

allocsys

Backend Developer — APIs, Infra & AI-Agent Tooling

allocsys@gmail.com · github.com/allocsys · allocsys.github.io/allocsys

SUMMARY

Backend-focused developer who builds the infrastructure layer AI agents actually run on — auth, rate limiting, multi-service integrations — rather than just wiring prompts together. Comfortable owning a project end-to-end: designing the API surface, hardening it (timing-safe auth, IP allowlisting, proxy-aware rate limiting), and documenting the reasoning so someone else can maintain it. Recent focus: Model Context Protocol (MCP) servers and agentic workflow automation (n8n + LangChain), plus root-causing bugs in real-world open-source infrastructure across JavaScript, Python, Go, and Rust codebases. Ships full-stack TypeScript apps on Cloudflare Workers.

PROJECTS

madmcp

Node.js · Express · MCP server · Redis

- Built and deployed an MCP server exposing ~50 tools across 8 connectors (GitHub, Cloudflare D1/KV/R2/Workers, Notion, Mem0, generic fetch, Exa, Gemini) to any MCP-compatible AI client.
- Designed a delegation-first architecture: one tool hands an open-ended, multi-step investigation to a second model that runs its own tool loop server-side, with Redis-backed checkpointing so a rate-limited run resumes from its last step instead of restarting.
- Separated tools that touch untrusted external content (web research, fetch) from those that touch internal systems (GitHub, Cloudflare, Notion), so external data can't reach credentials.
- Designed the auth layer: timing-safe key comparison, IP allowlisting scoped to Claude's connector range, and per-route rate limiting, ordered so a leaked key alone can't get in. Handled proxy-hop IP trust, large-push body limits, and per-connector token failure; PR-based workflow with GitHub Actions CI, vitest and supertest.

proetsy

Node.js · React · SQLite · Electron · Playwright

- Built a local-first pipeline that turns artwork into ready-to-paste Etsy listings and mockups: optional image analysis, listing generation, PSD-template mockups, and a human-in-the-loop review step (publishing stays manual by design).
- Built an LLM provider layer with Gemini primary and Claude fallback, key store, queue and rate limits, plus an embedding-based taste filter in pure JS/WASM (runs on Termux).
- Tested every module (unit, idempotency, route-level integration) and added Playwright E2E against a deterministic fixture LLM; shipped a CI-verified Electron Windows installer.

news-market-ai

Cloudflare Workers · D1 · KV · Gemini · Next.js

- Built a news-to-trade-signal pipeline: ingestion from free sources and custom scrapers, multi-agent LLM analysis, and signal generation, designed around free-tier Worker limits.
- Wrote a point-in-time backtester and documented rollout gates (backtest → paper → micro-live); live trading is deliberately off.
- Built a multi-key Gemini model-cascade client with KV cooldown state, and a mobile-friendly Next.js 16 ops dashboard served as a Worker with login and a session-gated API.

ai-campaign-builder

TypeScript · React · Cloudflare Workers · D1

- Built a monorepo where a business owner states a goal and an AI agent generates a loyalty or gamification campaign. Apps: owner dashboard, customer app, staff POS, review console, public microsites, and a backend API.
- Shared api-client, UI-kit and campaign-editor packages; JWT auth, D1 migrations, Vitest tests, and CI deploys that provision D1 and KV automatically.

peakplast

Cloudflare Workers · Telegram · Gemini · Zarinpal

- Built and run a production Telegram ordering bot: Gemini voice-order extraction with multi-key fallback, Zarinpal payment links and verification, an admin panel, and a REST API.
- Handled concurrency and abuse (per-chat locking, flood limits). Splitting the ~7,000-line single-file Worker into ES modules, one verified, deployable step at a time.

clothes-shop

Next.js · TypeScript · Tailwind CSS · RTL

- Built the front-end of a Persian (RTL) online clothing shop: home, shop, product, cart and account pages with a mobile tab bar. Backend and payments are planned next.

Agentic Email Triage & Ops Automation

n8n · LangChain · RAG

- Designed an n8n workflow where an AI agent classifies inbound email across Gmail, Telegram and WhatsApp, answers from a RAG knowledge base, and drafts replies only for non-urgent cases.
- Put the safety boundary in the system prompt: escalation-worthy emails are flagged for a human and never auto-replied to. Used an OpenAI → Gemini → Claude fallback chain and forced JSON output so Sheets, CRM and LINE nodes route deterministically; tested on 6 scenarios.

Developer tools (Termux / Android)

Bash · Node.js · Playwright · Android

- cf-manager: menu-driven Cloudflare admin panel in one bash script: Workers deploy/rollback, KV/D1/R2/Hyperdrive, OAuth2 + PKCE with multi-account switching, GitHub sync.
- logger (DOM mapper): maps a live DOM, network traffic and replayable actions via remote Browserless.io, working around Playwright's lack of Android support.
- ezgit (Git helper that redacts secrets before diffs reach an LLM) and openhands-android-client (native client for OpenHands coding agents).

OPEN SOURCE CONTRIBUTIONS

grafana/alloy

Go · PR #6750 · merged

- Root-caused a silent failure in loki.source.file / loki.source.journal: an unreadable legacy Promtail positions file was indistinguishable from a missing one, so the component silently re-ingested an entire log stream. The reader now returns an error that fails loudly at construction, while missing or malformed files stay non-fatal.

cloudflare/workers-sdk

TypeScript · 2 PRs · 1 merged, 1 open

- Root-caused a Windows-only crash from non-ASCII paths (Latin-1 header violation) across 5 call sites, verified with a Windows CI run — merged (PR #14713).
- Fixed local Images binding transform parity (fit/gravity/background) with production — open and in review (PR #14810).

transact-rs/sqlx

Rust · PR #4351 · open

- Fixed the Any driver emitting ? placeholders on Postgres (issue #3000) with a SQL-aware rewrite to \$N that skips string literals, quoted identifiers, dollar-quoted strings, and comments, applied on the Postgres path only so MySQL and SQLite are unaffected. CI passes on the fork.

release-plz/release-plz

Rust · PR #2954 · open

- Relaxed the fast-path condition in package-file comparison so registry-extracted packages skip a slower, occasionally-failing cargo package --list call.

anthropics/skills

Python · 4 PRs · open

- Fixed a PDF form-field parser bug, an xlsx external-link safety-check case-sensitivity gap, a docx Python 3.9 import crash, and a missing PyYAML dependency.

SKILLS

Languages

JavaScript / TypeScript, Node.js, Python, SQL, Bash; Go, Rust

Backend & APIs

Express, REST API design, OAuth2 / PKCE, JWT, rate limiting, IP allowlisting, webhooks, React, Next.js, Tailwind, Electron

AI / Agent Tooling

Model Context Protocol (MCP), LangChain, RAG / vector DBs (Qdrant), n8n, multi-provider LLM fallback, structured output, prompt design

Infrastructure

Git, Cloudflare (D1, KV, R2, Workers, Hyperdrive), Redis, SQLite, Render / Vercel, GitHub Actions CI, Vitest, Playwright, Termux / Android