職務経歴書(2024/08/24)

        - 使用技術
        - 課題、対処方法、振り返りなど
    - 汎用ECシステム開発
        - 概要
        - 期間
        - メンバー
        - 担当業務
        - 使用技術
        - 課題、対処方法、振り返りなど
    - B2B向け新規自社グループECシステム構築
        - 概要
        - 期間
        - メンバー
        - 担当業務
        - 使用技術
        - 課題、対処方法、振り返りなど
    - B2B向け新規自社ECサイト構築
        - 概要
        - 期間
        - メンバー
        - 担当業務
        - 使用技術
        - 課題、対処方法、振り返りなど
    - 既存自社ECサイトの機能追加及び改善
        - 概要
        - 期間
        - メンバー
        - 担当業務
        - 使用技術
        - 課題、対処方法、振り返りなど
    - B2B向け新規自社ECサイト構築
        - 概要

基本情報

- 氏名: 佐佐木基顕(Sasaki Motoaki)
- 誕生日: 1992年11月8日
- 国籍: 日本

概要

抽象的な強み

新しいことへのキャッチアップ能力

旺盛な好奇心と、飽くなき上昇志向、情報収集能力、収集した断片的な情報をもとに再構築するのに十分な抽象化・具体化能力を持っており、それをもとにしたキャッチアップ能力の高さが売りである。これは、本来の才能に加え、これまでに様々な分野に挑戦したり、変化の多い環境で活動する中で地

道に身につけたものである。そのため、技術的なことだけでなく、要件の理解能力などにも活かされている。

難易度の高いタスクに対する対応力

前述のキャッチアップ能力の高さから、自分の知らない分野であっても、即座に学習することで、平均以上の能力を発揮し、対処することができる。さらに、大学院時代に培った「じっくり考える力」を掛け合わせることで、難易度の高いタスクにも対応することができる。

具体的な強み

プロジェクト

- 不確定な要素が多かったり流動的な状態のプロジェクト
- 既存のスキルを活かしつつ、他の技術にも挑戦することが奨励されるプロジェクト
- 要件を本質的に理解した上で実装することが好まれるプロジェクト
- 開発者から要件レベルのフィードバックをすることが好まれるプロジェクト
- 要件や技術について確認することが好まれるプロジェクト
- コード品質をある程度重視しているプロジェクト

タスク

- 調査系タスク
  - 技術選定
  - 実現可能性の調査
- 設計系タスク
  - 拡張性を意識したDB設計
  - インフラ設計
  - バックエンド設計
- 実装系タスク
- 領域横断的なタスク(バックエンド、フロントエンド、インフラなど)
- 要件、保守性や可読性など、様々な観点でのコードレビュー
- アプリケーションのコンテナ(Dockerfile)化

- CI/CDパイプラインの構築
- 静的解析ツールの導入
- ユニットテストの導入
- フォーマットツールの導入
- バグの発見、原因の特定、改修案の提案、および改修

基本的な仕事の進め方

- どんなタスクであれ、相手が求めていることと自分がやろうとしていることの溝を可能な限り無くしてから作業に移るようにしている。また、プロジェクト遂行上の想定外のリスクや懸念が見つかれば、できるだけ早く共有する。
- 実装系のタスクでは、作業に入る前に、可能な限り要件を理解し、不明点や違和感があればフィードバックを行う。お互いが完成物に対して納得できた状態で開発を行うよう心がけている。
- コードを書く際は、保守性を重視し、特に、依存の方向、関心の分離を意識している。テストについては、最低限パブリックな関数の単体テストを、正常系、異常系に対して記述するようにしている。
- コードレビューでは、まず、変更を行うに至った背景を理解する。コードの変更は手段なので、その目的を知らなければ、正確なレビューはできないためである。コードを見る際は、要件を満たしているか、コードの保守性、バグの有無、セキュリティ的な懸念はないか、などを中心に行う。なお、プロジェクトの状況により、各レビュー観点をどの程度重要視するべきかは変わるため、柔軟に対応する。
- インフラ関係のタスクでは、基本的なベストプラクティスに沿った設計や、最小権限の原則に基づいた権限の調整を心がけている。時には、利便性とセキュリティのバランスを考える必要があるが、その際もセキュリティ的に問題がないか十分に考慮する。

職務経歴概要

最新のものから記載

株式会社VirtualWall

- 設立: 2021年06月

- 資本金: 1億5000万円（資本準備金含む）
- 社員数: 13人(2024/8時点)
- 在職期間:
  - 業務委託: 2022/12~2023/2
  - 正社員: 2023/3〜現在
- 職種: テックリード

自社プロダクトである「極楽譲渡」の開発に、プロジェクト初期から従事。バックエンドの開発を中心に、フロントエンド、インフラ、ブロックチェーン、CI/CD、エンジニア採用、教育など、幅広い業務を担当。キャッチアップ能力の高さを生かし、領域横断的な業務を多く担当している。また、実現可能性がわからないような難易度の高いタスクも多く担当した。

株式会社レスタス

- 設立: 2011年6月
- 資本金: 10億1,183万円（資本準備金を含む）
- 社員数: 43人(2023/2時点)
- 在職期間: 2020/4〜2023/2
- 職種: バックエンドエンジニア

自社ECシステムの開発に従事。最初 1 年ほどは、新規開発と保守開発の両方を担当し、それ以降は基本的に新規開発を担当。入社して以降、主にバックエンドの開発を担当しているが、フロントエンド、インフラの業務にも積極的に挑戦し、現在は、バックエンド、インフラをメインで担当している。

株式会社NSC

- 設立: 1971年3月11日
- 資本金: 4,550万円
- 社員数: 523人（2021年2月現在）
- 在職期間: 2018/4〜2020/1
- 職種: 製造職

本企業での経験の詳細については、業界業種ともに、異なるものであるため、説明を省略しています。

職務経歴詳細

最新のものから記載

株式会社VirtualWall

オンライン完結型の二次取引を前提とした、SaaS型投資系クラウドファンディングシステムの開発

概要

現状、ファンド商品の二次取引は紙ベースで行われているため、煩雑な作業が存在する上に、手続き完了にも時間がかかる。そのため、現状のファンド商品の多くは、二次取引を前提としたものではない。本プロジェクトでは、ブロックチェーン技術を活用し、二次取引をペーパーレス化することにより、ファンド商品を二次流通しやすくさせ、流動性を向上させる。

期間

2023/2~現在

メンバー

記載時点(2024/8)の構成を記載。

プロジェクト全体:21人

役割	人数
プロダクトマネージャー	2(正社員1、業務委託1)
プロジェクトマネージャー	1(業務委託1)
テックリード	1(自分)
バックエンドエンジニア	7(正社員2、業務委託5)
フロントエンドエンジニア	2(業務委託2)
インフラエンジニア	3(業務委託3)
デザイナー	1(業務委託1)

営業など	4

担当業務一覧

- フロントエンド
  - 開発環境の整備
    - 静的解析ツールの導入
      - eslint
      - stylelint
      - erblint
    - ユニットテストに必要なツールの選定および導入
      - vitest
      - happy-dom
    - フォーマットツールの導入
      - prettier
- バックエンド
  - 入出金関係の全体的なテーブル設計(合計20テーブルほど)
  - 入出金関係の機能の開発
  - 独自ドメインを利用可能にするための機能の実現可能性調査、業務フローの叩き台作成およびサーバーサイドの開発(ドメインレジストラ、AWSマネージドサービス、サーバーサイドの実装)
  - 排他制御用のレコードを用いた排他制御機構の実装
  - その他各種機能の開発
  - erb_lintのカスタムリンターの作成を通した、ガードレールの作成
- ブロックチェーン
  - スマートコントラクトの開発
  - 静的解析ツールの導入
    - Slither
    - Mythril
  - GitHub Actionsを用いたCIの整備
    - Slither
    - Mythril
    - forge fmt
    - forge test

- インフラ
  - 本番環境向けのDockerfileの作成
  - インフラ設計およびインフラ構成図の作成
    - ネットワーク構成図
    - システム構成図
  - 設定ファイルを元に、各ネットワークに割り当てるIPアドレスを計算し、出力するプログラムの作成
  - 構築されたインフラ上で、アプリケーションを動作させるようにする作業(アプリケーションとインフラの両方の知識、問題の切り分け能力が必要)
  - terraformコードの軽微な修正
- CI/CD
  - キャッシュを用いたGitHub Actionsの実行時間削減(ビルド時間3分->30秒)
  - CDパイプラインの作成
    - GitHub Actions
    - AWS Identity and Access Management(OIDC IDプロバイダーを利用)
    - Amazon Elastic Container Registry
    - Amazon Elastic Container Service
- その他
  - コードレビューの仕組みの改善
    - GitHubの機能を用いた、レビュアー自動割り当ての仕組みの導入
  - RDBのスキーマ情報から、ER図を自動生成するプログラムの作成
  - devcontainerを用いた開発環境の整備
  - エンジニア採用
    - 書類選考
    - 面接
      - 質疑応答
      - ライブコーディングテスト
  - コードレビュー
  - 開発メンバーからの技術的な質問・相談などへの回答
  - 教育

    - 業務未経験エンジニアのメンター(1ヶ月)

使用技術

- Ruby
- Solidity
- TypeScript
- JavaScript
- Shell Script
- Ruby on Rails
- Docker
- Docker Compose
- AWS Identity and Access Management
- Amazon Elastic Container Service
- Amazon Elastic Container Registry
- AWS Certificate Manager
- AWS Secrets Manager
- Amazon Simple Storage Service
- Amazon CloudFront
- Elastic Load Balancing
- Amazon RDS
- Terraform
- kaleido
- GitHub
- GitHub Actions
- reviewdog
- Foundry
- Slither
- Mythril

課題、対処方法、振り返りなど

インフラ設計についての振り返り

プロジェクトの中で、インフラチームのメンバーにRailsアプリケーションのデプロイに関する知識がないことから、誰がインフラの設計を行うかどうか、話し合うことがあった。その話し合いの中で、少なくともどちらかに詳しい、インフラあるいは、サーバーサイドエンジニアの誰かが行うのが良いのではないかという話になった。そこで、私は手をあげ、AWSの全体的な理解、各リソースに関する理解、AWSを用いてデプロイする際のベストプラクティスの調査などを行ったのち、インフラチームのメンバーにレビューしてもらいながら各構成図を作成した。これらの作業を通して構築されたインフラは、現在も問題なく稼働している。

独自ドメイン機能の開発についての振り返り

各テナントが、独自ドメインを用いてサービスを展開できるようにする機能を開発する必要があったが、開発メンバーの誰もその実現方法を知らなかった。そこで、他社サービスのドキュメントの記述内容からの仕組みの分析、実現手法の比較および提案、実際のAWSリソースを用いた実現可能性の確認、簡易的な業務フローの作成、実装範囲の相談、サーバーサイドの実装など、幅広く作業を行った。こういった領域横断的なタスクは得意分野である。

株式会社レスタス

既存APIのリプレイス

概要

既存のRubyで書かれたAPIをGoでリプレイスするプロジェクト。

期間

2022/9~2022/12(3ヶ月)

メンバー

プロジェクト全体:3人

役割	人数
バックエンドエンジニア兼インフラエンジニア	1(自分)

バックエンドエンジニア	2

担当業務一覧

- 技術選定
- GitHub Actionsを用いたCIの設定
- 開発環境の整備
- 既存のORMのモデルからGoのORM(gorm)のモデルを生成するプログラムの作成
- 既存のOpenAPIドキュメントからGoのコードを生成するプログラムの作成
- 既存APIのGoへの書き換え
- Kubernetes(GKE Autopilot)を用いた、staging環境、production環境の構築
- コードレビュー
- reviewdogを用いた、レビュー省力化の仕組み導入

使用技術

- Ruby
- Go
- Shell Script
- Gin
- Ruby on Rails
- gorm
- gin-swagger
- zap
- air
- reviewdog
- staticcheck
- golangci-lint
- Docker
- Docker Compose
- Kubernetes

- Cloud Build
- Google Container Registry
- Google Kubernetes Engine(Autopilot)
- GitHub
- GitHub Actions

課題、対処方法、振り返りなど

リードエンジニアとしての参画。 他のエンジニアがGoの実務未経験であったため、コードの品質を担保するために、開発環境のコード化(Dockerコンテナ化, VSCode設定の共有)や、CIの充実、実装の手本となるようAPIのいくつかを事前に移植、モデル変換プログラムの作成、などを行っており、現状プロジェクト進行上問題となるようなことは起こっていない。 また、Goの勉強会を不定期で開催し、メンバーの能力向上にも努めた。 技術選定では、可能な限りデファクトスタンダードなものや、Ruby on Rails経験者である他の開発者にとって学習コストの低いものを選んでいる。

汎用ECシステム開発

概要

汎用的なECシステムの構築プロジェクト。

期間

2022/2~~2022/8(新規開発: 7ヶ月) 2022/9~~2022/12(改善・保守: 4ヶ月)

メンバー

プロジェクト全体:5人

役割	人数
プロジェクトマネージャー兼バックエンドエンジニア兼インフラエンジニア	1
バックエンドエンジニア兼インフラエンジニア	1(自分)
フロントエンドエンジニア	1

デザイナー兼フロントエンドエンジニア	1
デザイナー	1

担当業務

- 担当機能における、要件定義、テーブル設計、実装、テスト(単体、API)作成
- OpenAPIドキュメント生成機能付きFWの導入による、OpenAPIドキュメント自動生成の仕組みの構築
- APIの実装
- GitHub Actionsを用いたCIの設定
- Kubernetes(GKE Autopilot)を用いた、staging環境、production環境の構築
- Kubernetes(GKE Autopilot)を用いた、検証環境自動構築の仕組みの構築
- デプロイ属人化を防ぐslackアプリ(TypeScript)の作成
- 画像を透過処理して保存するAPIサーバー(Go)の作成
- APMツールの導入(Cloud Trace)

使用技術

- Ruby
- Go
- Solidity
- Shell Script
- JavaScript
- TypeScript
- Ruby on Rails
- grape
- goa
- rubocop
- eslint
- prettier
- jest
- zap

- make
- esbuild
- staticcheck
- Docker
- Docker Compose
- Kubernetes
- Cloud Build
- Google Container Registry
- Google Cloud Storage
- Google Kubernetes Engine(Autopilot)
- GitHub
- GitHub Actions
- GitHub Copilot
- Foundry
- Visual Studio Code
- Cursor

課題、対処方法、振り返りなど

バックエンドエンジニアとして参画。のちにインフラも兼任。 OpenAPIドキュメント自動生成の仕組みを構築することで、APIドキュメントの管理コストの削減、APIドキュメントが常に最新のコードと同じ状態が担保される、フロントエンドからのAPIの明確化、などに加え、生成されたOpenAPIドキュメントをもとに、プロキシサーバーの設定ファイルを自動生成する仕組みへの応用、などの間接的なメリットも感じることができ、成功だったと感じている。

B2B向け新規自社グループECシステム構築

概要

グループ会社の、B2B向け新規ECシステムの構築プロジェクト。

期間

2021/7~2022/1(7ヶ月)

メンバー

プロジェクト全体:10人

役割	人数
プロジェクトマネージャー	1
バックエンドエンジニア	2(自分含む)
フロントエンドエンジニア	1
デザイナー兼フロントエンドエンジニア	1
デザイナー	2

担当業務

- 企画サイドと、エンジニア間での各種調整業務
- APIの実装
- 担当機能における、要件定義、テーブル設計、実装、単体テスト
- GitHub Actionsを用いたCIの設定

使用技術

- Ruby
- JavaScript
- Node.js
- Ruby on Rails
- rubocop
- Docker
- Docker Compose
- Google Cloud Functions
- GitHub
- GitHub Actions

課題、対処方法、振り返りなど

バックエンドエンジニアとして参画。のちに、開発をまとめていたバックエンドエンジニアの退職に伴い、調整業務も兼任。 引き続き、Ruby on Railsを用いた開発をメインで担当。 今までに扱ったことのない技術(ActiveJob, Github Actionsなど)に積極的に挑戦したり、以前よりさらに保守性の高いコードを書くことを意識して開発をおこなった。

B2B向け新規自社ECサイト構築

概要

リッチなフロントエンドアプリをベースとした、ECシステムの構築プロジェクト。

期間

2020/12~2021/6(7ヶ月)

メンバー

プロジェクト全体:7人

役割	人数
プロジェクトマネージャー	1
バックエンドエンジニア	1
フロントエンドエンジニア兼バックエンドエンジニア	1(自分)
フロントエンドエンジニア	2
デザイナー兼フロントエンドエンジニア	1
インフラエンジニア	1

担当業務

- PMと各エンジニアとの各種調整(プロジェクト中盤以降)
- フロントエンドでの画像アップロード用コンポーネント作成
- 画像アップロードAPIサーバー(Go)の作成
- フロントエンドアプリのDocker化

使用技術

- Go
- JavaScript
- Gin
- Vue.js
- Nuxt.js
- Docker
- Docker Compose
- Cloud Run
- GitHub
- GitHub Actions

課題、対処方法、振り返りなど

フロントエンドエンジニア兼バックエンドエンジニアとして参画。のちに、PM~エンジニア間の調整業務を兼任。 Nuxt.js, Goを用いた開発は初めての挑戦だったが、事前に学習を行なっていたので、問題なくキャッチアップすることができた。 また、フロントエンド、バックエンドに横断するような機能の実装を初めて担当した。

既存自社ECサイトの機能追加及び改善

概要

既存自社ECサイトに対する、機能追加及び改善のプロジェクト

期間

2020/12~2021/2(3ヶ月)

メンバー

プロジェクト全体:4人

役割	人数
プロジェクトマネージャー	1

バックエンドエンジニア(実装)	1
バックエンドエンジニア(要件定義~設計)	1(自分)
インフラエンジニア	1

担当業務

- 要件定義、テーブル設計、実装、テスト
- APIの実装

使用技術

- Ruby
- JavaScript
- TypeScript
- Ruby on Rails
- Vue.js
- Docker
- Docker Compose
- GitHub

課題、対処方法、振り返りなど

実装担当のバックエンドエンジニアのサポート(主に設計面)として参画。のちに、実装担当のバックエンドエンジニア退職に伴い、実装も担当。 改修対象のアプリで、TypeScript、VueCompositionAPIなど、未経験の技術が使われていたが、難なくキャッチアップし、改修することができた。 また、エンジニアになって初めてマネージメントの立場を経験することができた。

B2B向け新規自社ECサイト構築

概要

商材に依存しないECシステムの構築プロジェクト

期間

2020/8~2020/11(4ヶ月)

メンバー

プロジェクト全体:4人

役割	人数
プロジェクトマネージャー	1
バックエンドエンジニア	1(自分)
デザイナー兼フロントエンドエンジニア	1
インフラエンジニア	1

担当業務

- 要件定義、テーブル設計、実装、テスト
- 外部APIと連携した定期実行処理の設計・実装
- インフラエンジニアとの調整

使用技術

- Ruby
- JavaScript
- Ruby on Rails
- Docker
- Docker Compose
- GitHub

課題、対処方法、振り返りなど

バックエンドエンジニアとして参画。 要件定義~設計や、テーブル設計などを本格的に行なったのは初めてであった。 Kubernetesの機能を用いた定期実行処理や、外部APIと連携した処理を初めて実装するなど、技術的にも挑戦することができた。

B2B向け自社ECサイト構築

概要

B2B向け自社ECシステム構築

期間

2020/4~2020/7(4ヶ月)

メンバー

プロジェクト全体:6人

役割	人数
プロジェクトマネージャー	1
バックエンドエンジニア	2(自分含む)
フロントエンドエンジニア	1
デザイナー兼フロントエンドエンジニア	1
インフラエンジニア	1
※プロジェクトの途中からバックエンドエンジニアとして参画。	

担当業務

- バックエンド機能の開発・テストを担当。
- 一部機能は、仕様書を元に、自ら設計〜開発まで担当。

使用技術

- Ruby
- JavaScript
- Ruby on Rails
- Vue.js
- Docker
- Docker Compose
- GitHub

課題、対処方法、振り返りなど

バックエンドエンジニアとしてプロジェクトの途中から参画。 初めての実務レベルの開発を通じて、Ruby on Railsを用いた基本的な開発力を培うことができた。 Git, GitHubを用いたプルリクエストベースのチーム開発や、Dockerを用いた開発に慣れることができた。

アカウント

サービス	アカウント
GitHub	https://github.com/msasaki666
Qiita	https://qiita.com/motoakii
Zenn	https://zenn.dev/motoakii

その他

保有資格

- 日商簿記検定3級(2023/2/26受験)
- ITパスポート試験(2020/11/1受験)

OSS活動

業務上必要で対応したものを還元したり、業務の中で気づいたバグの修正を行ったことがある

- activerecord-multi-tenant gem...バグを修正した
- parallel gem...テストを実行対象のrubyバージョンを追加した

趣味

１つのことを長くやるより、いろんなことに挑戦するのが好きなタイプ。 食べ歩きや、ボードゲームは、比較的長く続いている趣味である。