

SAMUEL ALAKE

✉ g4titan1@gmail.com  [GitHub](#)  [Personal Website](#)

EDUCATION

Federal University of Technology, Akure
B.Sc. in Industrial Mathematics, CGPA: 4.19/5.00

Ondo State, Nigeria
Jan. 2020 – Nov. 2025

TECHNICAL INTERESTS

Programming language design and theory, operating systems, compiler design, cryptography, low level virtual machines, distributed systems, robotics, and blockchain: Ethereum.

ENGINEERING EXPERIENCE

Mirage Technologies AG
Founding Engineer — Remote

Zug, Switzerland
Jun. 2025 – Aug. 2026

Azoth

- Designed and built Azoth in Rust as a deterministic EVM bytecode transformation system.
- Built bytecode decoding, control flow analysis, CFG based passes, seeded rewriting, jump and deployment validation, fuzzing, and REVM end to end verification.
- Implemented dispatcher rewriting, constant and string masking, PUSH splitting, storage slot shuffling, and control flow reordering while preserving contract behavior.

API, Pricing & Execution Infrastructure

- Led backend development for obfuscation, compliance, pricing, attestation, escrow generation, and transaction execution.
- Designed private pricing for per chain mixed assets, signed time bound quotes, blinded batch signers, single transfer escrows, and reward funded gas advances.
- Implemented REVM gas simulation with Ethereum Merkle proofs, gas price estimation, fee history, deployment validation, and threshold based compliance screening.
- Built quote, authorization, execution, settlement, validation, and failure handling systems for complex onchain workflows.

Mirage's Escrow Contract

- Developed most of Mirage's Solidity escrow contracts and onchain proof verification infrastructure.
- Implemented Ethereum Merkle Patricia trie proofs, RLP and block header parsing, transaction and receipt inclusion proofs, ERC-20 event validation, and native transfer validation.
- Delivered EIP-712 executor authorization and single, ERC-20, native, and batch settlement flows with reservation expiry, partial completion, and value weighted rewards.

Nomad

- Contributed across Nomad's enclave, node, Ethereum, P2P, API, and shared type crates.
- Implemented mixed asset batch execution, Uniswap settlement swaps, CoW Protocol inventory funding, signed pricing policies, receipt proof generation, EOA lifecycle management, and Tempo support.
- Built restart safe recovery, transaction safety checks, contract bindings, protocol types, end to end tests, and SGX attested operational metrics across pricing, liquidity, execution, verification, retries, and settlement.

Frontend

- Owned frontend engineering for Mirage’s primary application, building single and batch transfer flows, wallet handling, ENS resolution, token approvals, fee estimation, transaction history, CSV export, and PDF receipts.
- Resolved production issues involving balances, nonce prediction, responsive layouts, loading states, and crosschain execution.

Blink

- Designed and built Blink across backend and frontend, including a concurrent Rust indexer for Ethereum contract creation and bytecode data through batched RPC tracing.
- Built Parquet storage, DuckDB analytics and rollups, incremental imports, live tail processing, and opcode indexing.

Nomad Sidecar

- Designed the initial metrics sidecar architecture and implemented most of its first version and supporting protocol types.
- Implemented SGX attested metrics, P2P distribution and backfill, verification, history, exports, liquidity aggregation, source attribution, and REST/SQL APIs.

Fleek Labs Inc.

Rust Engineer — Remote

New York, USA
Aug. 2024 – May 2025

JOE — JavaScript Optimization Engine

- Helped develop JOE, an ahead of time JavaScript optimization engine using symbolic execution and static analysis to improve performance and reliability.
- Architected JIR, the intermediate representation, with core values and algorithms aligned to ECMAScript semantics.
- Engineered symbolic types for type inference and expression evaluation while controlling path explosion.
- Developed state reduction by merging parallel execution contexts and improving conditional control flow analysis.
- Contributed to minification, tree shaking, LLVM code generation, lazy loading, error detection, and runtime optimization.

Fleek Machines Management Server

- Implemented secure port forwarding and network parsing for Fleek’s LibVirt and QEMU virtual machine infrastructure.
- Built VM creation, control, and deletion APIs, with vCPU and memory tracking, authorization, and conflict handling.
- Hardened IPv4, IPv6, domain, and wildcard inputs and optimized port discovery with QEMU monitor commands.

SELECTED PROJECTS, OSS CONTRIBUTIONS & WRITING

Zink

Compiler Engineer & Project Maintainer

Worked on compiler infrastructure and low level systems engineering in Rust.

OSS
Feb. 2025 – Jun. 2025

rustc codegen gcc

Contributed to the GCC backend for the Rust compiler, focusing on core features, bug fixes, and optimizations.

OSS

Sparsity of the Ethereum Virtual Machine Context Vector and its Consequences for Obfuscation

Research on observable EVM execution context, sparse state spaces, and their implications for program analysis and bytecode obfuscation.