Nwele Uchenna David

developeruche@gmail.com

Systems Engineer - Rust, Ethereum Execution Layer, zkVMs

Open to: Remote (global), Relocation

github.com/developeruche

developeruche.com

linkedin.com/in/developeruche

Work Experience

- **Ethereum Foundation - zkEVM Team** — Remote
  Research Engineer (Contract) — *May - August 2026*
  - Proposed and drafted a REST/SSZ `newPayloadWithWitness` Engine API endpoint for Reth and Ethrex, motivated by EIP-8025 and real-time zkVM attestation; benchmarks showed order-of-magnitude witness-delivery latency reduction over the existing round-trip.
  - Implemented `zilkworm-stateless`, a C++ zkVM guest program for stateless Ethereum block validation, covering EEST spec compliance and Merkle-Patricia Trie changes.
  - Contributed to `ere-guests`, a collection of stateless block-validation guest programs for Ethereum blocks, and to `ere`, the Foundation's unified zkVM interface and toolkit.
  - Integrated a purpose-built TCP engine into Scroll's `reth-witness-indexer`.
- **Ethereum Foundation - zkEVM Team** — Remote, Buenos Aires, Argentina
  Research Engineer, Ethereum Protocol Fellowship — *June - Nov 2025*
  - Benchmarked precompiles across four major zkVM platforms (RISC0, SP1, OpenVM, Ziren), showing that optimized software with lean precompiles outperforms generic implementations with fat precompiles by roughly 27×; published the findings as *Fat vs. Lean Precompiles in zkVMs*, arguing for a precompile strategy that avoids ecosystem fragmentation.
  - Identified AOT compilation as the primary lever for 10× zkVM performance gains, reaching 36M gas in 0.5s on consumer hardware, after profiling trace-generation bottlenecks.
  - Designed and prototyped a hybrid Ethereum architecture replacing the EVM with a RISC-V execution engine while preserving backward compatibility, via magic-number detection and a mini-EVM interpreter running as a RISC-V program.
  - Integrated the hybrid VM into a production Reth node and built `cargo-hybrid`, a CLI toolchain for scaffolding, compiling and deploying RISC-V smart contracts; benchmarked end-to-end against `revm`.
  - Prototyped "Stateless Attestors" with the Prysm team, enabling consensus-layer nodes to verify execution payloads through zkEVM proofs without a full execution-layer dependency.
  - Built a stateless Ethereum block validator running inside `ziskemu`; compiled Geth bare-metal with the Tamago toolchain to eliminate Linux syscall dependencies on zkVM targets.
- **Independent Research - Rust Systems, zkVMs & ZKPs** — Lagos, Nigeria
  Research Engineer (self-directed) — *June 2024 - June 2025*
  - Stepped back from full-time engineering to work on cryptography and zero-knowledge full time - reading papers daily, running research calls with peers, and implementing each result in Rust rather than stopping at the write-up; converged on zkVMs as the specialism.
  - Built the polynomial layer the rest of the work rests on - univariate and multilinear representations with their evaluation, interpolation and arithmetic.
  - Implemented the sum-check protocol and GKR, then succinct GKR - GKR composed with a multilinear KZG commitment scheme.
  - Implemented Groth16, and KZG polynomial commitments in both univariate and multilinear form.
  - Implemented PLONK end to end - prover, verifier, circuit compiler, and a transpiler lowering GKR circuits to Plonkish arithmetization.
  - Published the whole body of work as `cryptography-n-zk-research`, an open repository with accompanying notes on computational proof systems.
- **MyAI - Me Protocol** — USA, Remote
  Senior Blockchain Engineer — *Nov 2022 - July 2025*
  - Led the blockchain engineering team delivering Me Protocol V3, V4 and V5 alongside the CTO, owning architecture through to mainnet deployment.
  - Architected the protocol in Solidity and Yul on a custom Diamond-standard design, with Foundry-based deployment and testing and an in-house migration toolkit supporting both upgrades and

rollbacks across a system of contracts.
  - Led development of a custom optimistic rollup in Rust settling to Polygon for asset transfer and L2-to-L1 bridging; omitting a sandboxed VM kept it substantially lighter than a general-purpose EVM rollup for the workload.

- **Polytope Labs - Hyperbridge** Remote
  Blockchain Engineer *Jan - April 2024*
  - Rust engineer on Tesseract, the Hyperbridge cross-chain relayer, and Substrate runtime engineer on the Nexus runtime deployed to Polkadot mainnet.
  - Built HyperClient, a Rust library compiled to WebAssembly that lets Hyperbridge clients manage in-flight ISMP requests; implemented teleport support for DOT addresses.

- **Scroll** Remote
  Smart Contract Engineer (Contract) *April - June 2024*
  - Built the on-chain contract layer for Scroll Pass, a zero-knowledge event ticketing platform delivered with the Ethereum Community Fund.

- **Nethermind** Remote
  Smart Contract Engineer (Intern) *Jan - April 2023*
  - **Valantra:** built a DEX combining a Limit Order Book with constant-product and constant-sum AMM algorithms over a shared liquidity base, avoiding fragmentation; exchange, pool and management logic fully modularized.
  - **Libre:** contributed to a decentralized stock-market protocol for on-chain IPOs.

- **AfriOne, Nahmii & Web3Bridge** Remote / Lagos, Nigeria
  Smart Contract Engineer, Design Architect & Team Lead (Contract) *2021 - 2023*
  - **AfriOne:** led the blockchain team; shipped a contract system on a private Geth network backing a consumer finance product, plus an SDK for the mobile and backend teams.
  - **Nahmii:** built Bondii, an asset-bonding system, and a cross-chain staking platform unifying assets across Ethereum L1 and Nahmii 3; ported the Uniswap contracts and UI to Nahmii 3.
  - **Web3Bridge:** smart contract team lead for an on-chain governance system built on the Diamond standard.

Open Source & Research

- **Ethereum core tooling in Rust - Alloy, REVM, OpenZeppelin Stylus** *Merged contributions*
  - Contributor to **Alloy** (the ground-up rewrite of ethers-rs), **REVM** (the Rust EVM used across the client ecosystem), and **OpenZeppelin's** Stylus contract library.
- **RAX - RISC-V to x86-64 AOT compiler (Sublinear Labs)** *Contributor*
  - Contributor to **RAX**, a RISC-V (RV64IMA) to x86-64 ahead-of-time compiler for ELF binaries with interpreter and JIT backends; contributed Ethereum block-execution guest programs, stateless block-execution handling, and the test/benchmark flow.
- **RISC-V EVM Experiment - research & Rust proof of concept** *2025*
  - Evaluated RISC-V as an alternative execution environment for smart contracts; published *Bridging Worlds: A Performance and Feasibility Analysis of RISC-V Integration with the Ethereum Virtual Machine*. Shipped a supporting RV32IM assembler and emulator in Rust.

Skills

- **Languages:** Rust, Solidity, C++, TypeScript, Go
- **Ethereum:** Reth, REVM, Alloy, Foundry, Engine API, SSZ, Merkle-Patricia Tries, EIP authorship
- **Zero-Knowledge:** zkVMs (RISC0, SP1, OpenVM, Ziren, ZisK), Groth16, PLONK, KZG, GKR, sum-check
- **Systems:** RISC-V (RV32IM), compilers, emulators, AOT compilation, performance benchmarking, WebAssembly
- **Distributed Systems:** Substrate / Polkadot SDK, optimistic rollups, cross-chain messaging (ISMP)

Education

- **University of Benin - B.Eng, Computer Engineering** *Benin City, Nigeria 2019 - 2024*
- **Polkadot Blockchain Academy** Singapore
  Blockchain Development - Rust, Substrate Runtime Engineering *May - June 2024*