Chengwei Wang

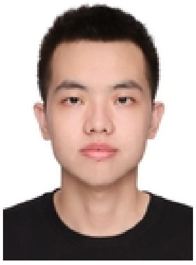
Email: 892630kux@gmail.com

Phone : +86 18362392398

GitHub: https://github.com/wangchengwei0122

LinkedIn: https://www.linkedin.com/in/chengwei-wang-060852381

Telegram: https://t.me/chengwei0122



Professional Skills

- Next.js + TypeScript; wagmi/viem for wallet UX, contract calls, and on-chain reads.
- UI architecture & DX: reusable components/hooks, typed domain models, and clean state separation for maintainability
- Frontend engineering: monorepo (pnpm/workspaces), build/release workflow, CI/CD; strong delivery reliability.
- Performance optimization: first-load/rendering, caching & compression, lazy loading/code splitting, long-list tuning.
- Web3/DeFi fundamentals: EVM/Solidity basics, allowances/approvals, tx lifecycle; slippage/price impact/fees

Projects

DEX Frontend | Uniswap-style Decentralized Exchange UI

Github: https://github.com/wangchengwei0122/dex-frontend

Live Demo: https://dex-frontend-bice.vercel.app/

Tech stack: Next.js (App Router) / TypeScript / Tailwind / RainbowKit / wagmi / viem / EVM

- Built a Uniswap-style DEX frontend supporting swap flows, network switching, and token selection
- Integrated wallet connection and multi-wallet compatibility via RainbowKit, with network switching and graceful fallbacks
- Implemented wallet + on-chain reads: balances, allowances, approvals (Approve), and the end-to-end swap transaction flow
- Shipped core DeFi UI logic: slippage tolerance, price impact, minimum received, and fee calculations
- Improved safety & usability with approval risk prompts, address validation, and clear confirmation/failure handling
- Decoupled frontend and contract logic by wrapping on-chain read/write operations into small adapter hooks; UI consumes typed domain models instead of raw RPC responses
- Ensured engineering quality with strict TypeScript end-to-end typing plus lint/format to reduce runtime risks in contract interactions

Fundr — Web3 Crowdfunding dApp

Github: https://github.com/wangchengwei0122/fundr

Tech stack: Next.js / TypeScript / wagmi / viem / Solidity / Node.js (Fastify) / Drizzle ORM / PostgreSQL / Cloudflare Workers (KV) / Indexer

- Built a complete Web3 crowdfunding dApp supporting campaign creation, on-chain contributions, role-based withdrawals, and transparent fund progress tracking
- Modules: Smart Contracts (Solidity/Foundry, ABI sync) | Indexer (listen to events → PostgreSQL) | API (Node.js + Fastify + Drizzle, single data source) | Edge (Cloudflare Workers + KV, proxy & caching) | Web (Next.js UI, wallet + UX)
- Adopted an indexer-first architecture: the indexer listens to contract events and writes to PostgreSQL to form a queryable off-chain index

- Implemented a Fastify REST API wrapping DB queries (filtering/pagination/aggregation) as the single source of truth (SSOT) for frontend and edge cache
- Added Cloudflare Edge Worker + KV to proxy API requests and implement read-through caching, reducing direct backend pressure and improving latency
- Frontend (Next.js App Router) never reads blockchain data directly: all reads go through API/Edge; wallet + tx flows are handled via wagmi/viem (create/contribute/withdraw)
- Improved indexing stability and consistency using batching, retry/backoff, and checkpoint mechanisms to keep DB state aligned with on-chain events

Work Experience

Qichacha Technology Co., Ltd. — Senior Front-End Engineer 2019.04 – Present

Responsible for frontend architecture and core delivery for the Open Platform, serving millions of enterprise and individual users, continuously shipping scalable and high-performance web applications in cross-functional collaboration

- Built SSR optimization with Nuxt.js + Node.js, improving initial load experience and SEO performance
- Integrated AI capabilities and delivered intelligent Q&A, content summarization, and recommendation features to boost enterprise productivity
- Drove department-wide monorepo adoption and engineering standards to improve collaboration efficiency and code reuse
- Built a modular UI component library and promoted cross-team component standardization
- Refactored multi-client codebases (Web, H5, in-app WebView / JSBridge) to improve maintainability and performance

Tongcheng Travel — Front-End Engineer 2018.05 – 2019.04

Built ticketing/booking/payment features across Web, Hybrid App, and WeChat Mini Programs.

- Added a Node.js BFF for API aggregation to reduce latency and improve stability.
- Modularized core flows with Vuex to improve maintainability and scalability
- Used dynamic import/code splitting to cut bundle size ~30% and speed up first load.
- Collaborated closely with UX/design to refine responsive layout and accessibility details

Jiangsu Weisheng Network Technology Co., Ltd. — Front-End Engineer 2017.04 – 2018.05

Responsible for Hybrid App and mini-program frontend development.

- Migrated legacy jQuery projects to Vue, improving modularity and maintainability
- Integrated native capabilities via JSBridge (maps, QR scan, payments, etc.)
- Led company-wide UI refactoring and componentization, promoting a unified component architecture
- Authored internal coding standards and documentation to improve collaboration quality

Education

Wuxi University of Technology– Automation Technology for ProductionProcesses 2012.09 – 2015.06

Additional Information

- English: Proficient reading (docs/RFCs/GitHub issues), good writing (PR/README), good listening (can follow work discussions);
- Continuous learner with strong adaptability: keep up with new frameworks/best practices and apply them pragmatically in real projects
- AI-assisted development (e.g., Cursor / Claude Code): ideation, code generation, refactoring, and debugging to improve delivery speed and quality
-