Poulav Bhowmick

Protocol Engineer | Ethereum Consensus, P2P Networking, Distributed Systems | Rust

Kolkata, India bpoulav@gmail.com poulav.xyz linkedin.com/in/poulavb github.com/PoulavBhowmick03

Summary

Protocol engineer working on core blockchain infrastructure in Rust. Implemented memory-aware state caching in an Ethereum consensus client to prevent OOM during non-finality, shipped core preconfirmation functionality for a based rollup sidecar, and built shielded settlement infrastructure for agent-to-agent commerce. 60+ merged pull requests across 11 Rust protocol repositories. Comfortable across PBS, fork choice, current EIPs, SSZ and libp2p, with a bias toward performance and reliability.

Experience

Rust Dev Intern *May 2026 – June 2026*
BSV A

- Built the transaction building and signing modules for the BSV Rust SDK, cutting memory overhead and tightening the cryptographic execution paths.
- Worked across the SPV wallet and the BRC-100 standard within the BSV protocol stack.

Protocol Developer Intern *Jan 2026 – April 2026*
Nethermind

- Contributed to Catalyst, Taiko rollup's preconfirmation sidecar, implementing core protocol functionality in Rust
- Worked on Pluto, the Rust implementation of Obol's distributed validator Charon, improving reliability and coordination mechanisms
- Implemented DKG flows and authored Validator and MEV integration test suites to harden the client before mainnet rollout

Ethereum Protocol Fellow, Lighthouse *June 2025 – Nov 2025*
Ethereum Foundation

- Selected as 1 of 20 fellows from 1,200+ applicants (top ˜1%)
- Implemented memory-aware state cache replacing count-based LRU, preventing OOM during non-finality periods *(PR #7803)*
- Validated cache behaviour via tracing, criterion benchmarks, and property-based tests aligned with consensus data flow
- Introduced slot-aware dynamic delay for block reconstruction, adapting to varying network conditions across mainnet and testnets
- Hardened libp2p Identify and Ping behaviours and improved peer lifecycle logging; added reproductions for flaky tests
- Wrote attestation-handling benchmarks to surface hotspots in consensus-critical paths
- Contributed to SSZ serialization and consensus-critical code paths

OnlyDust Fellow *June 2024 – Aug 2025*

- Selected as 1 of 40 fellows from ˜500 participants
- Built Rust backends using Actix and SQLx with integration tests, Docker packaging, and Nginx load balancing

Grant Recipient, StarkFinder (Starknet) and MidoFinance (Solana) *2024*

- Awarded seed grant from the Starknet Foundation and accepted into the Argent Telegram Accelerator for StarkFinder
- Received Solana x CoinDCX InstaGrants via Superteam for MidoFinance

Backend Developer *Aug 2024 – Oct 2024*
Invisible Studios

- Reduced p95 API latency by ˜30% through query-plan optimization and targeted indexing
- Built and maintained production APIs with real-time data synchronization

Projects

Erebus, shielded settlement infrastructure for AI agents *GitHub*

- Built coordination and shielded settlement infrastructure for agent-to-agent commerce: two agents open an encrypted channel carried in privacy-pool note salts, negotiate over MCP, and settle atomically through the STRK20 shielded pool, making the accepted offer and the payment a single proven state transition with no agreed-but-unpaid gap.
- Designed the v3 wire format, encrypting authenticated deal records under AES-256-GCM-SIV.
- Implemented selective disclosure scoped to one deal and one named recipient with an explicit expiry.

Operating System in Rust *GitHub*

- Built a minimal OS kernel in Rust with bootloader setup, paging, and interrupt handling
- Implemented custom memory management routines and basic task scheduling for kernel-level concurrency
- Gained hands-on experience with low-level systems programming on x86_64 architecture

Gossimini *GitHub*

- Built a CLI pub/sub node validating topic mesh, backoff, and ack paths
- Simulated N-peer networks with chaos testing (drops and partitions) and exposed metrics for rebroadcast and fanout
- Deepened async Rust, threading, and libp2p concurrency expertise

Merkle Tree Implementation in Rust *GitHub*

- Implemented inclusion and consistency proofs with SSZ-style hashing
- Added property-based and fuzz tests to validate correctness under adversarial inputs
- Shipped a CLI with fixtures for quick verification

Open Source Contributions

60+ merged pull requests across 11 Rust protocol repositories.

- **Stellar Private Payments** (Nethermind): implemented the native disclosure prover and shipped the spp disclosure generate/verify commands (#479) and spp notes (#524); fixed undefined behaviour found under Miri (#262, #297, #309)
- **Lighthouse** (Ethereum consensus client): made the max column reconstruction delay a function of slot time instead of a fixed 1s bound (#8067); added SSZ responses to HTTP API endpoints (#8907); fixed custody backfill sync reporting wrong time estimates (#8291)
- **Reth** (Paradigm, Ethereum execution client): 18 merged PRs across networking, trie, and node plumbing; made MultiConsumerLruCache track in-memory usage rather than entry count (#14034); inverted provider access in the sparse trie (#14662)
- **Kona** (OP Stack in Rust): replaced unbounded channels in the kona-service actors with bounded ones (#2045); propagated errors raised during NodeActor::start instead of swallowing them (#2322); removed a panic in the RPC actor (#1709)
- **rust-libp2p**: fixed a stream leak by garbage-collecting deregistered streams in Shared::accept (#5999); removed async-std from the uds transport (#6060)
- Also merged in revm, alloy, anchor (SSV), helix (MEV-boost relay), and pluto (Obol DVT)

Technologies

Languages: Rust, Solidity, C, C++, Cairo

Systems and Infrastructure: libp2p, Git, Docker, Kubernetes, Redis, Kafka, PostgreSQL, AWS

Education

Heritage Institute of Technology *Sept 2022 – June 2026*

B.Tech in Electronics and Communication Engineering

CGPA: 8.3 / 10