

# 業務経歴書

## 基本情報

名前	石井 希
居住地	東京都
GitHub	<a href="https://github.com/nozomiishii">https://github.com/nozomiishii</a>

## 概要

- フロントエンド、バックエンド、モバイル開発において全く0からの新規立ち上げフェーズから保守運用まで幅広く経験してきました
- プロジェクトリーダーとしてチームマネジメントやコードレビュー、技術指導、プロジェクトの進捗なども携わらせていただきました
- ドキュメントは読まれにくい、人的ミスは避けられないという実体験のもと、なるべく文章ではなくコードで開発環境や仕組みを作ること誰が書いても同じコードスタイルになるように工夫したりチームが快適に開発できる状態をつくるのが好きです
- 最近はRustでの組み込み開発にも関心が高く、将来は電子工作といった低レイヤーから、3Dモデリングなどのデザインまで幅広くコンピュータについて一生楽しみ学び続けてたと思います

## 技術

- 現在も実装できる技術だけを列挙しています。

### 言語

TypeScript | Bash | Rust

### フレームワーク

#### Frontend

Next.js | React | Redux | Redux Toolkit | Recoil | SWR | TanStack Query | TanStack Table | React Hook Form | React Icons | TailwindCSS

#### アニメーション

GSAP | Three.js | p5.js | Framer Motion

#### Backend

NestJS | Express | Socket.IO | OpenAPI | Prisma | TypeORM

#### Mobile

React Native

#### Test

Cypress | Jest | Testing Library | MSW | Supertest

### RDB/NoSQL

PostgreSQL | Redis

## GCP

Cloud Run | Cloud Storage | Stackdriver Monitoring | Cloud Logging | Container Registry | Cloud Functions | Firebase

## Monorepo

Yarn Workspaces | TurboRepo | Nx

## その他

GitHub Actions | Lefthook | Docker | Terraform | Sentry | Heroku | Vercel | SendGrid | Pusher | TinyMCE | Stripe | Agora | Figma

## 強みが発揮されやすい業務

- React Component設計開発
- TailwindCSSの導入、style開発
- LinterやFormatterの導入
- Git Hooksの導入
- アニメーション実装
- Monorepo開発
- StorybookでのUIカタログ作成
- 単体テストや統合テストの導入
- CIサービスの導入
- デプロイの自動化
- GCPでのアプリケーション導入、運用
- 監視ダッシュボードの導入
- 多言語化
- アクセシビリティに配慮した設計
- 通話サービスの導入
- ソケット通信の実装

## 主な業務経歴

### NFTマーケットの新規開発（2022年 3月 ～ 2022年 7月）

#### 【担当業務】フロントエンド開発/API設計/チームリーダー

- Next.js、Reactを使用したSPA開発
- TurboRepoを使用したMono Repo開発
- OpenAPIを使用したAPI設計
- TailwindCSSの導入
- SWR、Recoilを使用した状態管理
- チームリーダーとして他のプロジェクトとの進捗共有、API設計のレビュー

新規プロジェクト立ち上げから参画し、初めてのリモート、チームの半数が業務委託といった開発でした。Recoil, SWR, TurboRepoなど今まで使用したことがない技術もすばやくキャッチアップし実装することができました。開発開始のスク립トを少なくまとめることでチームの入れ替わりや新しいエンジニアが入ってきてもスムーズに開発が開始できるように工夫をしました。

また、OpenAPIをスキーマファーストで設計することでバックエンド、フロントエンドが同時に開発できるようにしたり、

LintやFormatterを作り込みCIで導入することで、のちの運用フェーズや新規機能案件の際に担当が変わってもコード内容が変わらないような開発環境を整えました。

## 通話サービス系スタートアップの新規立ち上げ（2021年 4月 ～ 2022年 1月）

### 【担当業務】アーキテクチャ設計/API開発/DevOps基盤構築/チームマネジメント/フロントエンド開発

- Nxを使用したMono Repo開発
- NestJS、Postgres、OpenAPIを使用したSchema FirstでのRestAPI設計、開発
- Jest、Cypress、Docker、GitHub Actions、Cloud Runを使用したCI/CD構築
- プロジェクトの進捗管理
- Next.js、Vercelを使用したSSRアプリケーション開発
- Agora、SocketIOを使用した通話機能開発
- TailwindCSSの導入、技術指導
- Vercel、AWSからのGCP移行
- GCPでのアプリケーション導入、運用
- Framer Motionを使用したアニメーション実装

まったく何もない状態から新規事業開発として参加し社内社外を含めたエンジニアチームへのマネジメントやスケジュール管理、未経験のエンジニアへの技術指導をしながら開発を進めていきました。

盛り込みたい機能は数多く出てきましたが人的リソースや時間は限られているため、ユーザーが初期リリース時に必要な通話機能に重点を置いた要件定義や技術制定を行い開発実装し無事リリースすることができました。

また、APIドキュメントやER図を自動作成し機能仕様が常にしっかりと更新されることで今後人数が増えても属人化しにくいように工夫しました。

CI/CDの導入はもちろんのこと、OpenAPIから生成された型定義ライブラリでフロントエンド、バックエンドを連携しました。型の不整合防止やバックエンドの開発を待たずにフロントエンドが開発を進められる環境を整え開発スピードを向上することができました。

Reactで使われるCSSの開発スタイルはModuleCSS、Styled Component、emotion、TailwindCSSと主要な開発を経験した上で最も開発スピードが期待でき、コードの統一感が出るTailwindCSSを採用しました。当時はまだ使ってるエンジニアが少なかったため勉強会を開いてチーム内の知識が共有されるように心がけました。

## クラウドファンディングリターン管理アプリケーション開発（2021年 6月）

### 【担当業務】PWAでのWebアプリケーション制作

- Heroku、Postgresを使用したデータベース構築
- Next.jsを使用したSSRアプリケーションを開発、PWA化
- GSAPを使用したアニメーション実装

2週間というタイトなスケジュール中、企業様とコミュニケーションを取りながら要件定義、設計、コーディングをすべて1人で担当し納品しました。

業務アプリケーションで使用するユーザーも多くないため、運用費用がかからない技術選定を行いました。時間が短い中でもユーザビリティを考え、GSAPを使ったリッチなアニメーションを取り入れ、ユーザー体験の向上を計りました。

## スマートロック/不動産管理アプリケーション開発（2020年 6月 ～ 2021年 3月）

### 【担当業務】自立分散型システムを用いた新規機能開発/現行機能の保守運用

- Reactを使用したSPAアプリケーション開発
- React Nativeを使用したクロスプラットフォーム開発、保守運用
- Swiftを使用したiOS開発
- Expressを使用したRestAPI開発
- Firebaseを使用したアプリケーションの保守運用
- Cloud Functionsを使用したサーバレスアプリケーション実装
- GCPでのアプリケーション監視

- Sentryでのアプリケーション監視

業務未経験からReact Native、Swiftをキャッチアップしプロダクトの保守運用しました。

営業部門、ファームウェア部門、デザイナー部門、QA部門と連携をとりながらスマートロックの設置導線やスマートロックのユーザー権限の制御を実装を行い、2週間サイクルでのリリースをこなしました。

toC向けのスマートロックアプリ開発だけではなく、toBに向けた不動産管理アプリケーション開発、社内システムの開発をしました。社内システムの開発ではスキューモーフィズムを取り入れUI/UXをこだわりキャンペーン等の通知機能を広告部門だけで自走できるに工夫しました。

また、コンポーネントをパッケージ化し社内の複数プロジェクトにおいて再利用できるような仕組みを提案し実装しました。コンポーネントを可視化できるようにStorybookの導入することで全体でのパッケージの使用率を高め実装スピード向上に貢献しました。