Farzan Saqib

Founding Engineer | Blockchain, Custody and Capital-Markets Infrastructure

Islamabad, Pakistan (open to remote) | farzansaqib.work@gmail.com | +92 325 5757658
linkedin.com/in/farzansaqib | github.com/EggsyOnCode | eggsyoncode.dev

PROFESSIONAL SUMMARY

Senior blockchain engineer and founding engineer at Lombard Systems, working where digital assets meet capital-markets infrastructure: collateral enforcement, custody and MPC wallet integrations, and tokenized-asset credit. I build the reference implementation of Delivery-versus-Lock (DvL), integrating regulated custodians and the Canton Network; before that I led post-quantum cryptography on an EVM-compatible Layer 1. Strong interest in privacy technology: zero-knowledge proofs, zkTLS, trusted execution environments (enclaves) and confidential computing. My own work covers liquidation design for tokenized stocks, cross-chain messaging, exchange engines, zero-knowledge protocols and modular blockchain clients.

WORK EXPERIENCE

Founding Engineer, **Lombard Systems**
Oct 2025 - Present | Remote | lombardsystems.com

Financing infrastructure for tokenized securities, without moving collateral, built on the Delivery-versus-Lock (DvL) standard.

- Built core blockchain infrastructure for the DvL reference implementation, supporting over $50M in notional loan volume across 5+ regulated financial institutions.
- Designed and implemented the Programmable Credit Receipt (PCR), the signed receipt that evidences each position, issued only after both the lock and the funding confirm; used in collateral arrangements with major exchanges, including Binance, where collateral stays off the exchange.
- Integrated regulated custodians Zodia Custody, Dfns and Copper, connecting their MPC wallets and vaults so collateral is locked where it sits, in the owner's own account.
- Led integrations across Ethereum, Avalanche, Solana and the Canton Network, the permissioned network backed by Goldman Sachs, J.P. Morgan and other institutions.
- Built real-time collateral mark-to-market monitoring on Coin Metrics data at sub-100 ms latency.
- Designed an event-driven integration layer for securities lending workflows: FIX 4.4, Kafka streaming, microservices and Sentry.
- Developed confidential smart contracts on the Oasis Sapphire runtime for privacy-preserving deployments in regulated environments.

Core Contributor, MetaOrcha, **Solvent Labs**
Jan 2026 - May 2026

- Core contributor to MetaOrcha (github.com/solvent-labsorg/metaorcha), an open-source runtime that combines AI agents built on different protocols (MCP, A2A, computer-use) into a single verified workflow.
- Designed the emerge.yaml agent manifest: a versioned, schema-validated contract for each agent's identity, protocols, capability scopes and authentication, governed through an RFC process.
- Built the protocol dispatch router (pluggable handlers, no hard-coded backends) and a 7-step execution pipeline with per-step verdicts and a JSON evidence package per run.
- Built the planning service (vector search plus an LLM planner producing a multi-agent execution graph) and published orcha-sdk, a Python agent SDK.

Blockchain Architect, **ArmChain (ArmUp)**
Mar 2025 - Dec 2025 | Melbourne, Australia (remote) | armchain.org

- Led development of ArmChain, a quantum-resistant EVM-compatible blockchain in Go; replaced Ethereum's native signatures with ML-DSA in the execution client (working prototype).
- Integrated liboqs, Cloudflare CIRCL and TQ42, and designed validator identity and signing modules on NIST post-quantum schemes (ML-DSA, Falcon, Kyber).
- Researched threshold signing for custody-grade keys: threshold Dilithium and GG18/GG20 threshold ECDSA, with hands-on work on the codebases.

- Contributed to consensus and infrastructure for Lachesis-based DAG protocols (Fantom, Sonic) in go-opera; used Prysm for validator duties and slashing simulations; built validator tooling and telemetry.

CTO, Blockchain Technologies (side venture), **PQC-Labs**
Jun 2025 - Dec 2025 | Self-employed | pqc-labs.com

- Delivered bespoke, security-first blockchain platforms for three clients, including a quantum-resistant EVM-compatible ecosystem and a permissioned RWA network with on-chain identity management.

Blockchain Engineer, **Vanar Chain**
May 2024 - Feb 2025 | Lahore, Pakistan | onvanar.com

- Maintained chain infrastructure (Geth, Prysm, Kubernetes, Grafana, Prometheus), developed and audited Solidity smart contracts (Foundry, Hardhat), and led technical onboarding of new dApps.

Blockchain Developer, **Freelance**
Jun 2023 - Aug 2023 | Remote

- Designed, deployed, tested and integrated smart contracts for client Web3 applications.

PROJECTS

liqfacility | github.com/EggsyOnCode/liqfacility
Liquidation facility for tokenized-stock loans while the stock market is closed: committed buyers, fills that restore loan health, and reopening call auctions. Benchmarked on 35 real market closures against five lending strategies; about 60% lower liquidation cost for borrowers. **Technologies:** Go, Solidity, Foundry, Morpho Blue.

Eavesdrop | github.com/EggsyOnCode/eavesdrop
Cross-chain interoperability protocol inspired by Chainlink CCIP and Off-Chain Reporting (OCR). **Technologies:** Go, libp2p, Anchor, Foundry.

Velho Exchange | github.com/EggsyOnCode/velhoexchange
Centralized exchange (CEX) with order books, a matching engine and a spot-trading API, running on live ETH accounts. **Technologies:** Go, Next.js.

Anomi | github.com/EggsyOnCode/anomi
Privacy-centric stablecoin on- and off-ramp built on zero-knowledge proofs (ZKPs), zkTLS and order books. **Technologies:** Go, Noir, Solidity, Rust, Anchor, React Native.

Xenolith | github.com/EggsyOnCode/xenolith
Modular blockchain built around an adapter and plugin architecture, with multi-threaded p2p networking. **Technologies:** Go, gRPC, libp2p.

Alacrity | eggsyoncode.github.io/projects/2_project/
RWA tokenization protocol: microservices platform for peer-to-peer commerce without advance payments. **Technologies:** NestJS, Foundry, PostgreSQL, AWS EKS, Vue.

Distributed CAS | github.com/EggsyOnCode/CAS
Peer-to-peer content-addressed storage with large-file streaming and encrypted remote storage. **Technologies:** Go.

SKILLS

Capital markets and tokenization: collateral and liquidation design, securities lending, FIX 4.4, tokenized RWAs, ERC-3643, lending/borrowing, liquidity pools, staking, programmable credit, market microstructure, Canton Network

Custody, cryptography and privacy: MPC wallet integrations (Zodia Custody, Dfns, Copper), threshold signatures (GG18/GG20, threshold Dilithium), post-quantum cryptography (ML-DSA, Dilithium, Falcon, Kyber), zero-knowledge proofs (ZKPs, Noir), zkTLS, trusted execution environments and confidential computing (Oasis Sapphire)

Blockchain and smart contracts: Solidity, Rust, Foundry, Hardhat, Ethers.js, Chainlink, LayerZero, The Graph, ERC-20, ERC-721, ERC-3643, Geth, Prysm, Anchor; Ethereum, Base, Solana, Avalanche, Arbitrum, zkSync, Polygon CDK, Hyperledger, Oasis Sapphire, Canton

Backend and distributed systems: Go (Golang), Rust, TypeScript, Python, microservices, event-driven architecture, gRPC, libp2p, Kafka, RabbitMQ, PostgreSQL, MongoDB, Prisma, TypeORM, NestJS, Node.js, GraphQL, React, Vue

DevOps and cloud: Docker, Kubernetes, Terraform, GitHub Actions, AWS, GCP, DigitalOcean, Sentry

AI and agents: Python, LangGraph, FastAPI, MCP, A2A, multi-agent orchestration, agent harnesses