

Jake Kim 김성주

I build financial software people trust with real money.

Product Engineer · Crypto-native Builder · AI-native Developer

jiovana.jake@gmail.com

github.com/howdyfrom2019

linkedin.com/in/sungjoo-kim-jake/

x.com/b_cryptojake

t.me/b_cryptojake

Yongin, Korea · Remote

Updated 2026-09-12

An AI-native product engineer who started in crypto and has built dApps, on-chain games, and an institutional trading terminal.

Right now I'm building OrderX, a multi-exchange crypto derivatives terminal. I use Claude and Codex as planner and reviewer, run the systems I design all the way to production, and like building big products with small teams.

Works

A terminal that starts from crypto derivatives and is heading toward an institutional trading desk. It handles crypto venues (Deribit, Hyperliquid, Lighter, Bybit, Binance) and traditional markets (OPRA US options, CME, IBKR) on one screen. Piloted with a large multistrat fund; \$10B notional processed and \$1M in revenue during private beta. I built the frontend from initial setup as the sole owner, pairing on the harder stretches, and designed its 18 feature domains.

IMPACT

- \$10B notional processed and \$1M revenue during pilot and private beta. Pilot run with a large multistrat fund.
- Unified order execution and multi-exchange support on the frontend, so traders place orders across accounts from one screen instead of hopping between exchanges.
- Steady-state browser memory from 3.8GB down to about 1GB, so an all-day trading session no longer needs reloads.
- AI workflow: edge cases that single-agent work used to leave behind are caught at the planner/reviewer stage, saving one to three-plus working days per feature. A small team finished the entire MVP flow. Localization automated through a Figma to i18n-agent pipeline.

PROJECTS

Order execution and multi-exchange support 2025.12 – 2026.09

Problem · Traders hold several accounts across venues like Deribit, Hyperliquid, and Bybit, and the same BTC option has a different symbol, decimals, and margin model on each. They used to hop between exchange screens to trade, and a slip sent money to the wrong account or instrument.

- One order form that selects several instruments and submits them together as a batch order. Each leg gets an anchor price so the preview shows an expected fill, and nothing leaves without passing validation.
- An account resolution rule that combines instrument, venue, and the user's choice to pick the destination account automatically, blocking wrong-account mistakes in the UI.
- Venue symbol formats (Deribit 'BTC-26DEC25-90000-C' vs OPRA 'IBIT_271217P00048000') handled as structured fields (underlying, expiry, strike, call/put) rather than strings, so adding a venue does not break order logic.
- Professional order types layered on top: TP/SL brackets, RFQ quotes, Spreader matrix orders, Arb builder, Hedge simulation, TWAP slicing, One-tap buy.
- Found that the UIContext passed to the AI for chat orders could not express batch orders or TP/SL, wrote the fix as an RFC, and changed the protocol with the backend team. Proposed '@account' tags to route each instrument to its own account.
- Showed that merging positions from several accounts into one table is unsafe because server messages carry no account identifier, and documented it as a backend change request.

Browser memory optimization (3.8GB → 1GB) 2026.04 – 2026.08

Problem · Real-time streams, widget state, TradingView iframes, and query caches accumulated until Chrome sat at 3.8GB, slowing the UI and eventually killing the tab.

- Split orderbook and position table updates down to row and symbol granularity to remove redundant re-renders and allocations.
- Replaced AG Grid with an in-house DataTable with row memoization, virtualization, and clean unsubscribe paths.
- Three-stage runtime recovery: partial resubscribe, then cache invalidation, then reload when returning from a background tab. Reloads are blocked while an order is being entered.

Translate-based in-house grid system 2026.06 – 2026.09

Problem · react-grid-layout recalculated layout and re-rendered React on every mousemove during resize, saturating the CPU.

- dnd-kit plus imperative DOM transform (translate) updates so React is out of the loop while dragging.
- Grid-to-pixel math isolated in pure functions and pinned with tests.
- Widget stretch and shareable page layouts with auto-generated OG images.

AI workflow adoption 2026.01 – 2026.09

Problem · A small team had to push MVP features across 18 domains at once. Single-model, single-agent work shipped fine on day one but left edge cases behind, so the same feature had to be reopened days or weeks later.

- Repository conventions, architecture, and domain rules fixed through context injection so any agent works to the same standard.
- Claude as planner, Codex as reviewer (planner / reviewer): design, implementation, and review separated, with plan/design documents and gap analysis as the audit trail.
- react-doctor cleanups run as a routine so debt from feature integration does not accumulate.
- Outcome: coverage over speed. Edge cases caught at review removed one to three-plus working days of rework per feature.

Figma MCP + i18n agent pipeline 2026.06 – 2026.09

Problem · Design changes and translated copy were applied by hand, so every release shipped with omissions and mismatches.

- Figma MCP reads the design source directly to apply component changes.
- An i18n agent finds missing keys, translates and registers them, and syncs with Lokalise. Localization runs on next-intl.

Chart library migration (lightweight-charts → TradingView) 2026.03 – 2026.09

Problem · The backend speaks Protobuf over HTTP plus WebSocket rather than a standard feed, so TradingView's stock Datafeed did not fit.

- Provider-pattern custom Datafeed. Position and execution overlays, live quote push, per-account overlays.
- Custom indicators (OI profile, visible-range profile, cross-exchange studies) and a range-preset toolbar. Analytics widgets such as FedWatch, an economic calendar, and SVI parameters.

Frontend: Next.js 15, React 19, TypeScript, Bun, Tailwind v4, Radix UI, Jotai, TanStack Query, React Hook Form, Zod, dnd-kit, framer-motion, react-virtuoso · Realtime: Protocol Buffers (ts-proto) over HTTP + WebSocket · Chart: TradingView Charting Library v31, TitanCharts, ECharts · AI: Vercel AI SDK, assistant-ui, ElevenLabs · Tooling: Playwright, Bun test, Lokalise, PostHog, OpenTelemetry, AWS Amplify

The Waifu Sweeper team under Yield Guild Games (YGG). Started as Princess Sweeper, later rebranded to Waifu Sweeper / Playsweeper. I owned the entire web client and contract stack and shipped design changes (reward multipliers, season leaderboards, Gold Mine mode, revive mode) on a weekly cadence. The service has ended; the link is a build on the original frontend core logic.

IMPACT

- 19,939 payments through the shop contract with zero failures, about \$1.5M in total: 72,500 USDC, 25,000 USDT, 104.65 ETH, 50,616,446 YGG. Shown at 2026-09-12 prices; hover for the live conversion.
- Deployed to Abstract mainnet after an external audit (2025.12). 94 Hardhat tests cover purchases, the oracle, NFTs, and the session tracker.
- Upgradeable proxies allowed token-economy changes (reward multipliers, season leaderboards, Gold Mine, revive mode) without redeloys, supporting repurchase and retention.
- Owned the whole client and contract stack solo. Bot traffic blocked with Cloudflare Turnstile.

PROJECTS

Operating method: upgradeable proxies and context injection 2025.09 – 2026.07

Problem · The token economy (reward multipliers, repurchase incentives, seasonal events) changed weekly, and every touch of the payment contract put user funds at risk.

- ShopManager, NFT, and PriceFeed run behind UUPS proxies with a storage layout review fixed into the deploy checklist. New capabilities such as the session tracker live in separate contracts rather than new proxy slots.
- Game rules, entities, and reward logic specified through context injection so in-game logic changes ship quickly and consistently.
- Result: 19,939 payments with no failures and weekly design releases.

AGW session-key game sessions 2026.05 – 2026.06

Problem · Repeated actions such as opening tiles, defeating monsters, finding items, and reviving had to be recorded on-chain without a wallet signature every time.

- A hook that creates, stores, and restores a 24-hour session key with a fee limit and per-contract, per-selector call policies.
- A WaifuGameSessionTracker contract that validates ordering with a per-player session id and action nonce.
- Passed the tracker address as an argument instead of adding storage to the already-deployed ShopManager proxy (storage layout safety), and documented why.

NFT gacha and energy-pack shop 2025.09 – 2025.12

- ShopManager (UUPS): ETH/USDC/USDT/YGG payments with explicit per-token decimals, EIP-712 signature verification with nonce replay protection, slippage handling.
- PriceFeed (UUPS): Pyth ETH/USD feed, confidence checks, cent-denominated pack pricing.
- WaifuSweeperNFT (UUPS ERC721): batch minting up to 30, paginated tokensOfOwner to stay under gas limits.
- Client: bundles Pyth price updates into the transaction, NFT as a payment method, YGG discounts.

Live-ops tooling and seasons 2026.01 – 2026.07

- Season and Gold Mine leaderboards, reward dashboard, revive mode, referral-code validation.
- Maintenance page with a service schedule, per-feature and date-based flags.
- GA4 event design, Twitter Pixel, SEO/OG, BGM and sound effects, image preloading.

Frontend: React 19, TypeScript, Vite 7, Tailwind v4, Radix UI, Jotai, TanStack Query, Framer Motion, react-router 7 · Web3: Wagmi, Viem, Ethers v6, RainbowKit, Abstract Global Wallet (sessions) · Contracts: Solidity, Hardhat, OpenZeppelin Upgradeable v5, Pyth SDK, TypeChain · Infra: Vercel, Cloudflare Turnstile, GA4

Built the web and app platform for Gomble, the Web3 division of Korea's top mobile game studio, 111percent. Spent a month remote before joining, handing over a token-airdrop project tied to exchange listings (Binance Alpha, Coinone, Bitget), then took over as the sole frontend engineer handling business requirements.

IMPACT

- Zero dev-side CS issues on the token airdrop (TGE) claim page.
- Designed a dApp that kept wallet connection and transaction delivery stable through a TGE with more than 10,000 daily spike requests.

PROJECTS

TGE page (token airdrop) 2025.03 – 2025.04

Problem · A claim page timed to an exchange listing draws a traffic spike that a single RPC node can't absorb without breaking wallet connections and transaction delivery, and the Linear Cliff Vesting schedule needed to be legible to non-technical users.

- Distributed requests across 10 public JSON-RPC node addresses.
- Applied UX writing in Wagmi that anticipates chain/wallet switches, keeping wallet connection and transaction delivery stable.
- Built a readable time-display UI for the Linear Cliff Vesting schedule.

iGaming project 2025.04 – 2025.06

- Built a casual clicker-style game in React.js.
- Designed and built a blockchain game combining a casual game with outcome betting, as a Solidity smart contract.
- Wired the website to the contract; designed a staking-plus-Web3-game reward structure contract.

Reward app 2025.06 – 2025.07

Problem · A native app's store review and update cadence isn't fast enough on its own; a web channel that iterates without review was needed alongside it.

- Built an ad-based reward platform with Expo (React Native) and AdMob.
- Built a matching PWA with React and Vite-pwa to iterate quickly without store review.

Other projects

- Built an outsourced penguin ice-breaking web game.
- Built an NFT staking/unstaking website.
- Frontend for an NFT verification and staking community that also takes NFT game commissions.
- Built the \$GM token claim site for a token listed on Binance Alpha, Coinone, and Bitget.

Frontend: React, TypeScript, Vite, Bun, Vite-pwa, React Native, Expo · Web3: Viem, Ethers, Wagmi, WalletConnect, Hardhat, Solidity · Infra: AWS S3 + Route53, Vercel

GXC · xCBT Blockchain Engineer

2024.10 – 2025.04

xCBT, a spin-off team set up by GXC Inc., built with blockchain specialists from Round Ventures and Pentachora Labs. An early blockchain-only mission platform: users who buy an NFT earn points over a two-week mission period, and top-ranked players win USDT-based rewards. Joined at the team-building stage and owned the website plus NFT smart contracts and purchase/burn verification logic, paired with one teammate who owned backend and infra.

IMPACT

- 10,000+ USDT in total payouts through the Xociety game mission.

PROJECTS

xCBT platform development 2025.01 – 2025.04

- Turborepo monorepo covering both the admin page and the client app.
- USDT (ERC-20) based NFT minting, with Thirdweb handling whitelist-gated purchases.
- OAuth for Twitter, Epic Games, and Discord in Next.js. Worked out a Twitter OAuth approach next-auth didn't cover out of the box.
- Raffle screen ahead of ticket purchase; reward structure paying USDT, next-season tickets, and xCBT tokens by leaderboard rank.

Frontend: TypeScript, Next.js, Tailwind CSS, Framer Motion · Web3: Solidity, Hardhat, Viem, Thirdweb, Abstract Global Wallet

Frontend for Yooldo, a blockchain gaming platform that helps Web2 games transition to Web3, plus game onboarding support. Yooldo is Linea's largest GameFi platform; its ESPORTS token is listed on Binance Alpha, Bitget, MEXC, and other major exchanges.

IMPACT

- User count up 20,000%+ after the rebrand (2023.07–11). 1.38M+ monthly transactions, \$152.73K+ monthly volume.
- Ranked #26 overall on DappRadar's gaming sector and #1 on Linea as of April 2024.
- The Yooldo x Linea Park cross-chain event brought in about 750,000 new users. Resolved traffic-driven issues including Vercel image-optimization costs and bandwidth overruns.
- Sold out an 8,888-piece collection in a week, grossing 300M+ KRW. Ran NFT campaigns with chains including Kroma, Astar, and Manta Network.
- Exchanged written English bug and improvement reports with MetaMask (Consensys), Unisat Wallet, and HaechiLabs' Facewallet team.

PROJECTS

Yooldo x Linea Park (cross-chain campaign) 2024.02 – 2024.04

Problem · The campaign needed to funnel a multi-chain audience into one specific chain, and the resulting 750,000 new users triggered heavy traffic problems: bandwidth overruns on image/video assets, spiking Vercel image-optimization costs, and edge cases in verifying deleted accounts' game IDs.

- Tracked multi-chain participation with GA metrics and responded to community issues in real time.
- Moved transaction execution from server-side to client-side plus receipt verification to cut load.
- Relaxed game-ID verification for deleted accounts and supported post-launch CS.

Yooldo rebrand 2023.07 – 2023.11

Problem · Legacy code and an aging stack (styled-components, localStorage token storage) were holding back a brand refresh.

- Split atomic components into an npm library (catze-ui); separated the asset-heavy landing page from the game and on-chain interaction pages.
- Migrated styled-components to Tailwind CSS, adopted SSR, moved access tokens from localStorage to cookies with an expiry timer.
- Built and maintained Jury DAO, a decentralized system for reporting and filtering abusive in-game users.

Game login and security 2023.08 – 2024.02

Problem · Yooldo's own published games (Trouble Punk, RPD, and others) needed secure login and detection of tampered game clients.

- Introduced universal/deep-link login (yooldo://game/troublepunk) for improved security.
- Tracked game asset S3 hashes through the back office to detect tampered launchers.
- Supported Play Store market-link entry for Android-only titles.

Back-office development and operations 2023.04 – 2024.08

- Cut the operational kill switch time by 10+ minutes. User/site interaction tracking, batch email sends, scheduled project updates.
- Shipped operational tools such as Jury DAO and the BRC-20 bridge through the back office; scheduled GitHub PR merges in the CI/CD pipeline.
- Contract, user, and on-chain interaction log tracking; set policy for wallet keys and infra credentials.

Platform frontend & transaction verification

- Development and operations of the blockchain game web platform, handling on-chain interaction events.
- Transaction-receipt analysis to verify in-game asset acquisition.
- Built and ran a Bitcoin bridge (BRC-20).

Frontend: TypeScript, Next.js, TanStack Query, Framer Motion, Storybook · Backend: Node.js, Nest.js, Prisma · Web3: Viem, Wagmi · Infra: Vercel, GitHub Actions

Worked at Bluepoint Partners, an accelerator for seed and pre-A tech startups, six months as an intern followed by two months on contract. Built the firm's in-house investment management tool.

IMPACT

- Automated investment-contract preparation, generating documents in under a second instead of three-plus days.
- Consolidated deal data that had been scattered across spreadsheets into time-series and location-based charts.
- Built a Slack-integrated alert for investment-proposal inflow from the homepage.

PROJECTS

Startup investment management tool

- Improved deal-sourcing UX through interviews with investment associates.
- Time-series and location-based conversion visualizations with recharts.
- Automated contract-document generation, cutting a three-plus-day task to under a second.
- Slack alert integration for investment-proposal inflow from the homepage.

Frontend: TypeScript, React, Jest, recharts, styled-components · Infra: AWS S3 + Route53

Built FOUND, a GPS-based lost-and-found reward app, at location-services startup Tasting Venture. Operated in the Yongin area; the service itself ran until February 2020.

IMPACT

- 5,000 downloads on the Play Store.

PROJECTS

FOUND

- Found-item reporting flow with Android Studio (Java), Retrofit, and Glide.
- Hybrid app built with Angular and the Ionic Framework.
- KakaoTalk chatbot for lost-item reports with a Flask/MongoDB webhook server.

Android: Java, Retrofit2, Glide · Backend: Python, Flask, MongoDB · Hybrid: Angular, Ionic Framework

Education & Credentials

Dankook University	B.S. Electrical & Electronic Engineering · 3.92/4.5	2016–2023
Engineer Information Processing	HRD Korea	2020
Patent · Bus fare payment method and server	KIPO	2020
Startup Idea Hackathon, Grand Prize	Gyeonggi CCEI	2018
Alchemy University Ethereum Bootcamp	Alchemy	2024

Glossary

derivatives

Contracts whose value derives from an underlying asset such as BTC. Options, futures, and perpetual futures are the main kinds; they carry more leverage and risk than spot.

multistrat fund

A hedge fund running several strategies at once (options, arbitrage, hedging). The kind of client OrderX piloted with.

multi-exchange

Handling crypto venues (Deribit, Hyperliquid, Lighter, Bybit, Binance) and traditional markets (OPRA US options, CME, IBKR) on one screen. Every venue has its own symbol format, decimals, and margin model, which makes unification hard.

batch order

Submitting orders for several instruments as one bundle, for example buying three call options at once for a spread strategy.

trading desk

The team and tooling an institution uses to execute orders across venues and manage risk in one place. OrderX aims to turn that job into software.

notional

The face value of contracts traded, not cash that changed hands. \$10B notional means ten billion dollars of contracts passed through the terminal.

order form

The screen where quantity, price, order type (limit, market), and conditions are entered and sent to the exchange. Mistakes cost money immediately, so state handling and validation matter most here.

anchor price

The reference price each leg of a batch order starts from (last trade, mid-quote, or a level the user clicked). It backs the estimated fill price shown in the order preview.

account resolution

The rule that decides which account an order goes to when a user holds several accounts on the same venue, combining instrument, venue, and the user's choice.

RFQ

Request for Quote. Instead of hitting the orderbook with size, ask market makers for a price on your terms and fill at the best quote. Common in institutional trading.

Arb builder

Arbitrage builder. When the same asset prices differently across venues, design and execute the buy-low, sell-high pair in one step.

TWAP

Time-Weighted Average Price. An algorithmic order that slices a large order over time to reduce market impact.

UIContext

The data structure that tells the AI what the user is currently looking at (selected instrument, account, order-form state) when placing orders through chat. If it is thin, the AI trades the wrong instrument.

orderbook

A live table of resting buy and sell orders at each price. It changes tens to hundreds of times per second, making it the hardest widget to render efficiently.

runtime recovery

The procedure that revives stale connections and data when a tab comes back from the background: light (resubscribe), medium (invalidate cache), heavy (reload), escalating only as needed.

translate-based grid

Moving widgets by changing the DOM's transform: translate value directly while dragging, bypassing React. No re-layout, so it stays at 60fps.

planner / reviewer

Splitting roles so Claude plans and implements while Codex reviews independently, rather than one AI doing everything. Different models catch each other's mistakes.

Figma MCP

A Model Context Protocol connection that lets an AI agent read Figma files directly, so design changes flow into code without manual handoff.

Protobuf

Protocol Buffers, a binary data format smaller and faster than JSON. OrderX uses it over both HTTP and WebSocket, generating types from .proto files.

Datafeed

The adapter that tells the TradingView charting library how to fetch and stream candles for a symbol. Non-standard backends need a custom one.

FedWatch

Market-implied probabilities of the next Fed rate decision, derived from futures prices. Traders check it before macro events.

Abstract chain

A consumer-focused Ethereum L2. Ships with a built-in wallet (Abstract Global Wallet) and session keys, which suits game UX.

Turnstile

Cloudflare's CAPTCHA alternative. Filters bots from browser signals without making users solve puzzles.

TP/SL

Take Profit / Stop Loss. Standing orders that lock in gains at a target price and cut losses at a limit price automatically.

Spreader

A widget for composing spread trades as a matrix: buy one instrument and sell another at the same time to capture the price difference, such as two futures with different expiries.

Hedge

Offsetting the risk of a position with an opposite trade. The hedge simulation shows how overall risk changes before you add a contract.

One-tap buy / Quick order

A fast-order UI that sends a preset size with a single click. Built for speed, so it needs guards against mis-clicks.

RFC

Request for Comments. A written proposal reviewed by other teams (here, backend) before a protocol change, so agreement comes before code.

position table

A table of current holdings with P&L and risk metrics (delta, gamma, and so on) per account. Every price tick recomputes P&L for every row.

react-grid-layout

A React library for drag-and-resize dashboards. It updates React state on every mouse move, which burns CPU on heavy screens.

context injection

Automatically feeding an AI coding agent the repository's rules, architecture, and domain knowledge every time (CLAUDE.md, AGENTS.md, skill docs). Any agent, any operator, same standard.

react-doctor

A diagnostic tool that finds unnecessary re-renders, hook misuse, and performance issues in React code. Run routinely after feature merges to clear debt.

i18n

Internationalization. UI copy lives in per-language dictionaries; an agent finds missing keys and fills them.

WebSocket

A persistent connection the server pushes data through. Prices, fills, and balances arrive over it in real time.

OI profile

Open Interest profile. Draws how many contracts are open at each price level beside the chart, revealing where large positions sit.

SVI

Stochastic Volatility Inspired. A formula that describes the implied-volatility curve of options with a few parameters, used by options desks to spot mispricing.

UUPS proxy

An upgrade pattern that swaps a deployed contract's logic while keeping its address and stored data. Essential when a payment contract changes weekly.

storage layout

The order variables occupy storage in an upgradeable contract. A mismatch in a new version overwrites existing user data, so it is reviewed before every deploy.

session key

A temporary key the wallet owner authorizes for specific contract functions, within a fee limit, for 24 hours. In-game actions no longer need a wallet popup each time.

Pyth oracle

A service that publishes live prices such as ETH/USD on-chain. It is the exchange-rate source when a dollar-priced item is paid in ETH.

EIP-712

An Ethereum standard for signing human-readable structured data. The server signs 'this item at this price is allowed', the contract verifies it, and a nonce blocks replay.