Perry's Résumé - Blockchain Engineering

Perry Kundert <*perry@kundert.ca*> +1-780-970-8148

2026-05-30 12:00:00

R&D engineer with 30+ years building secure, high-performance distributed systems; now focused full-time on blockchain infrastructure. Designed a complete wealth-backed monetary system on Ethereum EVM: ERC-20 tokens with identity-aware transfers, ERC-721 insured-asset NFTs, parametric insurance smart contracts, and a privacy-preserving bearer-note layer built on Groth16 SNARKs over Poseidon Merkle commitment pools. The work spans smart contract architecture in Solidity, cryptographic protocol composition (ElGamal re-encryption, Chaum-Pedersen equality proofs, Pointcheval-Sanders credential rerandomization), zk-circuit design, formal correctness verification, and economic mechanism design; all implemented and verified end-to-end on BN254. Comfortable reading specs, debugging hard problems, and learning new corners of the stack. (PDF, Text)

Technical Skills

- **Smart Contracts & EVM**: Solidity (ERC-20, ERC-721, custom), Ethereum, Polygon, Hardhat, Foundry, OpenZeppelin, DeFi integration (Uniswap AMM pools, Chainlink oracles)
- **Languages**: Rust (5+yrs), C++ (25+yrs), Python (15+yrs), C, Go, Solidity
- **Zero-Knowledge & Cryptography**: Groth16 SNARKs, circom/snarkjs, Poseidon hashes, Merkle trees, nullifier sets, ElGamal encryption, Chaum-Pedersen equality proofs, Schnorr signatures, Pointcheval-Sanders credentials, Fiat-Shamir NIZK transforms; implemented, composed, and formally verified on BN254
- **Distributed Systems & Consensus**: CAP/BFT systems, partition-tolerant protocol design, DHTs, agent-centric accounting, consensus without global coordination
- **Networking & Concurrency**: JSON-RPC, gossip protocols, mTLS 1.3, WebSockets, WebTransport, QUIC, lock-free wait-free concurrency, real-time systems
- **Performance**: SIMD (Intel AVX, ARM NEON), SDR signal processing, hot-path profiling, throughput optimization
- **DevOps & Infra**: Linux, Docker, CI/CD, Git, deterministic builds (Nix), GDB, Valgrind, Linux Perf

Experience

R&D Consultant, *Dominion R&D Corp.* (Remote, 2009–Present)

Architected a complete blockchain-based monetary system; smart contracts, cryptographic identity, zero-knowledge privacy primitives, and economic mechanism design. Guided teams through cryptographic and distributed systems architecture.

- Designed the **Alberta Buck** protocol, a full Ethereum EVM smart contract system comprising:
  - **BUCK** (ERC-20): Identity-aware fungible token with continuous demurrage accounting, on-chain credit-limit enforcement against pledged assets, and Chaum-Pedersen NIZK re-encryption proofs gating every counterparty-pair transfer.
  - **BUCK_CREDIT** (ERC-721): Insured real-world-asset NFTs with deterministic depreciation curves and parametric insurance integration; the collateral backing the ERC-20 supply.
  - **IdentityRegistry**: On-chain PS-signature credential registry with ElGamal-encrypted identity points, verified via BN254 precompile NIZK at registration; the trust anchor every transfer is gated on.
  - **Notes**: Privacy-preserving bearer-instrument layer. Tornado-style incremental Poseidon Merkle commitment pool with Groth16 SNARK batch-mint and per-flavor spend circuits. Three flavors (A1 addressed cheque, A2 private cheque, B1 bearer cheque) collapse to two circuit shapes covering every deferred-approve use case: self-wormholes, signed cheques, printable paper notes indistinguishable from cash.
  - **BUCK_K**: PID-controlled algorithmic issuance stabilizer referencing a commodity-basket price oracle; automated monetary policy without central bank intervention.
- Composed and verified the full cryptographic stack with executable `py_ecc` sanity checks on BN254 (formal verification by qualified mathematicians and cryptographers remains to be done):
  - Pointcheval-Sanders signature rerandomization for unlinkable credentials.
  - Chaum-Pedersen re-encryption equality proofs for bilateral identity disclosure; proved completeness, special soundness, and honest-verifier zero-knowledge, lifting to NIZK via Fiat-Shamir in the random oracle model.
  - A-spend and B-spend SNARK soundness, confirming that only the named recipient can deposit an addressed note and that bearer-cheque deposit always identifies the depositor.
  - Domain-separated Poseidon PRF nullifiers with double-spend prevention across both note flavors.
  - Mutual-decryptability invariant preservation; every note spend yields bilateral identity disclosure between counterparties.
- Designed economic mechanisms for MEV-resistant DeFi integration:
  - Analyzed how PID-controlled issuance (with proportional and derivative terms) automatically counteracts whale manipulation of BUCK/stablecoin AMM pools; the system transfers the attacker's assets to legitimate credit holders.
  - Modeled the effect of front-running-resistant oracle design: multiple independent controller implementations (PID, Kalman-filtered PID, MPC) with median-value aggregation prevent any single oracle compromise from moving the issuance multiplier.
  - Designed parametric default insurance with risk-proportional premium investment in a mutual insurance pool; algorithmic premium setting eliminates underwriter discretion.
- Built Python simulation and modeling infrastructure:
  - Full economic simulations of mortgage vs. BUCK issuance across 25-year horizons.
  - Seigniorage extraction analysis: proved commercial banks earn 2× the profit of a private lender on identical loans, solely from the regulatory privilege of ex-nihilo deposit creation.
  - Commodity-basket stability analysis (BCPI + labour) demonstrating BUCK would have been more stable than CAD$ over the last 50 years.

  - Statistical PID controller tuning for the `BUCK_K` Value Stabilization Factor.

- Developed widely-used open-source cryptographic libraries:
  - `python-slip39`: Cross-platform Shamir secret sharing for BIP-39 seed backup and recovery.
  - `ezpwd-reed-solomon`: High-throughput C++/JS Reed-Solomon FEC, used by aerospace and defense systems globally.
  - `cpppo`: Industrial EtherNet/IP protocol implementation in Python.
- Alberta Buck Architecture · BUCK Notes · Formal Proofs · Identity System
- Ethereum Implementation · Slide Deck · python-slip39 · ezpwd-reed-solomon

Distributed Systems R&D, *Holo Ltd.* (Kelowna, BC 2018–2020)

Architected and tested prototypes of HoloFuel's novel partition-tolerant transaction engine in Rust.

- Implemented agent-centric accounting enabling linear O(n) scaling vs. $O(n^2)$ blockchain bottlenecks; each agent held its own signed transaction chain, with bilateral settlement during network partitions.
- Validated simultaneous atomic transactions across partitioned networks; demonstrated that mutual-credit systems can settle without global consensus.
- Developed statistical models and PID control systems for economic stabilization across distributed agents.
- Contributed to R&D for Holochain's distributed application framework.

Software Engineer, *clearGRID Ltd.* (Kelowna, BC 2017–2022)

Built fault-tolerant distributed telemetry consensus without synchronized state machines.

- Architected real-time SDR system processing 25 Msps I/Q signals on commodity hardware; consensus on meter readings across lossy RF channels without global coordination.
- Optimized AVX/NEON SIMD vector processing for multi-channel demodulation at the network edge.
- Implemented fault-tolerant distributed consensus for partial data collected from multiple samples across unreliable channels.

Sr. IT Advisor, *Enbridge Pipelines* (Edmonton, AB 2002–2009)

Deployed partition-resilient SCADA maintaining safety-critical consensus during network splits.

- Architected Reed-Solomon erasure-coded multi-route protocol enabling real-time SCADA operation through crippling communications failures; a BFT-style availability guarantee implemented a decade before blockchain made the terminology common.
- Implemented multi-path routing maintaining cryptographic integrity across hostile network environments.
- Developed solid-state RTU firmware for 24/7 pipeline control with $<$1 defect per 10,000 LOC.
- Secured communications for \$100B+ critical energy infrastructure.

Software Developer, *Hewlett-Packard* (Calgary, AB 1989–1996)

Pioneered state-machine consensus for distributed industrial control systems.

- Re-engineered RTAP's core SCADA alarm system with DFA state machines for geographically distributed safety-critical workflows; deterministic state transitions without central coordination, deployed continent-wide and still operational after 30+ years.

Open Source & Publications

- python-slip39: Cross-platform SLIP-39 Shamir secret sharing implementation (Python)
- ezpwd-reed-solomon: High-performance Reed-Solomon FEC (C++, JavaScript)
- cpppo: Industrial EtherNet/IP and Modbus protocol implementation (Python)
- holofuel-model: Distributed mutual-credit economic simulation
- Authored series of technical papers on blockchain architecture, zero-knowledge identity systems, cryptographic protocol composition and formal verification, consensus algorithms, and monetary system design; perry.kundert.ca/range/finance/

Education

B.Sc. Computer Science, *University of Calgary* (Calgary, AB 1984–1989)