

Adam Reed

Senior Software Engineer · Distributed Systems (.NET)

adam@adamreed.me
adamreed.me · github.com/iamadamreed · github.com/srs-adamr
Texas, USA · Remote

Self-taught, coding since my teens. Nine years a founder, five of them running a call-center MSP as CEO, and since 2021 hands-on CTO to two sister companies. The move to a senior IC seat is deliberate: I want new problems, a strong team, and collaboration. I architected a ~19-service event-driven .NET platform on NATS JetStream for a regulated payments and telephony business, wrote most of it, and operate it in production. I work from first principles: delete the bloat, keep costs low, FOSS where possible, and pay for a product that just works. I built the AI-assisted SDLC the platform is developed with. I can lead or carry work alone.

■ CORE SKILLS

Languages	C# / .NET (6–10), SQL (MySQL/MariaDB, T-SQL), TypeScript, Python, Swift 6, Bash
Distributed systems	event-driven microservices, NATS JetStream, contract-first messaging (OpenAPI contracts over NATS), gRPC/protobuf, REST, idempotency & exactly-once design, optimistic-CAS state machines, high availability, fault tolerance
Concurrency & perf	multi-threading, channels & backpressure, Native AOT, zero-allocation (Span/IBufferWriter), GPU/ONNX CUDA, performance tuning
Operations	observability (OpenTelemetry, Grafana, Loki, Tempo, self-hosted), on-call & incident reviews, hybrid Azure & on-prem
AI-assisted SDLC	agent orchestration, MCP (Model Context Protocol) server authoring, agent SDK development, control-protocol reverse engineering, self-hosted LLM routing
Open source	TONL.NET zero-alloc serializer, legatus agent-fleet orchestrator (Swift 6), 20+ vendor SDKs on NuGet

■ EXPERIENCE

Reed Enterprising — Founder & CTO

2021–Present · Remote

My firm. I contract as hands-on CTO to two sister companies, Spire Recovery Solutions and Metacorp, and own their estates end to end: a ~19-microservice event-driven platform, a multi-region Azure footprint (~270 resources, 19 Container Apps), and an on-prem datacenter I designed and built: Proxmox clusters, ZFS storage in the hundreds of terabytes, a 6-GPU inference fleet, an Android device fleet, and a 3-node NATS cluster. I wrote most of the code and operate all of it.

Spire Recovery Solutions — CTO · ~200-seat collections operation, 90M+ call records, ~19-microservice platform, SOC 2 Type 2 / PCI / FDCA

- Wrote **Conclave**, the call-lifecycle service. Three telco sources report the same call out of order, so a pure, I/O-free reducer folds their events into one authoritative record under a bounded optimistic-CAS loop on NATS JetStream KV. One decision function serves the live path, retries, sweep, and DB fallback. It replaced the MySQL re-query hot path.
- Brought gRPC's contract discipline to **NATS**. Our OpenAPI model contracts double as message contracts, generated into other languages, with in-house Roslyn analyzers failing the build on contract drift, so services in different languages agree at compile time instead of at runtime.
- Wrote **Cerebro**, the identity and authorization plane: one auth model across HTTP and the message bus, revocable live, SOC 2 auditable. RS256 JWTs plus Ed25519-signed NATS credentials minted through an auth-callout, per-user subject grants that change with no broker reload, and an audit interceptor that redacts fields before they reach a log.
- Live-migrated a ~1-billion-row production MariaDB that had no replica to Azure MySQL, on a resumable 12-phase orchestrator I wrote with checkpoint validation and rollback at every phase.
- Built the AI infrastructure the platform is developed with: **ClaudeAgent.NET** (a zero-dependency SDK reverse-engineered from Claude Code's undocumented control protocol), Archon (the production chat agent platform, one Claude session per thread), and Oracle (an MCP server exposing read-only MySQL, Loki, Tempo, and Prometheus access with PII denied by a column-level grant boundary no query phrasing gets past).
- I am the **sole on-call** for the platform and I write the incident reviews. Quorum-aware rolling deploys check JetStream Raft replica health before restarting a node. W3C trace context rides NATS headers end to end, with FDCA-driven PII redaction built into the telemetry layer.
- Took NVIDIA's NeMo ASR service and **deconstructed it into a hand-crafted ONNX Runtime CUDA pipeline** so transcription runs in-house: direct P/Invoke, CUDA graph capture, my own batching and per-GPU dispatch, on GPU hosts I racked myself. When production hit SIGSEGVs in stream capture I built ONNX Runtime's unshipped CUDA Plugin EP from source, patching a release bug at the tag.
- Carried **RapidContact**, the e-signature and consumer-correspondence platform I first built at VTR, into Spire and rebuilt it: first on Azure Durable Functions and Cosmos DB, then as ~14 cooperating queue workers with ESIGN/UETA capture, a fail-closed compliance gate, and an HMAC-signed Cloudflare edge. For outages and DB maintenance, its FILL/DRAIN mode collapses the service to one webhook-capture endpoint buffering into an Azure queue, then replays through the live code paths.

- Shipped a native **Android dialer** (.NET MAUI, Telecom InCallService) to a physical 3-carrier device fleet that measures how carriers label our caller ID. Each test call costs money, so results publish before ack with dedup on JetStream: a crash cannot double-bill or lose a result.
- I own the production **Asterisk/FreePBX** itself: Lua caller-ID pool selection over ODBC, and a 7-tier geographic caller-ID fallback picked live per call from a North American Numbering Plan and Census dataset I assembled. Corsair, the inbound DID-to-dialplan resolver, falls back to a default route so a database failure never breaks inbound calls.
- Built **Terminus**, our in-house payment gateway. One abstraction across USAePay and Paywire, tokenization kept strictly separate from transactions.
- Built the infrastructure the same way, first principles and under budget: a direct cloud interconnect to Azure and a private circuit to our telephony carrier instead of VPN boxes and inspection appliances, plus 12+ self-hosted runners serving 129 workflows across 45 repos.
- The current platform is the **second generation** I have built at Spire. The first, 2021–2024 on Azure DevOps, ran 34 repos and ~7,400 commits, including Terminus, plus the CleanCollect collections core carried over from VTR, and bridged a legacy AS/400 dialer over IBM iAccess, ODBC, SFTP, and VPN. When the dialer's vendor SDK died I decompiled the SDK into five buildable .NET 9 libraries.
- Across VTR and Spire I have led teams of 4–10 developers for nine years, most of them junior hires I trained up inside a tight budget, and personally reviewed ~95% of the ~760 PRs they shipped in that time. One intern I mentored in C# is now an L2 developer at Microsoft. Standards live in the build: Roslyn analyzers and banned-API gates.

Metacorp — CTO · *sister company, collections platform, .NET modernization & data engineering*

- Ran three parallel Claude Code agents in isolated git worktrees to migrate **166 legacy stored procedures to MySQL 8** as individually human-reviewed PRs, in about six hours.
- Extracted a wire-compatible **.NET 10 Native AOT** API from their legacy monolith and moved traffic over piece by piece, on Azure Container Apps behind Cloudflare. Fixed a timing-attack vulnerability in its token check.
- Built an SFTP-to-Google-Drive skip-trace sync where the vendor deleting the uploaded file is the success signal: a per-file state machine with the database authoritative over the filesystem.

VTR Technology — Founder & CEO

2017–2022 · New York

Founded and ran a vertically integrated call-center MSP for five years: 59 customer agencies, 48 on recurring contracts, a 9-developer team I hired and led. The codebase ran 23 repos and ~4,800 commits, all hand-written years before AI coding tools. One customer became Spire, where I'm now CTO.

- We built call centers from the empty room up: structured cabling, cameras, UniFi networks, workstation fleets, business fiber, wholesale carrier voice, and hosted PBX, MySQL, web, and email across dozens of agency sites. I pulled cable and racked gear myself.
- Wrote the telecom core: Asterisk dialplans with ODBC caller-ID lookup, STIR/SHAKEN attestation, caller-ID reputation. That stack is the foundation Spire's platform was rebuilt on.
- Built **CleanCollect**, a multi-tenant debt-collections platform (ASP.NET, Azure Functions, Twilio, USAePay, skip-trace, automated debtor mail) that agencies ran their entire operations on. The v1.5 rewrite alone: 2,516 commits, 7 contributors, CI. Its core carried into Spire's platform.
- Built **RapidContact**, a multi-tenant e-signature and consumer-correspondence platform: templated letters, email, and SMS to consumers, and in-house e-signature capture with an IP, geolocation, and timestamp audit trail appended to each signed PDF. The stack: ASP.NET MVC, Azure Durable Functions, and Cosmos DB, with a .NET SDK an agency's collection system drives in code. 861 commits, 10 contributors, the org's most collaborative repo.

Earlier · Union Millwright & Self-Taught Developer (before 2017, New York) · *Local 1163 Millwright/Carpenter: power plants, waste treatment, and chip fabs, including installing FOUP-carrying robots at GlobalFoundries Fab 8. Started programming in my teens writing for WoWGliders, the World of Warcraft bot, and taught myself C# by reverse engineering.*

■ SELECTED SYSTEMS & OPEN SOURCE

- **TONL.NET**: a serializer written from nothing. Zero-allocation ref-struct readers and writers over Span and IBufferWriter, a Roslyn incremental source generator, 360+ spec-compliance tests. It is the production wire format our Oracle MCP server speaks to LLM agents.
- **legatus**: Swift 6 strict-concurrency agent-fleet orchestrator, ~10k LOC. One git worktree per agent, actor-isolated task claiming stress-tested at 50 concurrent claimers, live steering of Claude Code over its control protocol.
- **TCN.NET**: SDK over a vendored 715-file protobuf tree as the contract source of truth. Connect-RPC error-envelope parsing and NDJSON streaming surfaced as IAsyncEnumerable. Caught a real enum-drift bug in the vendor's API.
- **augment**: self-hosted LLM routing gateway. A GPU-hosted, grammar-constrained local classifier routes traffic across three providers using top-2 logprob margin as the escalation signal.
- **switchboard**: a .NET 10 Kestrel MITM proxy for Claude Code traffic that streams SSE live while recording every exchange to replayable JSON.
- Upstream fixes in **LiteLLM** (OAuth header forwarding), Synadia's callout.net NATS auth library (breaking-change compat), and the Orca agent orchestrator (per-worktree account pinning). 20+ typed vendor SDKs on NuGet across payments, telephony, SMS compliance, and HR, several Native AOT.

The work lives across two GitHub accounts: github.com/iamadamreed (personal and open source) · github.com/srs-adamr (Spire platform).