Alexander Sorrell

AI Principal / Builder / Breaker — Verification & Evaluation · (817) 996-6123 · alexander.sorrell.it@gmail.com

linkedin.com/in/alexander-sorrell-it · github.com/Alexander-Sorrell-IT

Colleyville, TX (Dallas–Fort Worth) — open to remote, onsite, or relocation

github.com/Alexander-Sorrell-IT/starreckon
npx starreckon

VERIFIED AUDIT · 30.1 MASTERWORK

PROFESSIONAL SUMMARY

Models propose. A deterministic function owns the verdict. Every decision ships with a receipt a stranger can re-check — a rule ID, a cryptographic hash chain, or the OS kernel itself.

PROOF OF WORK

- **I broke my own benchmark.** It reported 100%. Its oracles only tested the features each rebuild had already implemented. I rewrote them to grade the full original surface across the same 72 apps — 52.3%, zero fully passing. I kept both numbers on the record and retracted the headline.
- **Calibrated abstention under attack.** I ran live prompt injection against my own forensics court to find out what it does when someone is actively trying to poison the judge. F1 halved. Planted false positives stayed at zero, and every degraded run abstained. It fails silent, not wrong.
- **Two counters, two authors, one answer.** My token counter (starforge) and an independently written one by a different author — different language, no shared code — read the same 20,217 transcripts from 5 computers, each in its own container with the corpus read-only. They agree exactly on sessions, active days, hours, input and cache tokens, and differ by 0.07% on output. The designed failure mode is INCONCLUSIVE, never PASS.
- **The kernel certifies the boundary, not the code.** A socket probe executes inside the sandbox and outside it; the OS returns EPERM rather than relying on an application-level boolean. Reproducible at your desk — npx starreckon prove prints the exact command.
- **Byte-for-byte ground truth.** Clean-room reconstruction of an NVIDIA HGX factory tool's output to an identical md5 against the closed-source Windows original, non-circular by control. Replaying 10 real H100 boards through the rebuilt suite matches the factory's disposition 10/10, and it caught two boards shipped as BBX Pass while carrying 8 row-remaps.
- **The payout function cannot receive an AI verdict.** Re-derives policy deterministically server-side so nothing upstream can trick it into an unauthorized state. 8 of 8 scripted adversarial attacks held, zero violations.
- **Measured, not remembered.** 33 public repositories since May 2026: 156,754 lines, 3,168 named tests, 939 commits on 91 active days — counted by my own repo auditor, which strips vendored libraries, other people's code and duplicate copies before it counts anything.

ARTIFICIAL INTELLIGENCE & TECHNICAL SKILLS

AI / ML & Eval Science	LLM-as-judge pipelines; golden-set design & contamination control; self-preference bias calibration; principled abstention; adversarial evaluation & red-teaming; multi-agent orchestration; deterministic QA; MCP
Languages	Python, TypeScript / JavaScript, Rust (→ WASM), Solidity, Bash, C
Platforms & Systems	Splunk / SPL, Foundry, Docker, GitHub Actions CI, Linux namespaces (unshare) / sandboxing, PyPI & npm publishing, MCP servers, Next.js / React, Playwright, REST APIs, SDK design
Infrastructure & Security	NVIDIA DGX/HGX A100/H100 bring-up, diagnostics & firmware (retimer, NVSwitch, FPGA registers), Linux, macOS sandbox confinement, smart-contract & web vulnerability analysis, DFIR

PROFESSIONAL EXPERIENCE

Founder & Principal AI Systems Architect — SignalCore Inc (DE) & Bastion Fold LLC (NM) · Remote · 2025 – Present

Verification infrastructure for AI systems: deterministic evaluation harnesses, agent guardrails, and audit trails where the component that decides is physically separate from the component that acts. Multi-agent orchestration, LLM-as-judge pipelines, MCP servers written from scratch, adversarial red-teaming. SignalCore Inc (DE, Jul 2025) builds agent governance; Bastion Fold LLC (NM, Apr 2025) operates deterministic security verification. Shipped 11 packages and 33 public repositories of my own on github.com/Alexander-Sorrell-IT (939 commits on 91 active days); won the T3 ADK Bounty Challenge Launch Edition with On the Record; one PR merged upstream at KeeperHub. Python, TypeScript, Rust, Solidity.

Frontier-Model Red Team & Evaluation — Handshake AI (Project Vox), Mercor (Les Artistes) & Alignerr · Remote · 2026 contracts

Adversarial prompt red-teaming against frontier models, rubric-based response assessment, reference-answer authoring, and multi-turn reasoning data curation used to harden model alignment and truthfulness. Authored paid ML-environment tasks for Alignerr's ML Envs pipeline.

Server Repair Technician L1 → L2 — OnTrack Staffing, then SMS Infocomm · Jan 2024 – Apr 2025

Ran the full factory sort flow on NVIDIA HGX/DGX A100 and H100 AI-compute systems — system check, BBX, functional stress, in-system run-in — isolating GPU-vs-baseboard faults with golden-part swaps and flashing re-timer and NVSwitch firmware; trained incoming technicians. A sort line is a machine for refusing to believe hardware is good until an independent test agrees. That is the exact discipline I build into evaluation software.

Junior Claims Evaluator — Claimplus · Apr 2021 – Jan 2024

Reviewed and validated property-insurance claims data; structured evaluation workflows and deterministic policy checks.

BY THE NUMBERS (MEASURED ACROSS EVERY PUBLIC REPOSITORY)

- **33 public repositories** of my own on github.com/Alexander-Sorrell-IT: 156,754 lines, 3,168 named tests, 939 commits on 91 active days, 2026-05-20 → 2026-10-31.
- **By language:** Python 80,817, JavaScript 53,740, TypeScript 10,078, Shell 4,285, HTML 3,280, Solidity 3,167, Rust 866, CSS 370 lines.
- *How it is counted:* bundled libraries, other people's code, generated files and duplicate copies are excluded (77,343 lines left out); a file in more than one repo counts once; commits count only mine and my agents'.

PROJECTS — EVERY REPOSITORY, WRITTEN FROM ITS DOSSIER

Every system separates the component that proposes from the component that decides. Every public repository of mine with code in it, on github.com/Alexander-Sorrell-IT (26 written up, 7 small ones listed at the end). Descriptions are each repo's own words or verified write-ups; the Code line is what its biggest source files say they do and the grade its executed checks gave it; lines, tests and commits are measured by repo-dossier, with vendored, generated and duplicate code excluded.

AI EVALUATION, VERIFICATION & RED-TEAMING

starreckon — Local-only audit engine & developer Wrapped for AI-coding sessions. No-egress guarantee enforced by OS sandbox confinement rather than asserted in code — the kernel refuses the socket (EPERM); designed failure mode is INCONCLUSIVE, not PASS. 1,206 of 1,207 tests passing, zero dependencies, cryptographic allowlist pins. 2,820 npm downloads (npx starreckon). JavaScript

JavaScript 50,537 · HTML 1,748 · Python 436 · Shell 218 lines · 1,206 of 1,207 tests passing (run) · 228 commits on 29 days (2026-08-06 → 2026-10-30) · NPM starreckon 2,820 downloads · NPM starforge-cli 4,878 downloads · github.com/Alexander-Sorrell-IT/starreckon

Code: code quality B (83/100, from checks that ran) · scan.mjs: Metadata-first scanners for local AI-coding session logs · readers.mjs: Readers starreckon did not have

Run: setup · index · status · query · starreckon · --serial · --test · --accounts | Built on: faiss, numpy, sentence_transformers, fast-check

ai-architect-proof-of-work — Reproducible demos of my AI-security and evaluation tools. Every verdict is proven by execution: no LLM in any verdict path, no API keys, no network at runtime. 418 tests. Python, TypeScript, Solidity

Python 21,251 · TypeScript 6,252 · Solidity 2,693 · Shell 561 lines · 418 tests passing (run) · 14 commits on 5 days (2026-06-17 → 2026-10-31) · github.com/Alexander-Sorrell-IT/ai-architect-proof-of-work

Code: court_runner.py: Court runner — load a case, drive a single round, emit findings JSON · parcel.py: FAST parcel-readout engine — the data layer behind "see your land NOW"

Run: run · settle · --market · --live · --record · --asset · --points · --workers | Built on: orjson, pytest, pydantic, numpy, next, react, pandas, agents, viem, httpx

transcript-truth — 14-language multi-model ASR consensus. Ten named engines vote, **deduplicated by model family so a shared-lineage error cannot win a majority**. Model-proposed flags sit at weight zero and never enter the hard error score. The A–F verdict is a few lines of pure arithmetic that prints its own execution trace. Python

Python 20,796 · Shell 117 lines · 548 of 549 tests passing (run) · 117 commits on 20 days (2026-06-20 → 2026-08-17) · github.com/Alexander-Sorrell-IT/transcript-truth

Code: consensus.py: Consensus across independent acoustic witnesses · witness.py: Independent acoustic witnesses — strong ASRs of a different family than Whisper

Run: --commit · --check · --models · --keys · --transcript · --text · --lang · --model | Built on: faster_whisper, datasets, wordfreq, soundfile, sudachipy, pytest, transformers, torch, kiwipiepy, silero_vad

hexbreaker — Generates the exam, then sits it. **Forge** synthesizes DFIR cases deterministically (seed → manifest + separate answer key), adversarial mode included, so the graded set is built rather than scraped and the key never travels with the cases. **Court** is a five-role adversarial agent with a pure-Python judge and a runtime prompt-injection provocateur. F1 = 1.0 across five signed runs on the genuine NIST CFReDS disk image; 0.95–0.975 on standard generated cases; ~0.475–0.525 against a Forge corpus built specifically to fool it. SANS Find Evil! 2026. Python

Python 10,278 lines · 175 tests passing (run) · 60 commits on 13 days (2026-05-26 → 2026-08-17) · github.com/Alexander-Sorrell-IT/hexbreaker

Code: code quality A (97/100, from checks that ran) · court_runner.py: Court runner — load a case, drive a single round, emit findings JSON · build_trailer.py: Build the Hexbreaker TRAILER — VOICE-READY (no baked narration), nothing fabricated

Run: hexbreaker · --from-sweep · --vary · --seeds · --templates · --modes · --start-seed · --max-rounds | Built on: orjson, pytest, pydantic, PIL, numpy, Registry, faster_whisper, click, tenacity, httpx

glass-box-alpha — Verifiable AI reasoning on-chain: a four-agent ensemble commits each decision's reasoning hash before the outcome resolves. 150 tests across three stacks (59 Foundry, 76 Python, 15 TypeScript); 5 contracts deployed and source-verified on Mantle Sepolia. Python, Solidity, TypeScript

Python 3,601 · TypeScript 3,214 · Solidity 1,768 · CSS 50 lines · 150 tests passing (run) · 49 commits on 13 days (2026-05-26 → 2026-08-17) · github.com/Alexander-Sorrell-IT/glass-box-alpha

Code: code quality B (87/100, from checks that ran) · live.py: Live-runner composition root — the one file that binds the tested pieces to a real chain · service.py: Settler service — the off-chain orchestration loop

Run: run · settle · --market · --live · --record · --asset · --points · --workers | Built on: viem, loguru, react, next, openai, dotenv, eth_utils, pytest, @rainbow-me/rainbowkit, wagmi

robotruth — Deterministic receipts for AI-generated pull requests. Heuristics extract the model's claims, deterministic AST scanners read the diff, a pure function grades A–F. Every divergence between what the model claimed and what the diff shows cited at file:line. Zero models in the verdict path. MCP server enabled. Python

TypeScript 3,038 · Python 2,521 · CSS 286 · JavaScript 18 lines · 96 tests passing (run) · 83 commits on 11 days (2026-05-24 → 2026-10-31) · github.com/Alexander-Sorrell-IT/robotruth

Code: deploy.py: Deploy Receipt — the second surface of The Receipts Protocol

Run: --out · --n · --per-bot | Built on: next, react, httpx, robotruth_api, pydantic, pytest, fastapi, redis, robotruth_mcp, unidiff

grantscribe — You describe what you do. The Letter of Intent appears, in your voice, ready to submit — with a receipt the funder can verify.

Python 4,326 · HTML 137 · Shell 32 lines · 11 commits on 5 days (2026-05-24 → 2026-08-17) · github.com/Alexander-Sorrell-IT/grantscribe

Code: verify.py: GrantScribe — Verify every shipped claim, live · slack_app.py: GrantScribe — Slack front-end (Bolt, Socket Mode)

Run: list · find-grants · ask · learn · /setreport · /grants · /learn · /ask | Built on: dotenv, httpx, openai, mcp, playwright, numpy, soundfile, TTS, slack_sdk, parselmouth

invisible-seam — Finds where a project's claims contradict its behavior. Binary conflict isolation — The iPhone Problem doctrine made executable.
Python 3,658 · HTML 160 · Shell 138 lines · 64 tests passing (run) · 18 commits on 8 days (2026-09-25 → 2026-10-21) · github.com/Alexander-Sorrell-IT/invisible-seam
Code: code quality A (99/100, from checks that ran) · solidity_matcher.py: Solidity code matcher for the Invisible Seam engine
Run: seam · --report · --output | Built on: rich, click, yaml

pythia — Pythia — the agent forecaster credit market: AI agents trade narrative-rotation forwards on BNB Chain, settled hindsight-proof (no oracle, you re-derive it), with on-chain reputation as collateral. Proven forecasts are the collateral.
Python 3,298 lines · 25 tests passing (run) · 27 commits on 6 days (2026-06-17 → 2026-08-17) · github.com/Alexander-Sorrell-IT/pythia
Code: code quality A (97/100, from checks that ran) · escrow.py: Forward-settlement ACT — wrap ERC-8183 escrow around a NarrativePit forward · skill.py: Narrative Rotation Policy — the Track-2 backtestable CMC Skill
Built on: dotenv, requests, bnbagent, web3, eth_account, PIL, eth_utils, pytest

on-the-record — Receipt runtime for AI agents: an agent files before it acts, with the action and its evidence written atomically. **Won the T3 ADK Bounty Challenge, Launch Edition.** JavaScript
JavaScript 2,536 · Rust 603 · HTML 51 · Python 44 lines · 6 tests passing (run) · 12 commits on 7 days (2026-06-14 → 2026-08-17) · github.com/Alexander-Sorrell-IT/on-the-record
Code: code quality A (97/100, from checks that ran) · verifier.mjs: Standalone OFFLINE verifier for the "On the Record" receipt chain · render-filing.mjs: CLIENT-SIDE filing renderer for the "On the Record" receipt chain
Built on: @terminal3/t3n-sdk, @modelcontextprotocol/sdk, alloc, serde_json, sha2, @noble/curves, zod, playwright

sentinel — Grades somebody else's judge. Three layers: deterministic checks, UiPath's native LLM-as-judge left intact as the thing under test, and a hook-based pre-action interceptor as ground truth over both. Every scenario runs twice, guardrail off then on — the OFF→FAILED / ON→CERTIFIED flip is the only evidence a control fired rather than the test being easy. UiPath AgentHack 2026. Python
Python 2,850 · Shell 146 lines · 47 tests passing (run) · 35 commits on 4 days (2026-05-24 → 2026-09-13) · github.com/Alexander-Sorrell-IT/sentinel
Code: code quality A (97/100, from checks that ran) · demo_master.py: SENTINEL — Master Demo (all tracks, all platforms) One command · teleprompter.py: SENTINEL Teleprompter Reads phase signals from the demo's named pipe and displays the matching narration slide in a second terminal window
Run: --no-llm · --debug · --log · --cam · --no-rec | Built on: pydantic, anthropic, pytest, openai, requests, jinja2

anvilcourt-devvit — Receipts — every Reddit removal explains itself. Devvit mod tool: detects mod/AutoMod/spam removals, auto-explains to the author with appeal CTA, logs to a searchable audit trail.
TypeScript 2,921 lines · 67 tests passing (run) · 36 commits on 4 days (2026-05-24 → 2026-08-18) · github.com/Alexander-Sorrell-IT/anvilcourt-devvit
Code: code quality A (97/100, from checks that ran) · caseLaw.ts: No platform calls live here · audit.ts: Uses the @devvit/web/server redis client
Run: --watch | Built on: @devvit/web, vitest, esbuild, playwright

truthfit — Tailors a resume for ATS parsers and AI screeners using only the truth. The model proposes; a fail-closed verifier owns the verdict.
Python 2,160 lines · 19 tests passing (run) · 3 commits on 2 days (2026-10-17 → 2026-10-30) · github.com/Alexander-Sorrell-IT/truthfit
Code: code quality B (89/100, from checks that ran) · truthfit.py: Every job link in the given JSON/JSONL files, grouped by the engine that will read the application
Run: --resume · --jd-file · --jd-text · --url · --evidence · --company · --title · --out | Built on: pypdf

acme-refund-agent — Provable transaction authorization. Deterministic policy engine owns every approve and deny; an 8-attack scripted red-team battery held 8/8 with zero violations; emits tamper-evident audit receipts. Python
Python 1,576 · JavaScript 359 · HTML 120 · CSS 34 lines · 27 tests passing (run) · 16 commits on 6 days (2026-07-02 → 2026-08-17) · github.com/Alexander-Sorrell-IT/acme-refund-agent
Code: code quality A (97/100, from checks that ran) · verifier.mjs: Standalone OFFLINE verifier for the "On the Record" receipt chain · agent.py: The refund agent loop
Built on: dotenv, openai, fastapi, requests

terminal3-agent-mesh — An AI agent under hardware-enforced governance: it cannot overspend, cannot see the payment secret, and every action is audited.
TypeScript 905 · Rust 263 · HTML 226 lines · 2 tests passing (run) · 10 commits on 3 days (2026-06-06 → 2026-08-17) · github.com/Alexander-Sorrell-IT/terminal3-agent-mesh
Code: code quality A (97/100, from checks that ran) · agent-buyer.ts: Autonomous LLM buyer agent (brain = the claude CLI; no API key needed) · smoke.ts: SMOKE TEST — the gate
Built on: dotenv, @terminal3/t3n-sdk, playwright, ethers, alloc, serde_json

doomcaller — An adversarial patient that phones a healthcare voice agent and holds a real conversation; a role guard keeps the LLM caller in character and the grade is deterministic. 8 findings in 10 calls, 0 false positives. Python
Python 764 lines · 11 commits on 3 days (2026-06-21 → 2026-08-17) · github.com/Alexander-Sorrell-IT/doomcaller
Code: patient_llm_server.py: Vapi calls this OpenAI-compatible chat/completions for EVERY patient turn
Run: --facts · --llm · --report | Built on: groq, transcript_truth, uvicorn, fastapi, requests

AGENT INFRASTRUCTURE, GOVERNANCE & DEVELOPER TOOLING

agent-colosseum — Multi-model agent simulation platform — Nemotron hosts a box of AI agents, each backed by a different Crusoe model and rendered with a Perfect-Corp-generated character.
Python 6,217 · Shell 7 lines · 14 tests passing (run) · 41 commits on 6 days (2026-05-25 → 2026-08-17) · github.com/Alexander-Sorrell-IT/agent-colosseum
Code: app.py: Streamlit demo dashboard for Agent Colosseum · lark_red_team.py: Lark Gatekeeper Red Team — Automated adversarial testing of AI agent security boundaries
Run: list · run · design · compare-hosts · demo · lark · red-team · report | Built on: colosseum, dotenv, playwright, rich, openai, streamlit, requests, pydantic, pytest

argus — Splunk-native security triage for cross-chain DeFi. AI authors the detections; the triage verdict path makes zero AI calls, and only a real Foundry PASS on a mainnet fork may say CONFIRMED. Python, SPL, Solidity
Python 4,022 · Solidity 474 · JavaScript 406 · HTML 309 lines · 1 tests passing (run) · 37 commits on 7 days (2026-05-28 → 2026-08-17) · github.com/Alexander-Sorrell-IT/argus
Run: --chain · --days · --reset · --once · --tx-hash · --alert-type · --vuln-class · --contract | Built on: dotenv, playwright, requests, splunklib, mcp, PIL, pdfplumber, mlx_lm, transformers, torch

token-ledger
Python 3,023 · JavaScript 228 lines · 7 commits on 2 days (2026-08-05 → 2026-08-17) · github.com/Alexander-Sorrell-IT/token-ledger
Code: sessions.py: Is this tool's storage present at all? · check_hardware.py: Every Claude Code account with sessions on this machine
Run: token-ledger · --exclude-profile · --exclude-topic · --help · --no-snapshot · --out · --scan · --show-accounts | Built on: cryptography

CLI-Fleet — Launch multiple enforced Claude agent teams in parallel; hardware-aware wrapper over fleetcode + cli-enforcement + cli-wikia.
Shell 1,610 · Python 1,512 lines · 214 of 215 tests passing (run) · 18 commits on 9 days (2026-06-29 → 2026-10-11) · PyPI cli-fleet 6,973 downloads · github.com/Alexander-Sorrell-IT/CLI-Fleet
Code: code quality A (92/100, from checks that ran) · protocol.sh: Multi-team communication protocol — shared functions All teams use the same message format and directory layout · models.py: Per-model launch specs for multi-CLI fleets
Run: launch · status · dashboard · send · cleanup · where · daemon · start | Built on: pytest, cli_enforcement, cli_wikia

CLI-Wikia — Offline, pip-installable reference wiki for AI coding CLIs (Claude, DeepSeek, Copilot, ChatGPT, Gemini) with a search/read/ask command.
Python 2,884 lines · 83 tests passing (run) · 31 commits on 9 days (2026-06-28 → 2026-10-11) · PyPI cli-wikia 9,181 downloads · github.com/Alexander-Sorrell-IT/CLI-Wikia
Code: code quality A (99/100, from checks that ran) · cli.py: Command-line interface for cli-wikia · hooks.py: NOTHING here is a hardcoded per-model table
Run: models · list · read · versions · search · path · ask · update | Built on: pytest

keeperhub — KeeperHub Agent Economy Hackathon — argus + KeeperHub deterministic onchain execution layer. The agent with its own laws.
Python 2,395 · Shell 191 · Swift 151 lines · 17 commits on 3 days (2026-09-15 → 2026-09-17) · github.com/Alexander-Sorrell-IT/keeperhub
Code: demo.py: THEMIS — Demo (KeeperHub Agent Economy Hackathon) One command · core.py: Goddess of divine law
Run: --debug · --log · --no-new · --no-pause | Built on: requests

fleetcode — Launch multiple Claude Code agent teams in parallel terminal windows with a shared filesystem mailbox for cross-team coordination.
Shell 1,473 lines · 2 commits on 2 days (2026-06-29 → 2026-08-17) · github.com/Alexander-Sorrell-IT/fleetcode
Code: protocol.sh: Multi-team communication protocol — shared functions All teams use the same message format and directory layout
Run: --force · --background · --example

CLI-Collective — The Collective — one install for the AI coding-agent stack: cli-wikia + cli-enforcement + cli-fleet. One install for the whole stack.
Python 412 lines · 14 tests passing (run) · 18 commits on 8 days (2026-06-29 → 2026-08-28) · PyPI cli-collective 7,409 downloads · github.com/Alexander-Sorrell-IT/CLI-Collective
Code: code quality A (100/100, from checks that ran)
Run: status · daemon · start · stop · logs · cli-collective · --version · --foreground | Built on: tomli, pytest

PRODUCTS, APPS & SITES

tierva — Tierva — see your land: consumer Earth-intelligence platform (vision, research, strategy). Pacto = giving protocol; Pacto Seco = drought vertical.
Python 3,416 · Solidity 925 · HTML 529 lines · 45 of 48 tests passing (run) · 12 commits on 5 days (2026-06-04 → 2026-08-17) · github.com/Alexander-Sorrell-IT/tierva
Code: parcel.py: FAST parcel-readout engine — the data layer behind "see your land NOW" · consensus.py: Multi-sensor consensus engine — domain-agnostic parametric trigger
Run: --rpc · --pacto · --oracle-key · --pool · --cycle · --trigger-threshold · --breach-below · --breach-above | Built on: numpy, pandas, rioxarray, odc, requests, rasterio, xarray, shapely, fastapi, pystac_client

OTHER WORK

ghost-reviews — Detect AI-ghostwritten reviews: paste a product, see how much of its rating is real humans vs AI, and whether to trust it. Paste a product's reviews → AI exposes the fakes ("ghost" reviews) → you get a trust verdict + the real human-vs-AI rating gap.
Python 989 lines · 10 commits on 4 days (2026-06-07 → 2026-08-17) · github.com/Alexander-Sorrell-IT/ghost-reviews
Code: detector.py: The brain of ghost.reviews — ONE analyzing AI (no other live models) · nimble_client.py: Nimble client — pulls live review text from across the web
Built on: dotenv, streamlit, anthropic, pyarrow, tower, requests

Smaller repositories: account-service, apple-of-the-world, cad-fea-pipeline, Cantos-Anthems, Epics-Sagas, matrixbuilderops, refract.

starforge — Token counter for AI-coding sessions, checked against an independently written counter by a different author: over 20,217 transcripts from 5 computers they agree exactly on every axis but output tokens, where they differ by 0.07%. 4,878 npm downloads (starforge-cli). JavaScript

CONTRIBUTIONS TO OTHER PEOPLE'S REPOSITORIES

- **Merged** — KeeperHub/keeperhub: chore(plugins/math): add multi-source consensus-tolerance action node (2026-09-17) · github.com/KeeperHub/keeperhub/pull/2559
- **Closed** — profullstack/sh1pt: feat(targets): add pkg-pacman target (Arch custom repository) (2026-05-29) · github.com/profullstack/sh1pt/pull/477
- **Closed** — profullstack/sh1pt: feat(targets): add pkg-dnf target (dnf repo / Fedora COPR) (2026-05-29) · github.com/profullstack/sh1pt/pull/476
- **Closed** — profullstack/sh1pt: feat(targets): add pkg-chocolatey target (Chocolatey Community Repository) (2026-05-29) · github.com/profullstack/sh1pt/pull/475
- **Closed** — profullstack/sh1pt: feat(targets): add mobile-amazon target (Amazon Appstore, Android phones & tablets) (2026-05-29) · github.com/profullstack/sh1pt/pull/466

SHIPPED TOOLING & PUBLIC ADOPTION (TELEMETRY VERIFIED)

- **46,179 lifetime downloads** across 11 published packages and 129 releases on PyPI and npm (live, 2026-11-01).
- **PyPI (33,873):** cli-enforcement (10.3k), cli-wikia (9.2k), cli-collective (7.4k), cli-fleet (7.0k).
- **npm (12,306):** starforge-cli (4.9k), starreckon (2.8k), package-telemetry (1.9k), sitemap-mcp (1.1k), filelens-mcp (1.1k), verifier-mcp (0.2k), sorrell-engine-mcp (0.2k).
- *Note:* Gross telemetry includes registry mirrors and CI runs; volunteered as directional proof of real-world usage, not audited human seats.

EDUCATION & CERTIFICATIONS

B.S., Integrative Studies — University of North Texas · 2019 · CompTIA Security+ · Network+ · A+

Alexander Sorrell — Verified Package Telemetry & Live Proof

Identity: Alexander Sorrell (@Alexander-Sorrell-IT) · alexander.sorrell.it@gmail.com · (817) 996-6123

Source: Real-time Registry API query via package_telemetry.py · Executed: 2026-11-01

LIVE TERMINAL TELEMETRY OUTPUT (PACKAGE_TELEMETRY.PY)

```
================================================================================================================================================================
                                                ALEXANDER SORRELL — PACKAGE METRICS & TELEMETRY
================================================================================================================================================================
Type   | Package          | First Rel. | First Date  | Current    | Updated     | Days Live |  Last Day |  Today |  DL / Month |  Lifetime DL
----------------------------------------------------------------------------------------------------------------------------------------------------------------
pip    | cli-enforcement  | v0.1.0     | 2026-06-29  | v0.10.0    | 2026-10-11  |      125d |         6 |      0 |       1,308 |       10,297
pip    | cli-wikia        | v0.10.0    | 2026-06-29  | v0.21.0    | 2026-10-11  |      125d |         6 |      0 |       1,247 |        9,187
pip    | cli-collective   | v0.1.0     | 2026-06-29  | v0.7.0     | 2026-08-29  |      125d |         2 |      0 |         719 |        7,411
pip    | cli-fleet        | v0.1.0     | 2026-06-29  | v0.9.0     | 2026-10-11  |      125d |         5 |      0 |       1,060 |        6,978
----------------------------------------------------------------------------------------------------------------------------------------------------------------
npx    | starforge-cli    | v0.1.0     | 2026-08-07  | v0.14.0    | 2026-08-13  |       86d |         5 |      0 |          67 |        4,878
npx    | sitemap-mcp      | v0.1.0     | 2026-08-13  | v0.7.0     | 2026-08-28  |       80d |         5 |      0 |          37 |        1,130
npx    | filelens-mcp     | v0.1.0     | 2026-08-13  | v0.7.0     | 2026-08-28  |       80d |         0 |      0 |          43 |        1,124
npx    | starreckon       | v0.14.0    | 2026-08-14  | v0.31.0    | 2026-10-11  |       79d |        15 |      0 |         718 |        2,820
npx    | verifier-mcp     | v1.0.0     | 2026-09-04  | v1.0.0     | 2026-09-04  |       58d |         2 |      0 |          31 |          230
npx    | sorrell-engine-mcp | v1.0.0   | 2026-09-04  | v1.0.0     | 2026-09-04  |       58d |         1 |      0 |          22 |          207
npx    | package-telemetry | v1.0.0    | 2026-09-06  | v1.6.0     | 2026-10-11  |       56d |        13 |      0 |         929 |        1,917
================================================================================================================================================================
                                                AVERAGES & SUMMARY
================================================================================================================================================================
Category         |  Packages |  Avg Days Live |  Avg Last Day |  Avg Today |  Avg DL / Month |  Avg Lifetime / Pkg |        Total Lifetime DL
----------------------------------------------------------------------------------------------------------------------------------------------------------------
PyPI (pip)       |    4 pkgs |        125.0d |     4.8 / day |     0.0 / day |     1,083.5 / mo |       8,468.2 / pkg |                33,873
NPM (npx)        |    7 pkgs |         71.0d |     5.9 / day |     0.0 / day |       263.9 / mo |       1,758.0 / pkg |                12,306
----------------------------------------------------------------------------------------------------------------------------------------------------------------
ALL ECOSYSTEM    |   11 pkgs |         90.6d |     5.5 / day |     0.0 / day |       561.9 / mo |       4,198.1 / pkg |                46,179
================================================================================================================================================================


================================================================================================================================================================
                                                PROGRAM DIRECTORY & WHAT THEY DO
================================================================================================================================================================
Type   | Package            | What It Does / Architecture
----------------------------------------------------------------------------------------------------------------------------------------------------------------
pip    | cli-enforcement    | Model-agnostic hook-level behavioral enforcement engine for AI coding agents.
pip    | cli-wikia          | Offline, pip-installable reference wiki & search index for AI CLIs (Claude, Gemini, Copilot, DeepSeek).
pip    | cli-collective     | The Collective bundle: knowledge (cli-wikia) + control (cli-enforcement) + parallel execution (cli-fleet).
pip    | cli-fleet          | Hardware-aware parallel Claude Code agent orchestration via shared filesystem mailboxes.
----------------------------------------------------------------------------------------------------------------------------------------------------------------
npx    | starforge-cli      | Verification CLI auditing AI-coding telemetry with kernel-enforced no-egress proof.
npx    | sitemap-mcp        | MCP server providing website structural awareness without HTML token explosion.
npx    | filelens-mcp       | MCP server for single-call intelligent AST & structural file reading.
npx    | starreckon         | Local-only developer wrapped analyzer for disk logs (Claude Code, Cowork, Codex).
npx    | verifier-mcp       | Deterministic EU e-invoice compliance engine and paradox resolver (DevNetwork 2026).
npx    | sorrell-engine-mcp | Browser-native neural OS & Tiny Model Swarm runtime (OpenAI WebMCP Challenge).
npx    | package-telemetry  | Cross-registry package telemetry engine for tracking PyPI and NPM package metrics, releases, and downloads.
================================================================================================================================================================
Registry Ledger: packages_config.json  •  Canonical Output: package_telemetry_latest.json  •  Scope: PyPI & npm live registries
```

QR CODE TARGET ENGINE & REPRODUCIBLE VERIFICATION (NPX STARRECKON PROVE)

Routing Destination: github.com/Alexander-Sorrell-IT/starreckon · **Package:** npx starreckon

- **Reproducible Sandbox Execution:** npx starreckon prove prints the exact confinement command without running it, so you read the proof before you trust it. The command runs the scan under Apple sandbox-exec (Linux: unshare -rn) with a profile that denies network egress: (version 1) (allow default) (deny network*) (deny network-outbound) (deny network-inbound).
- **Kernel-Enforced Boundary:** The kernel refuses the socket (EPERM), so network egress is blocked below the application, not by an application-level boolean. A timeout is deliberately not counted as blocked.
- **The Central Engineering Thesis:** *The model proposes; deterministic executed checks own the verdict.*

PUBLIC INDEPENDENT VERIFICATION ENDPOINTS (ZERO INSTALL REQUIRED)

Anyone can query public package registries directly via cURL to independently verify these install metrics:

```
# 1. PyPI Verification (pypistats API, no key) — reproduces the PyPI figures above, which include mirrors
curl -s https://pypistats.org/api/packages/cli-enforcement/overall | jq '[.data[]|select(.category=="with_mirrors").downloads]|add'
#    without mirrors, the four PyPI packages total 9,892:
curl -s https://pypistats.org/api/packages/cli-enforcement/overall | jq '[.data[]|select(.category=="without_mirrors").downloads]|add'

# 2. npm Verification (via npm Registry API)
curl -s https://api.npmjs.org/downloads/point/last-month/starreckon | jq .downloads
curl -s https://api.npmjs.org/downloads/range/2020-01-01:2030-12-31/starforge-cli | jq '[.downloads[].downloads]|add'
```