

# Niek Kamer

Perf engineering on zk proof systems · Plonky3 maintainer

Amsterdam, NL · [info@axobello.com](mailto:info@axobello.com) · [github.com/Niek-Kamer](https://github.com/Niek-Kamer) · [niek-kamer.github.io](https://niek-kamer.github.io)

Available for zk perf engineering contracts and senior IC roles.

## Selected open-source work

**Plonky3**, maintainer (May 2026 – present)

[Plonky3/Plonky3](#)

- [PR #1623](#) (merged): `add_canonical_asm` skips redundant canonicalize on provably-canonical sites. -2.93% end-to-end Goldilocks Poseidon2 prove time on aarch64 NEON; -5.91% / -6.21% on external permute kernels.
- [Issue #1642](#) (audit) → [PR #1645](#) by @Nashtare: packed AVX-2 Poseidon1 running 0.77× scalar due to missing Karatsuba back-port. 1.20× wall-time speedup post-fix.
- [PR #1656](#) (in review): packed-NEON Poseidon2 dispatch routed through generic. -16.4% Merkle wall time on Pi 5 Cortex-A76 across log<sub>2</sub> leaves ∈ {14, 16, 18}.
- [PR #1667](#) (in review): cross-permute batched Poseidon2 for widths 8 and 12. -27.2% Zen 5 AVX-2, -31.5% Zen 5 AVX-512, -10.82% Pi 5 NEON (per-permute, width 8).

**zkmcu**, author

[Niek-Kamer/zkmcu](#)

- `no_std` Rust Groth16/BN254 verifier for microcontrollers. Cortex-M33 and RISC-V Hazard3 targets. ~1 s verify on a \$7 RP2350.
- Pairing arithmetic, optimized BN254 G1/G2 ops, fits within RP2350 RAM budget.

**poseidon2-harness**, author

[Niek-Kamer/poseidon2-harness](#)

- Dual-tree differential testing + PMU-cycle benches for Plonky3 Poseidon2 on aarch64 NEON.
- Pulls two Plonky3 commits side-by-side via cargo git deps. Bitwise + canonical oracles across both trees, per-layer + full-permute coverage, widths 8 / 12 / 16 / 20.
- Used during a Plonky3 Poseidon2 audit; turned up two correctness-adjacent findings.

**BNO055 driver fork**, maintainer

[Niek-Kamer/BNO055](#)

- `no_std` Rust driver for the Bosch BNO055 9-axis IMU, embedded-hal 1.0.
- Removed two unsafe blocks in calibration serialization. Fixed a real bug: `AxisRemap::y()` returned the X axis, hidden behind a clippy allow.
- 27-36% per-read latency reduction (page-tracking + bulk-read API). Full sensor loop 9.83 ms → 6.28 ms on Pi 5. Added 206 integration tests where the upstream had zero.

## Selected writing

All at [niek-kamer.github.io/writing](https://niek-kamer.github.io/writing). Methodology and measurement write-ups documenting the contributions above.

- *The trait-method boundary that cost Plonky3 Poseidon2 27%* May 2026
- *The AVX-512 kernel that was 1.79× slower than the compiler's* May 2026
- *The Poseidon2 regression my microbenchmark told me wasn't there* May 2026
- *The SIMD path that was 0.77× scalar* May 2026
- *The two instructions hiding in every Goldilocks Poseidon2 add* May 2026

## Skills

**Languages.** Rust (primary). TypeScript, C, Solidity (reading).

**Architectures.** aarch64 NEON (Cortex-A76, Apple Silicon code-path via shared NEON kernels), x86 AVX-2 / AVX-512 (Zen 5), Cortex-M33, RISC-V Hazard3.

**Performance tooling.** `perf_event_open` with grouped counters, `perf annotate` / `perf stat`, `criterion`, `cargo bench`, `objdump`, GDB. Comfortable reading and writing inline NEON / x86 assembly.

**Domains.** Zero-knowledge cryptography (Groth16, STARKs, FRI, Poseidon1/2 / Goldilocks / Babybear / BN254), `no_std` embedded Rust, perf-counter analysis on superscalar OoO cores.

## Experience

**Founder, Axobello** (2025 – present)

Saint Martin / Amsterdam

Independent software work. Current focus: zero-knowledge proof system performance engineering, embedded ZK verification.

## Education

MBO (Netherlands vocational secondary education). Self-taught in systems programming, cryptography, and zero-knowledge proof systems.