KEVIN JOSH E. GODOY

FULL STACK ENGINEER

+63 916 595 7387 | kevingodoybiz@gmail.com
LinkedIn: linkedin.com/in/kevinjgodoy | Portfolio: fatedev.vercel.app | GitHub: github.com/FateDVCG

Professional Summary

Full Stack Engineer who owns AI products end to end, from architecture and an on-device model on a phone to the self-hosted Kubernetes cluster behind a SaaS. Built the AI chat and ran releases at **PLCcode.ai** through its public launch, freelance backend lead for **Bloopop**, and solo builder of **Diwa** (a self-hosted AI agent) and **Salagubang Wars** (a Roblox RPG). I do AI-assisted development by orchestrating coding agents in parallel: spec first, independent review, and tests and evals as the gate. I measure before I optimize, and I default to failing closed where money or data is on the line. **Cum Laude** BSIT graduate.

Experience

Full Stack Engineer | PLCcode.ai | *Mar 2026 – Present*
AI tools for controls engineers: browser-based PLC programming, version control, and live equipment access. Publicly launched Oct 2026.

- On a small team, built PLC Chat and its retrieval layer, led the move to self-hosted Kubernetes, and ran staging and production releases through public launch.
- Built PLC Chat, the AI copilot that reads and edits ladder logic, and grew it into a multi-stage agent (orchestrator, coder, server-side reviewer) that streams edits into the editor as it works, asks the user when unsure, and runs on a self-hosted model or cloud models via a live model switcher.
- Made the AI honest about its edits: every write reports a real outcome, "applied" only shows after the save lands, stale plans are refused, and read-only modes are enforced on the server, including a launch-week fix for a stale plan that erased rungs in production.
- Raised the self-hosted model's tool calling from 0/10 to 7/10 on a 10-case eval and reordered prompts for prefix-cache reuse (0% to 98% in a two-request test).
- Contributed to GitPLC, our git workflow for PLC files on self-hosted Gitea, and built "Update from Studio 5000," which re-imports a newer Studio project as one revertable commit that replaces the code, plus single sign-on across the app, Gitea, and remote access.
- Built the RAG layer behind chat (hybrid search lifted recall@10 from 0.68 to 0.90 on a synthetic eval set), then found it was frozen in time at import, so edits never reached the index, and rebuilt it as a per-file incremental index (a save was searchable in 65 s in a staging test).
- Cut rung loss on three real Studio 5000 projects from 90.8% to 0 (4,966 rungs), shipped L5X export that Studio accepts, and a from-scratch rung checker that cut false positives on a 3,137-rung project from 44,362 to 13.
- Shipped integrations end to end: remote desktop to plant machines with Windows and Linux agent installers, and admin impersonation with guardrails.
- Moved the platform onto self-hosted Kubernetes with Fleet GitOps, built release tooling (a fail-closed rollback gate and promote/revert console), and root-caused a silent 2-day control-plane outage that health checks missed.
- Helped build our self-hosted, segmented server infrastructure: moved the team to zero-trust access with per-person SSO, verified with 15 checks including 5 negative controls, split secrets per network zone, and set up backups with a tested restore.
- **Tech:** Next.js, TypeScript, Python, FastAPI, PostgreSQL, Redis, Gitea, Keycloak, Haystack, Qdrant, vLLM, Langfuse, Kubernetes, Fleet GitOps, Traefik, Prometheus, Grafana.

Full Stack Developer (Lead, Backend Focus) | Dalbit LLC | *Aug 2025 – Present (Freelance, on maintenance retainer since Oct 2026)*
Bloopop (bloopop.io): a creator-fan platform with a tokenized pay-per-message economy.

- Led backend and platform work over 14 months, owning the database, payments and subscriptions, messaging, and releases.
- Caught and fixed money bugs on a live platform: a double card charge on auto top-up and unkeyed Stripe payouts, now keyed from database state, with a guard test that fails if any Stripe create call lacks a key.
- Shipped delegated team-member accounts where fan-facing views show the creator, not the team member, routed through one authorization chokepoint and released as an 82-migration production deploy with an ordered runbook.
- Collapsed messaging reads from up to 6 database round trips into a single RPC with keyset pagination (under 7 ms p95 in a local database benchmark with 80k messages).
- Designed a gamification engine (XP, levels, badges, weekly decay) in SQL functions and Supabase Edge Functions.
- Built a tiered AI moderation pipeline (OpenAI scoring, Claude Haiku for gray-zone intent, AWS Rekognition for images) with shadow mode and an offline replay harness.
- **Tech:** Next.js, TypeScript, Supabase, PostgreSQL, Edge Functions (Deno), Stripe, OpenAI, Anthropic, AWS Rekognition, Vercel.

Junior Full Stack Developer | Unforgiving Games | *Aug 2025 – Mar 2026 (Freelance)*
NEVERQUEST: a real-time multiplayer MMORPG on Elixir and Phoenix LiveView.

- Contributor on a small team: built the first in-browser zone editor (PIXI.js, later rebuilt by the lead), the NPC dialogue system, and the quest system core, and fixed bugs in real-time movement sync (Elixir, Phoenix LiveView).

Full Stack Developer | Real-time Inventory Management System | *Jun 2025 – Jul 2025 (Freelance)*

- Built an offline-first inventory and sales system with real-time stock tracking and alerts, IndexedDB sync, JWT auth with RBAC, and AES-256 encrypted backups (React, Electron, SQLite).

Projects

Diwa | Self-hosted AI agent (solo) | *Jun 2026 – Present*

Always-on AI agent on my own Proxmox homelab that automates client billing, smart-home control, and live ops for my Roblox game, through Discord chat and in-room voice.

- Went from spec to a live Discord agent filling a real Google Docs invoice in two days, then guarded the billing path: idempotent weekly rollover, grace-window backfill, hard hours limits, and PTO handling that keeps untrusted text out of billing rows.
- Routed every tool call through a default-deny broker: an allowlist pinned to exact (operation, target) pairs, a tool-less LLM for untrusted text, and a SHA-256 hash-chained audit log. Natural-language commands only propose JSON intents that deterministic code resolves, risk-tiers, confirms, and can undo.
- Extended it into live ops for Salagubang Wars: playtester applications synced to the game's allowlist with etag-safe Roblox Open Cloud writes, redeem codes with owner approval, and a daily analytics and error digest, without widening its authority.
- Built in-room voice through a custom Home Assistant integration, then traced an 8-week voice outage blamed on hardware to an upstream mpv and PipeWire deadlock, lifting playback completion from 50% to 95% in my tests.
- Built "Bantay," monitoring and self-heal for the homelab (dead-man heartbeats, post-condition checks, drift watches), each added after a real silent failure.
- **Tech:** Python (asyncio), Discord, Home Assistant, Google Docs API, Roblox Open Cloud, Groq, Proxmox VE.

Salagubang Wars | Roblox action RPG (solo creator, in playtest) | *Sep 2026 – Present*

Filipino-folklore creature-collection action RPG about catching, raising, and battling salagubang beetles.

- Designed the game and led a pivot from turn-based to real-time action combat in about a week: a pure 20 Hz server simulation with server-side dodge i-frames, real-time PvP, and a 12-rung trainer league tuned with bot win-rate simulations.
- Run and orchestrate an agentic dev team (owned lanes, a coordinator agent, evidence-gated "done," a shared Studio lock), shipping 13 live builds as director and final reviewer, then ran a friends' playtest where the coordinator triaged 36 reports into 29 issues the same day.
- Kept the tests honest: a headless Lune test harness, mutation testing that caught 46 of 46 injected faults, and parallel runs that cut a full pass from 1,156 s to 469 s on one machine.
- Built data and monetization in strict Luau: idempotent DataStore transactions (fixing a double-sell race), one receipt handler, vouchers that grant nothing without a held voucher, and per-match entropy that closed a predictable-RNG exploit.
- Built the art pipeline: Claude Design concepts and a parametric Blender generator produce 13 beetle species, 9 mobs, and 9 bosses on one 44-bone rig, uploaded via Open Cloud and verified clip by clip against the source animation.
- **Tech:** Luau (strict), Rojo, Lune, Roblox Open Cloud, Blender (Python), Cloudflare Workers.

Urie.ai | Offline-first AI fitness app (solo) | *Mar 2026 – Jul 2026*

- Android fitness and nutrition app (Expo, React Native, SQLite) with an AI coach running a 3 GB Gemma model on the phone. Cut repeat-turn time-to-first-token from about 34 s to about 4.5 s with KV-cache reuse and a cached prompt prefix (measured on one mid-range phone).
- Kept workouts and GPS runs alive on aggressive Android OEMs (ColorOS) with a five-layer defense: auto-unloading the LLM in the background, correct foreground service types, adaptive GPS polling, a custom Kotlin wake-lock module, and an Expo config plugin.
- Privacy-first AI routing (on-device, then Groq, then Gemini) with a consent gate before cloud calls, plus a Supabase backend and admin dashboard. Reached Google Play internal testing.

Elizabeth.ai | Multi-tenant SaaS for Facebook ordering | *Feb 2026*

- Built a multi-tenant Next.js and Supabase platform (early-access waitlist) automating Facebook comment and Messenger ordering for Filipino micro-businesses, with RLS policies, Graph API webhooks, and Meta compliance callbacks.
- Built a hybrid Taglish order parser: a regex pre-pass handles clear orders at zero AI cost (share not yet measured), with a 3-provider AI cascade and per-provider circuit breakers for the rest.

Education

B.S. Information Technology (Major in Network Administration), Systems Plus College Foundation, May 2025 | **Cum Laude**. Dean's Lister (1st to 3rd year), Academic Excellence in Computer Networks. Capstone: PermitEase, a Django, Celery, and Redis platform automating university exam permits.

Skills

Languages & Game Dev: TypeScript, JavaScript, Python, SQL, Elixir, Kotlin, Luau (Roblox Studio, Rojo, Lune, Open Cloud), Blender
Frontend & Mobile: React 19, Next.js, React Native, Expo, Tailwind CSS, Phoenix LiveView, Electron
Backend & Data: Node.js, REST APIs, webhooks, OAuth/OIDC, Django, Celery, PostgreSQL, Supabase (RLS, Edge Functions), SQLite, Redis, Qdrant, Stripe
Infra & Ops: Proxmox VE, Kubernetes (RKE2), Fleet GitOps, Traefik, Prometheus, Grafana, Docker, GitHub Actions, Cloudflare
AI: Agentic team orchestration (Claude Code), RAG (Haystack, Qdrant), retrieval evals, vLLM, Langfuse, on-device LLMs, Claude, OpenAI, Gemini, Groq

Certifications

Cisco Networking Academy (CCNA Introduction to Networks, CCNA SRWE, CyberOps Associate, Cybersecurity Essentials, NDG Linux Essentials, Python) and Certiport (Python, Databases, Networking, HTML5), 2022 to 2024.