraform]:** led architectural migration from self-hosted nodes (c2d-highmem-32) to federated RPC endpoints (e2-highmem-8) — **75% VM downscale**, **\$3–5k/month** savings; authored Terraform across 2 production deployments with automated snapshot rotation (15min RPO).
- **[Fault tolerance — Go/Rust]:** designed automated recovery systems: mutex-protected retry counting with exponential backoff, last-good-block rollback, Prometheus metrics integration, graceful degradation with controlled process exit after max retries.

Amazon, AWS S3 — Software Engineer (Events)

Cambridge, UK. Jul 2021 – Nov 2023

Engineered global S3 event bus and durability subsystems at multi-billion event/day scale with deterministic latency guarantees.

- **[Pipeline — Java/Go]:** designed and shipped highly concurrent multi-service pipeline integrating four AWS services; sustained stable tail latency under extreme event volumes (billions/sec worldwide).

- **[Codec optimisation — C/Java]:** migrated streaming pipelines from gzip to Brotli with optimised encoder/decoder stacks — cut storage and streaming cost by ~50% while preserving throughput and durability guarantees.
- **[Testing infrastructure — Java]:** built deterministic integration/canary harnesses reducing critical test wall time from 3.5h to 12m, enabling faster deploy windows and safer rollouts; reduced S3-wide deployment time ~5%.
- **[Durability — AWS infra]:** architected durability metrics, alerts and reconciliation flows (idempotent receipts + check-pointing) to enforce 99.99% durability and 99.999% auditing accuracy across event archival components.
- **[Integrity — HW accelerated CRC-32C]:** implemented end-to-end checksums and checksum-based dead-lettering to detect and quarantine intermittent corruption at scale.
- **[Disaster recovery — EMR/MapReduce]:** developed replay tooling to restore archived event sets; introduced measured backpressure into frontline services based on microbenchmark results.
- **[Hot-path profiling — JVM]:** led syscall/allocation hotspot elimination via perf/flamegraphs; introduced zero-copy paths in latency-sensitive loops, reducing CPU cycles/event by 30–50% in event archival service.

EMBL-EBI — Backend Engineer (Europe PMC, europepmc.org)

Cambridge, UK. Feb 2020 – Apr 2021

Engineered scalable event-driven life science research article ingestion pipelines for paper search engine. Work cited in Nucleic Acids Research (2021).

- **[Inference pipeline — Python/PyTorch]:** integrated PyTorch inference workflows into event-driven backends; implemented scalable RabbitMQ microservices to process high-volume document streams with deterministic throughput.
- **[Event-driven backend — RabbitMQ/Java]:** migrated legacy ingestion flows to microservice architecture; introduced VM-independent smart thread scaling for zero-config deployment across heterogeneous environments.
- **[UI migration — Vue]:** drove completion of Europe PMC's frontend migration to Vue, modernising user experience and aligning with event-driven backend services.

Samsung R&D Institute — Software Engineer

Dhaka, Bangladesh. Apr 2018 – Dec 2019

Awarded Icon Engineer Certificate; achieved SWC Professional Certificate for programming competency.

- **[Rendering — C/C++]:** optimised Galaxy Note pen-stroke rendering library by redesigning memory layout and allocation strategies; stabilised frame-timing variance and reduced ARM64 device power draw by ~5%/day for powerusers benchmarked.
- **[Toolchain / Builds]:** migrated large native C++ codebase from GNU to Clang; created ~50% faster build toolchain through zero-overhead optimisations.
- **[Mobile / SDK — Swift]:** led iOS social feed modules; architected cross-app SDKs for Samsung Cloud, implemented complete testing framework and CI pipelines; solo implemented ~20% of multi-product scale codebase.

Far-East IT Solutions — Software Engineer (Remote)

Dhaka, Bangladesh. Nov 2017 – Apr 2018

Engineered end-to-end flight booking platform (Node.js backend, Postgres, Redis).

- **[Algorithm — Node/Python]:** designed multipath A* search with weighted Gaussian heuristics to optimise multi-leg multi-carrier routing under fare constraints — reduced query cost and improved recommendation quality.
- **[OCR / SSR]:** implemented OCR ingestion (Tesseract) and partial server-side rendering to lower latency and improve throughput.

Publications

- **Nucleic Acids Research** (2021) — implemented production data pipelines and indexing. academic.oup.com/NAR D1507
- **IEEE** (2019) — sensor-swarm fire evacuation system. IEEE Xplore: 8631910
- **IEEE** (2019) — IoT swarm signal reactive path optimisation. IEEE Xplore: 8631945

Education

BSc (Hons), Computer Science & Engineering — Military Institute of Science and Technology (MIST), Dhaka 2014–2018

Thesis: DTLS multicast protocol — design and benchmarks for low-compute encrypted IoT protocols using Contiki/Cooja.

Selected Awards & Activities

- Champion — MIST Robofest (Line Follower Robot) 2015/2016; Champion — Robo-fight (MIST Robofest 2015).
- Champion — IUT Esonance (Fast Line Follower Robot, 2016); Champion — NSU Cybernauts (2016); Top-10 inter-university programming.
- Problem setter — Regional University Programming Contest, Bangladesh (2022).
- National runner-up — Bangladesh Mathematical Olympiad (secondary school).