



Senior Full-Cycle Engineer — Build Your Startup (I build startups in 7–21 days)

Backend (Go/Node/Python) • Frontend (React/Next) • XR/Gamedev
(Unity/PlayCanvas) • AI • Web3 (Solidity)

🔗 GitHub: github.com/amad3i | 💬 Telegram: t.me/amad3i | ✉ Email: IBuildYourStartup@gmail.com

🌐 Location: Remote (Bangkok, Thailand, UTC +7) | 💰 USDT (TRC-20)

🎯 Positioning

I am “**Build Your Startup.**”

Senior Full-Cycle Engineer with **5+ years** of production experience. 🚀 I build products **from idea to release**: design, architecture, front/back, deploy.

BYS approach: I start on the day of the upfront payment, show a demo within **36–72 hours**, and ship a working MVP to production in **7–21 days**.

How I achieve speed:

- 🕒 I work **14–16 hours** a day.
- 🤖 I use **AI accelerators** (ChatGPT-5/Claude/DeepSeek, Cursor/Copilot).
- 📦 My own templates (auth/roles/CRUD/files/billing/notifications), open-source and code reuse.

Strength: A deep and broad profile across diverse technologies: from high-performance backends (Go/Node/Python) and complex 3D games & immersive interfaces (GameDev/XR/WebXR) to AI apps or dApps with smart contracts (Solidity).

🏆 Key Achievements

- 🚀 **10+ Startups Delivered from Scratch!**

BYS: take a “napkin” idea → in **7–21 days** roll out a finished product you can show to users and investors. **Demo — in 36–72 hours.**

- 🕶️ **20+ commercial XR/WebXR projects** for cross-platform and mobile browsers (iOS/Android), production quality and stable FPS.
 - 🛠️ **Full-stack production:** from Go/Node/Python backends with queues/caches/observability to a Next.js front-end and 3D/WebGL scenes in Unity/Playcanvas. **>80% of Projects are FULL-CYCLE from SCRATCH.**
 - 🗣️ **AI-PWA “Polyglot”:** a web app that translates videos into 15 languages. “Upload a video — get global language coverage.”
 - 🎮 **VR-Game “[SpectrumGames](#)”** — Photon multiplayer, released on Steam:
-

🧠 Tech Stack

Backend : Go (net/http, chi), Node.js (NestJS/Fastify), Python (FastAPI/Django); PostgreSQL, Docker, k8s, Redis, Kafka/RabbitMQ, gRPC/REST, WebSockets, OAuth2/OIDC, JWT.

Frontend : React/Next.js (SSR/ISR), Vue/Nuxt, TypeScript/JavaScript, PWA, Zustand, React Query, React Hook Form/Zod, Tailwind CSS, Webpack/Vite, Playwright/Cypress, Electron, Web Workers, Intersection Observer/Virtualization, Lazy Loading, Bundle Optimization.

Infra/DevOps/Observability : Docker/Compose, Kubernetes (k8s/k3s), Helm, ArgoCD (GitOps), Nginx (+Lua/edge), Envoy/Istio; CI/CD (GitHub Actions), Prometheus/VictoriaMetrics, Grafana/Loki/Tempo, OpenTelemetry, Sentry; TLS, HTTP/2/3 (QUIC), rate-limits, blue-green/canary deploys, rollbacks.




Web3 : Solidity (Hardhat, Foundry), smart contracts ERC-20/721/1155/4337, EIP-712, Oracles (Chainlink), DeFi (AMM, staking), security (Slither, fuzzing), integrations (Ethers.js/viem, The Graph, Alchemy/Infura, WalletConnect, MetaMask), L2 (Optimism, Arbitrum, zkSync).

GameDev & XR : Unity (C#, URP, Addressables/AssetBundles), AR/VR (SteamVR/Oculus SDK, AR Foundation, Photon PUN), WebXR/JS (Three.js, PlayCanvas, 8th Wall, Zappar, AR.js, Cocos Creator, Pixi.js); optimization/graphics (LOD, atlases, pooling, KTX2, meshopt, PBR, HLSL/Shader Graph, baked lightmaps/GI); platforms: Steam, Desktop, Oculus/Quest, Android/iOS, Web.





Work Experience

Founder/Full-Cycle Engineer | [BuildYourStartup](#) — Jan 2025 — Present

BYS: IndieHacking-as-a-Service. Turn-key startup creation.

-  **Process:** Research → design/Figma → architecture → front/back → payments/analytics → CI/CD/deploy → monitoring → Release.
-  **Speed:** 14–16 h/day, AI accelerators (ChatGPT-5/Claude/DeepSeek, Cursor/Copilot), my templates (auth/roles/CRUD/files/billing/notifications), open-source/reuse.
-  **Stack — per product:**
 - **Go + React/Next.js** — Go (net/http/chi or gRPC), PostgreSQL/Redis, S3, OAuth2/JWT, WebSockets; Next.js (SSR/ISR).
 - **Node.js + Next.js** — NestJS/Fastify, PostgreSQL (Prisma/TypeORM), Redis/BullMQ, Kafka/RabbitMQ; Next.js (SSR/ISR).
 - **Python + FastAPI + React/Next.js** — FastAPI, PostgreSQL/Redis, Celery/RQ, Pydantic; front-end in Next.js.
 -  **AI** (added to any track): integrations via OpenRouter, OpenAI API (GPT-4/5), Claude, Llama, DeepSeek — assistants, search/automation, moderation. No training LLMs from scratch.
 -  **XR/VR** (separate track): Unity (AR Foundation, SteamVR/Oculus, Photon, URP, Addressables), WebXR (8th Wall/Zappar/AR.js), Three.js/PlayCanvas.
 -  **dApp/Web3** (separate track): Solidity (Hardhat/Foundry), integrations Ethers/viem, WalletConnect/MetaMask, The Graph, L2 (Optimism/Arbitrum/zkSync). Front/back — predominantly TypeScript/Node.

Results:

-  Founded my first successful company.
-  Delivered >10 startups from scratch for clients.
-  Brought cycle time down to 7–21 days (depending on complexity) from zero to production.
-  Got rid of impostor syndrome.

Backend Platform Engineer (Golang) | [Aviasales](#) — Apr 2024 — Dec 2024

Stack: Go, PostgreSQL, Redis (Cluster/Sentinel), Nginx+Lua (edge), Docker, GitHub Actions (monorepo), Kafka/RabbitMQ, k8s.

- 🛠️ Designed and built HTTP/gRPC services (net/http, chi), clean layers, stable APIs.
- 📦 Infrastructure: libraries/SDKs, service generators and templates (repo structure, configs, middleware, health/metrics, logging/tracing), internal dev assistants.
- 🐘 PostgreSQL (database/sql/sqlx): transactions/isolation, migrations, indexes, query-plan profiling.
- 📧 Queues: consumers/producers for Kafka/RabbitMQ, retries, DLQ, “exactly-once” via outbox/idempotency.
- 🚀 Reliability/perf: worker pools, channels, context/timeout/cancel, backoff, rate-limiters, circuit-breakers.
- 💾 Cache patterns: cache-aside, invalidation/TTL, local coherency at point-of-use.
- 📊 Observability: OpenTelemetry (traces/metrics), structured logs with correlation, alerts on p95/error-rate.
- 🧪 Tests/environments: unit/integration, contract tests for APIs, local environments in Docker Compose.
- 🔍 Profiling: pprof, reduced allocations/GC pressure, optimized hot paths.
- 📁 Monorepo: migration, unified build/delivery practices, caches/artifacts, release/rollback standards.
- 🚚 Delivery: GitHub Actions → k8s (blue-green/canary), secrets/configs, env promotions, alerts/SLO.
- 🏗️ Architecture: RFC/ADR, service boundaries, DB/cache/broker contracts.







🎯 Results:

- ✅ Lead time “commit→prod” -50-70%, deploy frequency ×2-3, MTTR -40-60%, change-failure rate -20-30%.
- ✅ New service rollout per standard ×3-4 faster (on new teams), onboarding -40-50%.
- ✅ p95 -20-35% on hot routes thanks to edge (Nginx+Lua), caches and idempotency.
- ✅ CI pipeline time -35-50%, share of services on unified standard 80-90%;
- ✅ 100% of new ones with traces/metrics/health and “one-click” rollbacks with zero downtime.





🧠 Full-Stack & Blockchain Engineer | [AtomMining](#) / Enigma — Jan 2023 — Mar 2024

Stack: Go/Node.js, React/Next.js (TS, SSR/ISR), PostgreSQL, Redis, WebSockets, Three.js/WebGL, Solidity/Hardhat/Foundry, Ethers.js/viem, Alchemy/Infura, Docker, GitHub

Actions.



-  Backend (Go/Node): REST/WebSockets, clean layers, idempotency (Idempotency-Key, outbox), retries/backoff, worker pools/channels, rate-limiters/circuit-breakers; cache patterns (cache-aside, TTL/invalidation); DB schemas, migrations, indexes, query-plan profiling.
-  Frontend (Next.js/TS): dashboards and complex forms (React Hook Form + Zod), data layer (React Query), SSR/ISR, virtualized tables, asset upload/streaming, Tailwind; wallet integrations and real-time statuses.
-  Web3 (Solidity/integrations): ERC-20/721/1155 contracts, EIP-712 (signatures/permit), mint/burn/royalties, events; integrations Ethers.js/viem, WalletConnect/MetaMask; providers Alchemy/Infura (fallback/auto-switching).
-  Indexing/events: block/topic filters, reorg-safe processing (txHash+logIndex deduplication), The Graph (subgraphs) where justified; outbox → Kafka/RabbitMQ for further delivery.
-  WebGL/Three.js: GLTF/GLB pipeline, KTX2/Basis packing, meshopt, LOD/instancing, VRAM budgets, lazy loading/streaming; stable FPS on mobile.
-  Quality/security: unit/integration/contract tests (Jest/Playwright), Hardhat tests (chai/mocha), Foundry (fuzz/invariant), static analysis Slither; logging/metrics, tag-based releases (GitHub Actions), environments in Docker.

Results:

-  First load of heavy WebGL scenes -35-60%, median FPS +20-30% (KTX2/meshopt/LOD, lazy loads).
-  Errors/retries in background pipelines -50-70% (idempotency + queues/backoff).
-  Critical algorithms $O(n^2) \rightarrow O(n \log n)$ ($\times 10-20$ speedup) on indexers/aggregations.
-  Web3 integrations stable under spikes: reorg-safe processing, dedup via txHash/logIndex, providers with fallback → no missed/duplicate events.

XR / JS Engineer | [ARVR3D](#) – Oct 2022 – Dec 2022

Stack: Three.js/WebGL (GLTF/GLB, KTX2/BasisU, meshopt, LOD/instancing), WebXR, TypeScript, PlayCanvas; UI (HTML/CSS/Canvas), analytics (GA/Amplitude), CDN/HTTP cache

-  Scenes/mechanics: camera rigs (orbit/follow/gyroscope), raycaster/picking, state FSM, input system (tap/drag/pinch/tilt), hovers/aiming, hints/tutorials.
-  Visuals/shading: PBR materials, tone mapping, gamma/exposure, sky/IBL, simple custom shaders (GLSL) for effects (UV panners/scroll/Fresnel), shader

preloading/warming.

- ⚡ Optimization: asset pipeline (KTX2 for textures, meshopt for geometry), atlases, LOD/instancing, VRAM budgets; lazy-loading chunks, predictive prefetch; eliminating stutters (parallelizing decoders + off-main-thread).
- 🌐 Web integration: state/routing (SPA), events/analytics, deep-links; CDN caching, cache-busting, build versioning; crash/perf logging.
- 📱 Cross-browser: iOS Safari and Android Chrome — fixes for precision/NPOT/ASTC fallback; graceful degradation (low presets).

🎯 Results:

- ✅ First load -25-40%, stable FPS across a wide device matrix;
 - ✅ Reduced peak allocations → fewer GC pauses;
 - ✅ TTI on low-end devices -15-25%.
-

🌐 XR / WebXR Engineer | evr.systems — Mar 2022 — Oct 2022 • Remote

Stack: 8th Wall (image/world/face anchors), PlayCanvas, React (SPA shell), WebGL (GLTF/GLB, KTX2/BasisU, meshopt), TypeScript; analytics/events, CDN





- 📱 Mobile WebXR: full scenes with triggers/mechanics, HUD/UI, integration into React SPA (routing/state), events/analytics, AB flags.
- 🎯 Tracking/reliability: tuning XR8 (image targets, world anchors), calibration, stabilization; handling camera permissions, Safari limitations (autoplay, WebKit): fallbacks, recovery scenarios.
- ⚡ Perf/memory: asset repacking (KTX2/atlas), meshopt/LOD, VRAM/texture/video budgets; streaming/lazy loading; preloading on visibility events (IntersectionObserver).
- 🌈 Graphics: PBR pipeline, lightmaps/AO bake, simple FX (Fresnel/normal maps/UV panners) without heavy reflections (mobile limits).
- 🚚 Delivery: build versioning, CDN cache/invalidation, device/browser reports, fallback non-AR mode for weak/no-camera devices.

🎯 Results:




- ✅ Bundle size -20-35%, cold start -15-30%, predictable tracking and stable FPS on iOS/Android even on mobile networks;
 - ✅ Noticeably fewer camera/permission drop-offs.
-

Project: [SpectrumGames \(VR\)](#) — released on Steam

Stack: Unity (C#), URP, Photon PUN (PhotonView/RPC/RaiseEvent/Custom Properties), Steamworks/Oculus API, Addressables/AssetBundles, ScriptableObjects, Rider/Git




-  Netcode/sessions: matchmaking/lobbies, party/invites (Steam/Oculus), state serialization, snapshots, inter/extrapolation, lag compensation, interest management, sync update rate limiting.
-  Game systems: FSM, event bus, inventories/spawn, object pooling, controller interactions (VR-UX), secure RPCs.
-  Perf/memory: fewer draw calls (static/dynamic batching, instancing), LOD/occlusion, baked lightmaps/AO, VRAM budgets; allocation and GC-pressure control; addressable groups/catalogs, lazy streaming of assets.
-  CI/builds: SteamVR/Oculus build scripts, “one-click” releases/rollbacks, asset versioning; telemetry/crashes.

Results:




-  Released on Steam;
 -  Network artifacts -30-50%, stable FPS in key scenes;
 -  Cold scene start -20-35% via Addressables/pooling.
-

XR Engineer | [Instafilter](#) — Sep 2020 — May 2021 • Remote

Stack: PlayCanvas, 8th Wall, React, WebGL; asset pipelines (GLB/KTX2/atlasses), CDN







-  Full-cycle WebXR: scenes/logic, mobile constraints (size/VRAM/shaders), deploy/CDN, testing on real devices/networks; stability/perf reports.
-  Cross-browser: fixes for camera/permissions/network lag; fallback and recovery scenarios; degradation for low-end devices.
-  Integration: embedding into web products (events/analytics), configurable feature flags.

Results:



-  Stable production builds for mobile browsers on time;
-  Respected TTI/size budgets, predictable tracking, less “jelly” on iOS Safari.
-  ~5 delivered projects.

Freelance | Full-Stack / XR / Web3 / AI — 2019 — 2025 • Remote (in parallel with primary roles)




Stack: Go/Node/Python; React/Next; Unity/Three.js/WebXR; Solidity/Hardhat/Foundry; OpenAI/Claude/Llama/DeepSeek via OpenRouter; Postgres/Redis/S3; Docker/GitHub Actions/CDN.

-  **Web (product-grade):** SaaS, portals/communities, CRMs/dashboards, public APIs, PWA; auth (OAuth2/OIDC/JWT), payments, notifications, analytics; SSR/ISR, SEO/perf.
-  **Backend:** REST/gRPC, queues/workers, idempotency/outbox, cache patterns, profiling/observability; migrations/indexes/query plans.
-  **AI:** assistants/agents with actions, moderation, search/automation; integrations via OpenRouter (OpenAI/Claude/Llama/DeepSeek). No training LLMs from scratch.
-  **Web3:** dApps/wallets, ERC-20/721/1155 contracts, EIP-712, integrations Ethers/viem, WalletConnect/MetaMask, The Graph, L2 (Optimism/Arbitrum/zkSync).
-  **XR/Web:** interactive scenes in Three.js/WebXR and Unity/URP; GLTF/KTX2/meshopt pipeline, VRAM budgets, LOD/instancing; stable FPS on mobile.
-  **Delivery/quality:** tag-based releases, environments in Docker/Compose, CI (lint/test/build/deploy), CDN cache/invalidation, documentation/hand-off.

Education & Additional

-  **Higher education:** NSTU (Novosibirsk State Technical University)
-  **Languages:**
 - **Russian:** Native
 - **English:** C1 (Advanced) — reading technical documentation, handling correspondence and calls

Formats & Terms

-  **Mode: Remote.** I have a powerful workstation and stable internet. Experience working in distributed teams worldwide (GMT -8 to GMT +7).
-  **Time zone:** Flexible (UTC +7). I can align with your core working hours.
-  **Availability:**
 - **Contract/Project-based:** Full-time for the project duration.

- **Part-time:** Up to 20–30 hours per week (negotiable).
 - 📄 **Terms:**
 - 💵 **Payment:** Fixed per project or hourly (\$X/hour). **50% upfront** for all clients.
 - 🪙 **Crypto payments: Yes!** I accept only USDT (TRC-20), BTC, ETH.
-

📞 DM ME — I`ll Build Your Startup

💬 Telegram: <https://t.me/Amad3i>
(the fastest way to reach me)

🔗 GitHub: <https://github.com/amad3i>

✉ Email: IBuildYourStartup@gmail.com