

Gustavo Schneider

AI Infrastructure Engineer — execution infrastructure for AI agents · Rust · Distributed Systems

São Paulo, Brazil (Remote) · gustavo@humangr.com · humanguardrail.github.io · github.com/HumanGuardrail · linkedin.com/in/gustavo-schneider

SUMMARY

I've always had a problem-solving mindset, and I've come to see that the real bottleneck in software development today is neither the people nor writing code — it's the infrastructure, methods, and processes around them, almost none of it built for the pace at which we now work. That's why I built HuGR: infrastructure for developers to build and ship safely and fast, with ownership and transparency end to end — a content-addressed cache, the compute and runtime that run on it, and the tooling agents use to ship code, written in Rust — from edge workers down to the container runtime. I've had the rare opportunity to watch what's probably the fastest revolution any industry has ever gone through, up close, and I've spent the last two years out on that frontier — understanding the limits and challenges of working with LLMs, and developing the processes and methods to get production-grade software out of them. Everything is built to be verified, not trusted: TLA+, red-team, real benchmarks.

EXPERIENCE

Founder & Principal Engineer — HuGR

Remote · March 2026 – Present

Building developer- and AI-agent infrastructure in Rust, so fleets of coding agents can ship at full speed without governance, CI, or builds becoming the bottleneck. Two flagships: **CoreLink**, the content-addressed platform (92 Rust crates across three repositories), and **hugit**, a git-native forge for agent fleets.

CoreLink — content-addressed platform

- **Cache** — multi-tenant content-addressed store + action cache (BLAKE3 content addressing; fixed or FastCDC chunking), so identical work is never repeated. Within an org it's a *shared* cache — a teammate's build becomes your instant hit — while cryptographic tenant isolation (model-checked in TLA+) walls it off from every other org. One store serves Bazel (REAPI v2), Turborepo and sccache/cargo, plus brew, npm, pip and OCI cache surfaces; Rust in Cloudflare Containers behind an edge control plane, five production environments live.
- **Runners** — ephemeral, cache-warm compute billed by concurrency, not per-minute. Each job in a fresh, fail-closed microVM, destroyed after; digest-pinned images verified at the acquire gate (unpinned refused); per-lease, write-only, domain-separated tokens instead of master credentials in the box; every result is an Ed25519-signed attestation binding the full outcome. Cap-safe cross-instance admission on Postgres advisory locks, proven live; hardened by a recursive adversarial audit — 28 confirmed findings, incl. real container-escape vectors, all closed, re-audited to zero P0/P1.
- **Workspaces** — the working tree as a content-addressed object, compute as a disposable cursor: `c!w` snapshots a directory, hydrates it on any machine in seconds, and memoizes commands so they never re-run on unchanged inputs. Shipped to Homebrew via a five-target release pipeline with minisign-signed artifacts and per-crate SBOMs.

hugit — git-compatible, LLM-native forge for agent fleets (Rust, ~148k LOC, 20 crates; free product, closed source)

- Serves the real git wire protocol (push/clone) directly from content-addressed storage — receive-pack with thin-pack support and live ref hot-swap; refs are event-sourced with compensating-event undo, so force-push data loss is structurally unexpressible.
- Union-tested landing queue: batches branches and tests them atomically; on red, a bisect isolates the culprit in $\leq \lceil \log_2 n \rceil$ probes — a bound the test suite proves. Checks are memoized against CoreLink's Action Cache — identical checks never re-run.
- Every commit carries an intent envelope (charter, model, cost, claims), and claim fences materialize only the paths an agent declared; tree-sitter symbol outlines across 8 languages at the read API.
- Ed25519/SHA-256 attestation spine hardened through 13 adversarial security-review rounds; 1,971 tests green in CI, clippy - D + cargo-deny gates (0 advisories across 250 pinned deps). Deployed multi-repo with a live read+write API; push/clone proven in CI — single-tenant today, multi-tenant staged.

Method & correctness

- Built **HuGR TechLead** — method and tooling for orchestrating clusters of LLM coding agents: judgment centralized, contract-frozen work packages, write-sliced so 20+ agents integrate with zero merge conflicts, each routed to the cheapest model that fits.
- Its verification gate, **HuGR Relay**, matches each stopping agent to its own Definition of Done deterministically and advances it only on real checks (tests, lint, grep) — a “done” that fails its gate is blocked and sent back for rework.
- 48 TLA+ specifications (tenant isolation, GC, failover split-brain, erasure atomicity) model-checked in CI; BYOK envelope encryption across four KMS providers, offline-verifiable Ed25519-signed hash-chained audit log, GDPR/LGPD erasure with signed attestation; SLSA L3 supply chain with SHA-256-pinned cross-repo conformance vectors.
- Code-grounded architecture wiki (Open Knowledge Format): 157 concepts, each naming the source files it explains and pinned to a commit, drift-gated in CI — when cited code changes, a pipeline dispatches an agent that reconciles the concept automatically.

Freelance Developer

Independent · 2025

- Freelance development, and the technical groundwork that led to founding HuGR.

Product Owner, later Project Manager — OpenSolo Soluções

Remote · 2022 — 2024

- Founding Product Owner (day zero): led product from first stakeholder conversations through specification, development, and delivery of a B2B commodities marketplace (catalog, orders, logistics, supplier onboarding) across Latin America.
- Designed and delivered B2B foreign-exchange modules integrated with Banco Bexs — primary liaison between the bank and OpenSolo through the entire integration lifecycle, in a trilingual (EN/ES/PT) environment; promoted to Project Manager, running delivery and risk across concurrent workstreams.

Project Manager — Temple Roofing

Melbourne, Australia · 2019 — 2022

- Ran operations for a construction company: payments, materials and tooling inventory, purchasing, and team scheduling.

SELECTED PROJECTS

- **Lightr** — daemonless, imageless container runtime in Rust (~90k LOC, open source): rootless cold starts at **30.8 ms (4× rootless podman, identical isolation)** on public CI, memoized re-builds **~2,000× (vs Docker redoing the build)**, a **Kubernetes CRI backend — critest v1.33 34/34 on the real backend — at 9.32× smaller resident footprint than containerd**, and a hand-rolled seccomp→cBPF compiler with a simulator test-bench. Built in 21 days / 482 commits by an agent fleet I orchestrated, with a public truth ledger. github.com/HumanGuardrail/hugr-lightr
- **Datarail** — end-to-end-sealed data rail and Kafka-wire-compatible broker in Rust whose storage never sees plaintext (`#![forbid(unsafe_code)]`, open source): sealed delivery (AEAD + Ed25519 + X25519 key-wrap), offline-verifiable Merkle delivery receipts, exactly-once into Postgres (CI-gated against real Postgres 16). Author-commissioned adversarial audit (run outside my own harness) of the real broker binary: **3.0 ms cold start, 3.3 MB idle RSS through a 50,000-message zero-loss cycle** — data directory grepped clean of plaintext. github.com/HumanGuardrail/HuGR_Datarail
- **HuGR Tools** — MCP tool-server fleet on Cloudflare Workers, ran in production as multi-tenant SaaS (v1.0.0, open source): **630 tools across 37 deployable workers** (Cloudflare, Stripe, Vercel, Sentry, PagerDuty, Supabase, 26 AWS packages, 5 Datadog), every worker on one frozen kernel — 3-tier fail-closed auth (RS256 → HMAC → introspection), Durable-Object distributed rate limiting, one dispatch pipeline. An OpenAPI→MCP forge generated the AWS surface; the value is the handwritten composite tools — Datadog temporal root-cause correlation, Stripe multi-currency VAT reconciliation, Vercel deployment-failure triage proven against a real failed deploy, S3 bucket-security audits. BYOK provider credentials per-request (never stored); **8,538 tests passing, 0 failing** incl. Stryker mutation testing with the full kill/survive breakdown published; staged-canary deploys with automatic rollback. github.com/HumanGuardrail/HuGR_Tools

TECHNICAL SKILLS

- **Languages:** Rust, Python, TypeScript, Go, SQL, Bash
- **AI/LLM engineering:** RAG pipelines (LlamaIndex, Qdrant, embedding pipelines), knowledge graphs (Graphiti, FalkorDB, Neo4j), knowledge engineering (Karpathy-style code wikis, Open Knowledge Format/OKF), guardrails (NeMo Guardrails, LLM Guard), evaluation (RAGAS, DeepEval, LLM-as-judge), LangGraph/LangChain/CrewAI/DSPy/Pydantic AI, Temporal, vLLM/Ollama
- **AI-agent infra & LLMOps:** sandboxing (namespaces, seccomp/cBPF, Firecracker/Virtualization.framework microVMs), secrets brokering, ephemeral compute, memoized execution; LLM-agent fleet orchestration (Claude Code, Codex), multi-agent review, MCP, context engineering
- **Distributed systems & cloud:** Cloudflare (Workers, Durable Objects, R2, D1, Containers), PostgreSQL/Neon, replication & failover, advisory-lock concurrency; Terraform, Docker
- **Storage & protocols:** content-addressed storage, BLAKE3, FastCDC dedup; OCI, Kubernetes CRI, gRPC/protobuf, Bazel REAPI v2, Turborepo, sccache, OpenAPI 3.1
- **Security & cryptography:** Ed25519, HMAC, Argon2id, AES-GCM; envelope encryption / BYOK (AWS, GCP, Azure, Vault); SBOM, code signing, SLSA, cargo-deny; STRIDE/LINDDUN
- **Correctness:** TLA+, property-based testing, fuzzing, mutation testing, differential CI benchmarking; SLOs, OpenTelemetry
- **Practices:** Working Backwards, spec-driven & contract-first development, story-driven tests, disjoint multi-agent orchestration (zero-decision work packets, context hygiene, model-tiered dispatch, prep→dispatch→verify→integrate→clean loop), ADRs, tense-law documentation (measured claims on named hardware, or explicitly a target), CI/CD with formal-methods & mutation gates

ADDITIONAL

- **Product & process:** founding Product Owner → Project Manager background; Agile/Scrum, roadmap design, stakeholder management, formal technical specification (Lamport-style).

EDUCATION

- **Business Diploma (II)** — Greystone College · Melbourne, Australia · 2019–2020
- **Advertising** — Universidade Presbiteriana Mackenzie · São Paulo · 2016–2019 (*incomplete*)
- **Economics** — FAAP · São Paulo · 2014–2017 (*incomplete*)

LANGUAGES

English (fluent) · Portuguese (native) · Spanish (advanced)