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

SunLabs AG · OrderX FE Part Lead 2025.10 – Now

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

YGG · Waifu Sweeper Frontend & Contract Engineer 2025.08 – 2026.07

The Waifu Sweeper team under Yield Guild Games (YGG). Started as Princess Sweeper, later rebranded to Waifu Sweeper / Playpsweeper. 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 redeploys, 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

111percent · Gomble Frontend Engineer 2025.04 –

Built the web and app platform for Gomble, the Web3 division of Korean game studio 111percent.

IMPACT

- Designed a dApp that kept wallet connection and transaction delivery stable through a TGE with more than 10,000 daily spike requests.

PROJECTS

Ad-reward app

- Native mobile app (React Native, Expo) and PWA.

NFT community site

- NFT verification, staking, and a request flow for commissioning NFT games.

Betting game contracts

- Upgradeable smart contracts using the proxy pattern.

$GM token claim site

- 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

Smart contract development and verification plus frontend integration for xCBT, a platform where players preview monthly blockchain game releases and compete for rewards.

IMPACT

- Built the social-verification mission platform that generated 10K+ USDT.

PROJECTS

Social-verification mission platform

- Mission features.
- Wrote and updated NFT-related smart contracts.
- Client and back-office monorepo (pnpm, Turborepo).

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

CatzeLabs · Yooldo Frontend Engineer 2023.04 – 2024.10

Frontend for Yooldo, a blockchain gaming platform that helps Web2 games transition to Web3, plus game onboarding support.

IMPACT

- Onboarded more than 900,000 users onto the platform.
- Contributed to up to $150k monthly revenue through NFT purchase and verification UX.
- Helped secure investment from Consensys, the company behind MetaMask.
- Built eight or more in-house NFT mint pages grossing over 300M KRW.

PROJECTS

Yooldo platform frontend

- Platform development, operations, and on-chain event handling.
- Transaction-receipt analysis to verify in-game asset acquisition.
- Back-office features and a shared UI component library published to npm.

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

Bluepoint Partners Frontend Engineer 2021.07 – 2022.03

Built the in-house investment management tool for an accelerator focused on seed and pre-A tech startups.

IMPACT

- Automated investment-contract preparation that used to take more than three days.
- Consolidated portfolio tracking that had been scattered across teams.

PROJECTS

Startup investment management tool

- Improved deal-sourcing UX through interviews with investment associates.
- Time-series and conversion visualizations with recharts.

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

Tasting Venture · FOUND Founder 2018.12 – 2019.06

Built and operated FOUND, a GPS-based lost-and-found reward app, in the Yongin area.

IMPACT

- 5,000 downloads on the Play Store.

PROJECTS

FOUND

- Found-item reporting flow (Android, Java).
- KakaoTalk chatbot for lost-item reports with a Flask/MongoDB webhook server.

Android: Java, Retrofit2, Glide · Backend: Python, Flask, MongoDB

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.

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.

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.

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

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.

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.

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.

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.

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.