MacBobby Chibuzor

Senior Distributed Systems & Backend Engineer (Rust · Go · TypeScript · Python)

7+ years designing and operating distributed systems and financial infrastructure under real production load — exchange order books, FX liquidity routing, multi-bank settlement pipelines, event-driven microservices, and Kubernetes-native cloud platforms. Proven depth in backend engineering (Rust, Go, TypeScript, Python), cloud infrastructure (AWS, GCP, Terraform), and production observability at scale. Active open-source contributor to Kraken's Rust CLI and SDK ecosystem, CoW Protocol's solver pipeline, and CosmWasm governance engines. Published technical author on LogRocket and freeCodeCamp.

macbobbychibuzor@gmail.com · github.com/theghostmac · theghostmac.codes · theghostmac.codes/resume · linkedin.com/in/macbobbychibuzor · Remote (EMEA / South Africa)

OPEN SOURCE & EXCHANGE INFRASTRUCTURE CONTRIBUTIONS

Kraken CLI (krakenfx/kraken-cli) · *Open Source Contributor (Rust) · 2026*

- **PR #24 (Merged) / PR #20 — Spot Paper Trading Slippage Simulation:** Implemented configurable `--slippage-rate` simulation for paper market orders in Rust — filling market buys at `ask × (1 + slippage_rate)` and sells at `bid × (1 − slippage_rate)` with backward-compatible state serialization and full integration test coverage so paper P&L stops overstating live execution.
- **PR #54 (Open) — Trailing-Stop Anchor Synchronization on Order Edit:** Fixed `FuturesPaperState::edit_order` in the Futures Paper engine so editing `--stop-price` on an open `trailing-stop` order also updates `trailing_anchor`, preventing subsequent price ticks from overwriting user edits from a stale anchor.
- **PR #55 (Open) — Triggered Order Observability in Futures Paper CLI:** Surfaced `stop_price` and `trigger_signal` in both JSON and table outputs of `kraken futures paper orders` and `order-status`, enabling automated agents and traders to inspect trigger levels on open `stop`, `take-profit`, and `trailing-stop` orders.
- **PR #56 (Open) — Opposite-Side Position Margin Netting for Triggered Orders:** Fixed `FuturesPaperState::place_triggered_order` to net margin against existing opposite-side positions via `compute_margin_delta` instead of unconditionally reserving full initial margin (`notional / leverage`) on protective stop-loss and take-profit orders.

Kraken API SDK (krakenfx/kraken-api-sdk) · *Open Source Contributor (Rust, Go) · 2026*

- **Issue #2 (Proposed) — Cross-Binding REST Architecture & Custom Base URL Injection:** Proposed phased Go & Rust SDK architecture covering REST client ergonomics, cross-binding error contracts, and custom base URL injection on `SpotRestClient` for deterministic mock-server testing.

CoW Protocol (cowprotocol/services) · *Open Source Contributor (Rust) · 2026*

- **PR #4127 — Unified Order Execution Simulator:** Built unified order simulation logic for the CoW Protocol solver pipeline in Rust.
- **PR #4088 — Ergonomic Quote v2 Endpoint:** Implemented an ergonomic `/quote` v2 endpoint targeting core DEX settlement protocol infrastructure in Rust.

Tempo Foundry (tempoxyz/tempo-foundry) · *Open Source Contributor (Rust) · 2026*

- **PR #256 — Fork-Mode Invariant Fuzzer Fix:** Diagnosed and fixed an invariant test hang when executing fuzz runs with `--fork-url` in Rust.

DAO-DAO (daodao.zone · github.com/DA0-DA0/dao-contracts) · *Rust Engineer & DAO Member · 2023 – Present*

- Contributed active-threshold voting logic to the `cw4` voting module, NFT ID address querying, and comprehensive governance and membership test suites on the CosmWasm DAO governor — a replicated Rust state machine governing thousands of on-chain organisations across the Interchain.

PROFESSIONAL EXPERIENCE

Integration & Support Engineer · WeWire

Jun 2026 – Present · Remote

FX Liquidity Routing & Execution Engine ("Project LP FX")

- Built the rate-provider abstraction and routing layer for WeWire's FX liquidity engine: a strategy-pattern interface (`BaseRateProviderStrategy`, `RateProviderRegistry`, `RateRoutingService`) supporting global and per-customer provider priority lists, per-provider markups, and automatic fallback across liquidity providers (OpenExchangeRates, Hercle RFQ).
- Designed and delivered an end-to-end hybrid FX execution engine — covering background rate polling, caching, validation, customer rate locking immune to downstream repricing, RFQ-based third-party trade execution with `Client-Id` idempotency, FX slip/margin accounting, and a Cockpit admin UI for failed-trade recovery.
- Audited the multi-provider execution flow and shipped four critical production safety and revenue fixes in a single stacked release: atomic quote claim (`executionClaimedAt` guarded `updateMany`) eliminating a double-execution race condition, synchronous bypass for internal conversions to eliminate false-positive TTL alerts, provider-level markup application across third-party RFQ quotes, and global vendor-name masking in exception filters.
- Authored the foundational engineering specifications the team builds against: ADR-004 (FX strategy pattern & markup architecture), the FX rate routing architecture specification, and the *Adding a Rate Provider* developer guide.

Banking Partner Integrations, Contract Testing & Webhook Infrastructure

- Led end-to-end integration of FirstBank Ghana for GHS virtual account issuance and GHST wallet pairing, replacing the incumbent provider with zero customer disruption through a staged rollout and automated deposit reconciliation.
- Designed ADR-005 and built a 78-test, no-mocks Provider Integration Contract Test Suite running against real PostgreSQL, Redis, and Bull queues in CI — uncovering 14 cross-provider compliance and reliability gaps (unverified inbound webhook routes, silent poller failures, and stuck `REQUESTED` state transitions).
- Engineered Bull-backed deposit reconciliation workers with cross-task Redis distributed locking and Redis-backed alert throttling (`PollFailureAlerter` requiring 3 consecutive recovery ticks before re-arming) across FirstBank Ghana and CalBank deposit pollers.
- Audited the Bridge USD banking integration end-to-end: root-caused a silent workflow-engine hang caused by circular Axios error serialization in `extractErrorPayload`, fixed an idempotency-key reuse flaw that permanently blocked USD account re-requests after a single 4xx failure, and shipped structured, vendor-masked KYC rejection reasons (`Account.statusReason`) across API and webhook payloads.

Automated FX Rate Broadcast Engine & Production Operations

- Reverse-engineered an 84-tab trading desk FX workbook into a deterministic 33-pair pricing and WhatsApp broadcast engine serving 21+ institutional client groups daily, complete with golden tests matching spreadsheet outputs to 1e−8, Google Sheets live cell-label validation, and a maker-checker WhatsApp approval workflow.
- Built an internal Sumsub KYC operations tool implementing HMAC-SHA256 request signing from scratch to inspect and manage sandbox applicants without Postman, and supported ~15 engineering teams (and judged ~18 submissions) during WeWire's HACKHIVE 2.0 hackathon.

Stack: *TypeScript, NestJS, Prisma, PostgreSQL, Redis, BullMQ, AWS (ECS, RDS, CloudWatch, S3), Sumsub, Bridge, OpenPayd, FirstBank Ghana, CalBank, Hercle RFQ, Docker, GitHub Actions*

DevOps / Infrastructure Engineer (Contract) · Rocksteady Technology

Mar 2026 – Aug 2026 · Remote

Performance Engineering & Incident Recovery

- Led the production rollout of a reports-service architecture overhaul, replacing a Kubernetes Job spawned per report with a persistent Redis-backed worker pool (Java/Spring) and cutting report generation latency by ~75× (from ~10 minutes to ~8 seconds).
- Resolved a cascading multi-system outage on a customer-facing enterprise environment within a single day — untangling a masked database misconfiguration, ~1,600-commit deployment drift, a frontend build defect, ingress misrouting, and broken legacy data synchronization.
- Recovered 34,000+ records of inaccessible operational data from a legacy PostgreSQL path and designed a live two-way change-data-capture (CDC) synchronization pipeline to eliminate dual-write consistency risks.

Cloud Cost Optimization & Platform Automation

- Drove consolidation of 8 QA and non-production Kubernetes environments onto a single shared KIND host architecture, targeting ~$19,000/month in non-production AWS savings.
- Engineered the consolidated platform end-to-end: Terraform-provisioned EC2 host, AWS SSM remote operations, IAM roles for ECR and Secrets Manager, `ingress-nginx` with VPN routing, and an S3-staged Helm render/apply pipeline wired into Prometheus, Loki, and Grafana.
- Migrated multi-tenant frontends to an AWS CloudFront CDN architecture (`external-dns`, version-cookie routing) across dozens of production tenants, unified Alertmanager routing into a single on-call Slack workflow, built idempotent operator CLIs to replace ad-hoc SQL fixes, and served on the weekly 24-hour on-call rotation.

Stack: *Python, TypeScript, Bash, Terraform, AWS (EC2, EKS, CloudFront, S3, SSM, Secrets Manager), Kubernetes, KIND, Helm, Docker, ArgoCD, Jenkins, GitHub Actions, PostgreSQL, MongoDB, Redis, RabbitMQ, Elasticsearch, Prometheus, Grafana, Loki, Alertmanager*

Senior Backend & Blockchain Engineer (Contract) · Jeroid Pay

Nov 2025 – Dec 2025 · Remote

- Designed a stateful, event-driven Rust backend for concurrent multi-user financial flows over WhatsApp — combining async message routing via Twilio webhook queues, a PostgreSQL-backed state machine enforcing strict transaction ordering, and atomic cross-network settlement guarantees.
- Integrated an LLM-powered intent classification layer (Groq) directly into the backend request pipeline, routing natural-language financial commands to deterministic handlers without brittle rule-

based branching.

- Architected clean hexagonal adapter boundaries between the core domain and network-specific settlement and custody clients (Solana, Base, Polygon, Fireblocks, and Smile ID KYC), keeping business logic network-agnostic.

Stack: *Rust, PostgreSQL, Solana, Base, Polygon, Fireblocks, Smile ID, Twilio, Groq LLM, Docker*

Senior Infrastructure Engineer · Stride Labs

Oct 2024 – Jul 2025 · Remote

Distributed Infrastructure & Platform Engineering

- Consolidated 16 microservices into 8 cohesive deployments — reducing GCP infrastructure spend by ~$5,000/month (50%) while improving service cohesion, lowering inter-service latency, and shrinking failure blast radius.
- Migrated a stateful TEE automation service from brittle JSON file persistence to PostgreSQL with ACID transactional semantics; built full Kubernetes orchestration (Helm charts, rolling updates, readiness/liveness probes) and GitHub Actions CI/CD pipelines.
- Introduced typed TypeScript SDK clients to replace shell-script integration tests, enabling reproducible, parallelisable test runs against a localnet and catching interface regressions prior to deployment.

High-Throughput Token Economics Automation & Data Pipelines

- Built the automated STRD buyback-and-burn auction engine (10-minute Kubernetes CronJobs burning ~5.3M STRD/year), executing multi-step swap routing over the Osmosis Swap Router API with optimal path selection across liquidity pools, slippage guards, and IBC transfers via the Skip API with Packet Forward Middleware (PFM) detection and 2-hop fallback logic.
- Architected the stBGT points-program analytics and indexing service on Berachain using Python/FastAPI: tiered hot/warm caching, automated nightly indexing jobs, and a PostgreSQL write path tuned for append-heavy event ingestion with fast read aggregations.
- Adapted the Uniswap V3 subgraph for the Stride Swap DEX: implemented concentrated-liquidity tick histogram indexing, wrote a full Matchstick unit and integration test suite, and deployed to Goldsky — saving $3,600/month compared to The Graph's hosted service.
- Engineered production Berachain operator bots: a Proof-of-Liquidity (POL) swapper via Ooga Booga, a validator boost-cycle operator with Merkle proof verification, and a real-time validator slashing detector.

Stack: *Go, Python (FastAPI), TypeScript, Solidity, PostgreSQL, Redis, Kubernetes, Helm, Docker, GCP, GitHub Actions, GraphQL, The Graph (Matchstick), Goldsky, Cosmos SDK / IBC*

Lead Software Engineer · Cloudax Holdings LTD

Jun 2023 – Sep 2024 · Remote

Platform Engineering

- Architected and deployed cloud infrastructure on GCP from a greenfield start using Terraform, building reproducible Infrastructure-as-Code (IaC) pipelines for a crypto-fintech product suite.
- Built internal developer tooling and automation scripts in Bash and Python to streamline deployment workflows and eliminate manual operational toil.
- Designed and operated monitoring and alerting stacks with Prometheus and Grafana, giving engineering teams real-time visibility into system health, latency, and anomalous behaviour.

Product Development & Engineering Leadership

- Led a multidisciplinary team of engineers building crypto-native applications — blockchain price oracles, fiat banking integrations, and P2P escrow trading — from initial architecture to the first 2,000 active users.
- Built core backend APIs and on-chain smart contracts powering settlement and escrow workflows.

Stack: *Terraform, GCP, Python, TypeScript, Bash, PostgreSQL, Prometheus, Grafana, Solidity*

Blockchain Security Researcher · Halborn Security

Dec 2022 – Jun 2023 · Remote

- Researched zero-knowledge proof constructions (zkSNARKs) and Layer-2 scalability architectures — focusing on distributed consensus, cryptographic commitments, and state validity proofs.
- Built smart contract security analysis tooling and zkSNARK proofs of concept in Go, Rust, and Python; authored technical security research reaching ~2,100 monthly developer readers.

Stack: *Rust, Go, Python, Solidity*

Software Engineer · IOD Cloud Technologies

Dec 2021 – Jan 2023 · Remote

- Designed cloud infrastructure and automation workflows with Terraform and Docker, and executed production AWS ECS vs. EKS benchmarks for enterprise clients (Cisco, Logz.io, Zoho), measuring scheduling latency, autoscaling behaviour, and cost profiles to guide architecture decisions.
- Instrumented distributed tracing across multi-service environments using OpenTelemetry with Prometheus and Grafana dashboards; authored 15 in-depth technical guides on Kubernetes, observability, and distributed systems.

Stack: *Go, Python, Terraform, Docker, Kubernetes, AWS (ECS, EKS, Lambda), Prometheus, Grafana, OpenTelemetry*

Software Engineer · ReachExt K.K. (EJable.com)

Mar 2019 – Nov 2021 · Remote · Tokyo, Japan

- Built and operated backend services in Go and Python (Flask) for a high-traffic content platform: RESTful APIs, PostgreSQL query optimisation, Redis caching layers, AWS infrastructure (EC2, Lambda, S3), and automated CI/CD pipelines.
- Mentored junior engineers on API design, relational database performance, and cloud infrastructure fundamentals; authored 12+ technical engineering articles.

Stack: *Go, Python, Flask, PostgreSQL, Redis, AWS (EC2, Lambda, S3), Docker*

SELECTED SYSTEMS & ENGINEERING PROJECTS

- **Meridian Protocol (github.com/theghostmac/meridian-protocol) — Rust, Solidity, EVM, SVM (2026):** Intent matching engine with on-chain settlement: built a price-time priority limit order book in Rust, EVM settlement contracts packing 256 fill nonces per storage slot (reducing 256 fills to a single cold SLOAD), and cross-chain event indexers across EVM and Solana VM networks.
- **Order Book & Matching Engine Lab (theghostmac.codes/blog/matching-engines) — TypeScript, React (2025):** Interactive exchange micro-architecture suite accompanying an engineering deep-dive on matching engines: features a live limit order book simulator, a step-by-

step price-time matching algorithm visualiser, and a data-structure algorithmic complexity benchmark.

- **BookFlix (bookflix.app · App Store / Download) — Python, FastAPI, Qwen LLM, Kokoro TTS, RAG (2026):** Production AI reading platform live on the App Store: architected and built the FastAPI backend serving Qwen for spoiler-free chapter recaps and character glossaries, Kokoro for neural text-to-speech synthesis, and Retrieval-Augmented Generation (RAG) over book corpora.
- **regress-cli (github.com/theghostmac/regress-cli) — Rust (2026):** Statistical performance benchmarking CLI in Rust that executes two versions of a binary or service against an identical workload suite and reports with statistical confidence whether latency and throughput improved or regressed.
- **Cross-Chain Liquidity Monitor (github.com/theghostmac/liquidity-monitor) — Rust, WebSocket, PostgreSQL (2025):** Real-time market-data pipeline in Rust that streams and indexes concentrated liquidity events from Uniswap V3 and Raydium pools, tracks orderbook/pool depth changes, and flags cross-venue arbitrage gaps between Solana and EVM chains.
- **Solana Terminal Trading Bot (github.com/theghostmac/solana-trading-bot) — Rust, Solana (2025):** Low-latency command-line trading execution engine for Solana tokens written in Rust (most-starred personal repository on GitHub).
- **Raft Consensus & CRDTs in Rust (github.com/theghostmac/raft-consensus · Write-up · github.com/theghostmac/crdt) — Rust (2024–2025):** From-scratch Rust implementations of the Raft distributed consensus algorithm (leader election, log replication, safety invariants) and state/operation-based Conflict-Free Replicated Data Types (CRDTs).
- **P2P Intent Matching & Bonding Curve Labs (p2p-intent-matching-engine · bonding-curve-test · evm-cosmos-bridge) — Rust, Solidity, Foundry (2024–2025):** Peer-to-peer swap intent matcher with partial fills and Uniswap V3 fallback, hardened exponential bonding curve with overflow/economic-attack verification in Foundry, and a minimal EVM ↔ Cosmos cross-chain bridge in Rust and Solidity.
- **Fintech & Trading Engines in Go (paystack-go-sdk · bankie.go · tradermac · pluto) — Go, PostgreSQL (2023–2024):** Open-source Go client SDK for Paystack's payments API, a PostgreSQL-backed banking ledger service (`bankie.go`), a real-time options trading engine fed by Alpha Vantage (`tradermac`), and a modular blockchain built from scratch in Go (`pluto`).

SELECTED TECHNICAL PUBLICATIONS

- **LogRocket (Rust & Distributed Systems):** A complete guide to running Rust on Arduino (Jul 2024) · Linting in Rust with Clippy (Feb 2023) · Create an API in Rust with SQLite and Rocket (Jan 2023)
- **LogRocket (Fintech, Security & Chaos Engineering):** Inspect your code with Docker and SonarQube (Oct 2022) · Leveraging chaos engineering to test Ethereum clients (Aug 2022) · Smart contract development: Common mistakes to avoid (Jun 2022) · Streamline KYC verification with Hyperledger Fabric and Go (Apr 2022)
- **freeCodeCamp & Engineering Blog:** Haskell programming language: how to install and use Haskell (*freeCodeCamp*, Mar 2022) · 20+ deep-dive engineering articles on Rust, Go, matching engines, and distributed consensus at theghostmac.codes/blog and dev.to/theghostmac.

TECHNICAL SKILLS

Distributed Systems & Financial Infrastructure: Limit order books, price-time priority matching engines, FX quote & liquidity routing (RFQ), double-entry ledgers, multi-rail settlement, idempotency, event sourcing, CQRS, consensus algorithms (Raft/BFT), CRDTs, backpressure, fault tolerance

Languages & Backend Engineering: Rust (Tokio, Reqwest, Clap, Serde, Rocket, CosmWasm), Go, TypeScript (NestJS, Node.js), Python (FastAPI, Flask), Solidity — REST, gRPC, WebSocket v2 streaming, GraphQL, HMAC/mTLS security

Data & Messaging: PostgreSQL (query tuning, indexing, partitioning, Prisma), Redis (distributed locks, Lua scripts, caching, streams), BullMQ, Kafka, RabbitMQ, MongoDB, SQLite, DuckDB, The Graph / Goldsky subgraphs

Cloud, DevOps & Observability: AWS (ECS, EKS, EC2, RDS, CloudFront, S3, SSM, Secrets Manager, Lambda), GCP, Terraform, Kubernetes (Helm, KIND, ArgoCD), Docker, GitHub Actions, Prometheus, Grafana, Loki, Alertmanager, OpenTelemetry, Svix

AI Integration: LLM-powered backend pipelines (Groq, Qwen, OpenAI), Retrieval-Augmented Generation (RAG), Kokoro TTS, Model Context Protocol (MCP) agent tooling

EDUCATION & CONTINUOUS LEARNING

B.Eng. Mechatronics Engineering Federal University of Abeokuta — Thesis: Drone Maize-Disease Detection & Prepaid Electricity Meter Settlement on Ethereum/ESP32

Topics in Mathematics with Applications in Finance MIT OpenCourseWare

Building Cloud Computing Solutions at Scale Duke University / Coursera