Zain Dana Harper

Systems Engineer

Seattle, Washington | Open to remote, hybrid, onsite, contract, and project work zaindharper@gmail.com | https://harperz9.github.io | https://github.com/HarperZ9

SUMMARY

Systems engineer whose current work is verification infrastructure for AI agents: accountability engines, capability-typed tooling, and re-checkable evidence surfaces. Public work also spans compilers and language tooling, real-time graphics and color science, technical documentation, and operations. Builds inspectable tools with explicit boundaries, tests, provenance, and reproducible release paths. Independent engineering practice since 2023, grounded in earlier technical support, freelance technical writing, compliance documentation, and eleven years of field operations leadership.

SELECTED ENGINEERING WORK

Flywheel accountability engine for AI agents | Python (stdlib-only engine), Flutter

- Built an engine in which every agent tool call passes a default-deny capability gate and emits a sealed, hash-chained, offline-verifiable receipt binding what was allowed to what actually happened (witnessed argument and output digests, never raw content).
- Receipts compose into a transitive witness graph: a drifted action degrades exactly its downstream dependents, and a third party can re-verify the whole chain offline.
- Shipped an infrastructure control package (network egress receipts, credential scanning, isolation acceptance tests, a dual-confirmation kill switch, cross-layer event correlation) and published a dated assessment mapping it against the failure classes of the July 2026 agentic-security incidents.
- Merged to the default branch as of 2026-08-03: capability-typed shell admission with Unicode-spoof neutralization, a domain oracle registry routing claims to independent verifiers across three live domains (pytest for code, the Lean kernel for mathematics, and a measurement gate for empirical claims) with an honest UNVERIFIABLE verdict elsewhere, a model-neutral router with quota failover across providers, a subscription-auth adapter that consumes an authorized token rather than minting one, and session tooling to browse, resume, and export past verified runs.
- Completed the engine's first preregistered confirmatory run (2026-08-04): zero verdict disagreements across 2,646 certificate bodies in nine model contexts, statistically significant held-out selection uplift at every model size on the solvable task family, honest nulls kept, and the whole analysis anchored in a signed, offline-verifiable ledger.

BuildLang compiler and language tooling | Rust, C, HLSL, VS Code

- Built a systems language where ambient access is part of a function's type: typed capability effects (filesystem, network, foreign calls) that a callback or closure cannot silently launder, with Hindley-Milner inference and experimental linear types.
- Ships as the `buildc` toolchain (build, run, test, repl, fmt, pkg, LSP) with a production C backend, HLSL and GLSL shader output, and re-checkable build receipts: `buildc check --receipt` seals what a build observed and `buildc receipt verify` re-derives it later.

Project Telos and model-era engineering tools | Python, TypeScript, Node.js

- Built a public ecosystem for repository intelligence, evidence intake, multi-agent orchestration, evaluation, replayable records, and claim verification.
- Shipped focused tools including Index 2.9.0 for repository maps and context packs, Gather 1.6.1 for provenance-aware research intake, Forum 1.13.0 for model-agnostic orchestration, and Crucible 1.2.0 for worker/verifier evaluation workflows.

Real-time graphics, color, and display tooling | C++, HLSL, Python

- Released a Skyrim real-time graphics project whose current public career materials report more than 900,000 downloads.
- Built D3D11/HLSL post-processing and rendering systems involving tone mapping, TAA, SSR, SSGI, GTAO, volumetrics, and read-only shared-memory IPC.

OPEN-SOURCE CONTRIBUTIONS

- 22 code, test, and documentation pull requests merged into 19 public repositories maintained by others, including Datasette, tomlkit, pydantic-ai, DeepEval, pydash, and grimp.
- 11 further pull requests open and awaiting maintainer review across 11 repositories, including the Model Context Protocol Python SDK, Drizzle ORM, Datasette, and LLM. Open is not accepted.

SPEAKING

- "Pick the Lock for Everyone: Building Verifiable AI Workflows in Python," invited talk, Puget Sound Programming Python (PuPPy), scheduled 2026-08-19.

EXPERIENCE

Independent Systems Engineer / Founder-Builder | Zentropy Labs (Project Telos, HarperZ9) Seattle / remote | 2023-present

- Own architecture, implementation, integration, documentation, testing, packaging, public demos, and release evidence across a multi-repository systems portfolio.
- Coordinate agentic development workflows while preserving first-party review, verifier separation, testable contracts, and public claim discipline.

Freelance Technical Writer / Documentation and Product Operations Remote | 2017-present

- Produce API and implementation guides, security and compliance documentation, proposals, release notes, and support material.
- Work with NIST 800-171, CMMC readiness, SOC 2, ISO 27001, DFARS, incident response, and audit-support concepts in a technical-writing capacity.

Operations and Commercial Arboriculture Lead | Family business Seattle area | 2015-present

- Operated tree crews from the ground for eleven years: ran the rigging systems, judged clearances and how far limbs would swing relative to structures and people, and served as the second set of eyes for the person in the air.
- Led client intake, estimates, site assessment, scheduling, safety judgment, and crew and vendor coordination.

Technical Networking Support, Xbox Division | Microsoft Redmond, Washington | 2013-2014

- Diagnosed TCP/IP, DNS, NAT, firewall, router, and account-adjacent console networking issues across phone and chat support.

TECHNICAL STRENGTHS

Languages: Python, Rust, C++, TypeScript/JavaScript, Lua, HLSL, C#, PowerShell, Bash **Verification and AI:** capability gates, sealed receipts, offline re-verification, provenance, replayable ledgers, model routing, tool-use loops, MCP surfaces, worker/verifier separation, evaluation **Systems:** compilers, type systems, typed effects, code generation, D3D11, shader pipelines, shared-memory IPC, CMake **Delivery:** Git and GitHub, pytest, GitHub Actions, release notes, developer documentation, Linux

PUBLIC PROOF

Portfolio: https://harperz9.github.io/portfolio.html Repositories and releases: https://github.com/HarperZ9