

3	2024年8月 - 2025/2/28	<p>動画に関する制作・配信・運用システムの運用</p> <p>[概要] 動画の制作・配信・運用を管理するシステムの開発に携わりました。 本プロジェクトでは、安定した動画配信とスムーズな運用を実現するため、サーバーサイドの開発やデータベース設計、API設計を担当しました。</p> <p>[担当業務] ■ サーバーサイド開発 ・ 動画の管理や配信に関わるバックエンド機能の実装 ・ 高トラフィックを考慮したパフォーマンス最適化 ・ ユーザーのアクセス管理や認証機能の実装</p> <p>■ データベース設計 ・ 動画メタデータ、配信ステータス、ユーザー情報などを管理するDBスキーマの設計 ・ クエリの最適化によるパフォーマンス向上 ・ 運用を考慮した拡張性の高い構造の設計</p> <p>■ API設計 ・ フロントエンドや外部サービスと連携するためのREST APIの設計・実装 ・ APIのレスポンス最適化や認証・認可の設計</p> <p>■ インフラ構築・運用 (GCP) ・ GCP環境上でのアプリケーション実行基盤の構築・設定 (Compute Engine / GKE / Cloud Load Balancing) ・ Dockerを用いたコンテナ化およびGKEへのデプロイ ・ Terraformによるインフラ構成管理 (VPC / サブネット / IAM / GKEクラスタ設定など) ・ Cloud Logging / Cloud Monitoring を用いたログ分析およびアラート対応 ・ Cloud Storage を利用した動画データの保存および配信連携</p> <p>・ Cronによるバッチ処理の定期実行・運用 ・ Pub/Sub を利用した非同期処理・メッセージング基盤の運用補助</p> <p>・ コードレビュー ・ チームの開発品質を向上させるためのレビュー対応 ・ コーディング規約の遵守や、保守性・拡張性を意識した指摘・改善</p> <p>[成果] ・ スムーズな動画配信を実現し、システムの安定性を向上 ・ API設計の改善により、フロントエンドとの連携がスムーズに ・ データベースの最適化により、検索・配信処理のパフォーマンスを向上</p>	フルスタック エンジニア チーム人数：8名 PM：1名 サーバサイド：5名 Webフロント：2名	Python Go TypeScript	MySQL	ブラウザ Ubuntu20.04	<p>・ インフラ / クラウド GCP (Compute Engine / GKE / Cloud Storage / Cloud Logging / Cloud Monitoring) Terraform (IaCによるインフラ構成管理) Kubernetes (GKE上でのコンテナ運用 / スケーリング対応) Docker (アプリケーションのコンテナ化およびデプロイ)</p> <p>・ フロントエンド React / Next.js (管理画面の実装、APIとの連携)</p> <p>・ 開発ツール GitHub (Pull Requestベースでのコードレビュー運用) Backlog (課題・進捗管理) Slack (非同期コミュニケーション・障害対応連携)</p>	● ● ● ● ●
(6ヶ月)								
4	2023年10月 - 2024年7月	<p>toC向け口コミサイトのリニューアル</p> <p>[概要] toC向け口コミサイトのリニューアルプロジェクトにおいて、システム全体の設計・開発に携わりました。 ユーザー体験の向上とパフォーマンス最適化を目的に、サーバーサイドおよびフロントエンドの開発、データベース設計、API設計を担当しました。 また、AWS環境上でのインフラ構築・運用にも関わり、スケラブルな構成の実現に貢献しました。</p> <p>[担当業務] ■ サーバーサイド開発 ・ 口コミ投稿・編集・削除機能の実装 ・ ユーザー認証・認可の仕組みを構築 ・ 高トラフィックを考慮したキャッシュ戦略の導入 ・ パフォーマンス改善 (クエリ最適化、負荷分散対応 など) ・ フロントエンド開発 ・ UI/UXを考慮したデザインのリニューアル ・ ユーザーが直感的に口コミを投稿・検索できる機能の実装 ・ レスポンシブ対応によるスマホ・PCでの最適化</p> <p>・ データベース設計 ・ 口コミ情報、ユーザー情報、評価データを管理するDBスキーマの設計 ・ データの正規化とパフォーマンス向上を意識したテーブル設計</p> <p>・ API設計・実装 ・ フロントエンドと連携するREST APIの設計・開発 ・ 認証・認可機能の実装 (JWTトークン管理など) ・ 外部API連携の設計 (地図情報、画像処理 など)</p> <p>・ コードレビュー ・ コーディング規約の統一と品質向上のためのフィードバック ・ 可読性・保守性を考慮したリファクタリングの提案 ・ パフォーマンス最適化やセキュリティ対策の指摘</p> <p>■ フロントエンド開発 ・ UI/UXを考慮したデザインのリニューアル ・ ユーザーが直感的に口コミを投稿・検索できる機能の実装 ・ レスポンシブ対応によるスマホ・PCでの最適化</p> <p>■ インフラ構築・運用 (AWS) ・ AWS上でのアプリケーション実行基盤の構築・運用 (EC2 / ECS / ALB など) ・ Dockerを用いたコンテナ化およびECSへのデプロイ ・ Terraformによるインフラ構成管理 (VPC / IAM / ECS / EC2など) ・ SQSを用いた非同期処理基盤の構築および運用 ・ SESを利用したメール送信機能の実装 (通知・認証メールなど) ・ Cognitoを用いたユーザー認証・認可基盤の構築 ・ CloudWatchを用いたログ監視およびアラート対応 ・ 承認フロー (メール認証・アカウント有効化など)の設計・実装</p> <p>[成果] ・ サイト全体のリニューアルにより、UXを大幅に改善 ・ DB設計とAPI最適化により、データ処理の速度を向上 ・ コードレビューを通じて、チーム全体の開発品質を向上</p>	フルスタック エンジニア チーム人数：12名 PM：2名 サーバサイド：6名 Webフロント：4名	TypeScript Next.js	MySQL	ブラウザ Ubuntu20.04	<p>Python 3.11 / FastAPI (API・非同期処理) / Next.js (フロントエンド) AWS (EC2 / ECS / SQS / SES / Cognito / CloudWatch) Docker Terraform GitHub StopLight Backlog Slack</p>	● ● ● ● ●
(10ヶ月)								

言語	自己評価	OS、DB、フレームワーク、 周辺技術	自己評価	業務範囲	自己評価	ヒューマンスキル、経験等	自己評価
Python	AA	Django	AA	要件定義	AA	人材育成	A
Rust	A	FastAPI	A	基本設計	AA	チームのモチベーションマネジメント	B
PHP	AA	Actix-Web	A	実装	AA	折衝	B
JavaScript(TypeScript)	A	Laravel	A	事前調査	A	プロジェクトの炎上消火	AA
Go	A	MySQL	AA	提案、企画	A	ドキュメント作成	A
Java	A	PostgreSQL	AA	テスト工程	A	メンタルケア	A
Perl	B	Redis	A				
C#	B	Vue.js, Nuxt.js	A				
		React, Next.js	A				

※評価基準 AA:開発ポリシーを有し、チームをリードする高い生産性。 A:周辺技術をおさえ、高い生産性を発揮。 B:基礎技術をおさえ、通常の生産性を発揮する。
C:一人で調べながらの作業が可能。