

職務経歴書

2024/08/03

佐佐木基顕

目次

1. 基本情報
2. 職務経歴概要
 - 2.1. 株式会社VirtualWall
 - 2.2. 株式会社レスタス
 - 2.3. 株式会社NSC
3. 職務経歴詳細
 - 3.1. 既存APIのリプレイス
 - 3.2. 汎用ECシステム開発
 - 3.3. B2B向け新規自社グループECシステム構築
 - 3.4. B2B向け新規自社ECサイト構築
 - 3.5. 既存自社ECサイトの機能追加及び改善
 - 3.6. B2B向け新規自社ECサイト構築
 - 3.7. B2B向け新規商材向け自社ECサイト構築
4. スキルシート
 - 4.1. ハードスキル
 - 4.2. ソフトスキル
5. アカウント

1. 基本情報

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

2. 職務経歴概要

最新のものから記載

2.1. 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。新しいことへの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状態のタスクも多く担当した。

2.2. 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。

2.3. 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、以下の職務経歴詳細には記載していない。

3. 職務経歴詳細

3.1. 株式会社VirtualWall

3.1.1. 極楽譲渡

3.1.1.1. 概要

ファンド商品のプライマリ・セカンダリ取引を扱うSaaSシステムの新規開発。ブロックチェーン技術を活用し、これまでにないセカンダリ取引のペーパーレス化により、効率的な取引フローを実現する。

3.1.1.2. 期間

19ヶ月(2024/8時点)

3.1.1.3. 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

3.1.1.4. 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)
 - フォーマットツールの導入(prettier)
 - テストングツールの調査および選定
- 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、開発、及び各所との連携

- 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
- ブロックチェーン
 - スマートコントラクトの開発
- インフラ
 - 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
 - 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
 - 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)
 - GitHub Actionsを用いた、CDパイプラインの作成
- 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
 - 教育
 - 業務未経験エンジニアのメンター(1ヶ月)

3.1.1.5. 使用技術

- Ruby
- Solidity
- TypeScript
- JavaScript
- Shell Script
- Ruby on Rails
- Docker
- Docker Compose
- 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

3.1.1.6. 課題、対処方法、振り返りなど

未経験の技術、実現可能性のわからないタスク、領域横断的に考える必要のあるタスクに対して、積極的に取り組むことができた。

独自ドメイン

排他制御

AWS

CDパイプライン

インフラ構成図

スマコン開発

採用

3.2. 株式会社レスタス

3.2.1. 既存APIのリプレイス

3.2.1.1. 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。

3.2.1.2. 期間

3ヶ月

3.2.1.3. 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

3.2.1.4. 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

3.2.1.5. 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

3.2.1.6. 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。

3.3. 汎用ECシステム開発

3.3.0.7. 概要

汎用的なECシステムの構築プロジェクト。

3.3.0.8. 期間

参画~リリースまで: 7ヶ月

以後、改善・保守などを現在まで継続

3.3.0.9. 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

3.3.0.10. 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)

3.3.0.11. 使用技術

- Ruby
- Go
- 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

3.3.0.12. 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。

3.4. B2B向け新規自社グループECシステム構築

3.4.0.13. 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。

3.4.0.14. 期間

7ヶ月

3.4.0.15. 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

3.4.0.16. 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

3.4.0.17. 使用技術

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

3.4.0.18. 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。

3.5. B2B向け新規自社ECサイト構築

3.5.0.19. 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。

3.5.0.20. 期間

7ヶ月

3.5.0.21. 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

役割	人数
インフラエンジニア	1

3.5.0.22. 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

3.5.0.23. 使用技術

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

3.5.0.24. 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。

3.6. 既存自社ECサイトの機能追加及び改善

3.6.0.25. 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

3.6.0.26. 期間

3ヶ月

3.6.0.27. 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

役割	人数
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

3.6.0.28. 担当業務

- 要件定義、テーブル設計、実装、テスト
- APIの実装

3.6.0.29. 使用技術

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

3.6.0.30. 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ジメントの立場を経験することができた。

3.7. B2B向け新規自社ECサイト構築

3.7.0.31. 概要

商材に依存しないECシステムの構築プロジェクト

3.7.0.32. 期間

3ヶ月

3.7.0.33. 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

役割	人数
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

3.7.0.34. 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

3.7.0.35. 使用技術

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

3.7.0.36. 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。

3.8. B2B向け自社ECサイト構築

3.8.0.37. 概要

B2B向け自社ECシステム構築

3.8.0.38. 期間

4ヶ月

3.8.0.39. 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

役割	人数
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。	

3.8.0.40. 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。

3.8.0.41. 使用技術

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

3.8.0.42. 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。

4. スキルシート

4.1. ハードスキル

カテゴリー	スキル	経験期間	備考
OS	macOS	2年7ヶ月	業務で使用。
OS	Linux	2年7ヶ月	Dockerを用いた開発や、アプリケーション実行環境として使用。
言語	Ruby	2年7ヶ月	Ruby on Rails, grapeを用いたWebアプリケーション開発で主に使用。
言語	Go	10ヶ月	Gin,goaを用いたWebアプリケーション開発で主に使用。
言語	Python	3ヶ月	FastAPIを用いた開発で使用。 基本的なプログラミングが可能。

カテゴリー	スキル	経験期間	備考
言語	JavaScript	1年6ヶ月	Nuxt.jsを用いたフロントエンドアプリケーション開発、Ruby on Railsを用いた開発で使用
言語	TypeScript	1ヶ月	slackのチャットボットアプリケーションの開発で使用
言語	Shell Script	1ヶ月	GitHubActionsやCloudBuildでのワークフローの記述や、簡単なスクリプトを作成する際などに使用
フレームワーク	Ruby on Rails	2年7ヶ月	Webアプリケーションの開発で複数回使用経験あり。
フレームワーク	grape	7ヶ月	RailsアプリケーションのAPI実装で使用経験あり。
フレームワーク	Gin	5ヶ月	Webアプリケーションの開発で2度使用経験あり。
フレームワーク	goa	2ヶ月	Webアプリケーションの開発で使用経験あり。
フレームワーク	FastAPI	3ヶ月	Webアプリケーションの開発で使用経験あり。 基本的なプログラミングが可能。
フレームワーク	Vue.js	1年3ヶ月	OptionsAPIを用いたプログラミングが難なく可能。SFC、VueRouter、Vuexなどを用いたSPAの作成で使用。 CompositionAPIは基本的なコードの読み書きが可能。
フレームワーク	Nuxt.js	5ヶ月	フロントエンドアプリケーションの開発で使用。
データベース	SQL(PostgreSQL)	2年7ヶ月	基本的には、ORM経由で使用。 ORMが出力したSQLを確認するため、読むことは可能。 基本的なCRUD操作以外は、調べながら書くことが可能。
インフラ	Docker	2年7ヶ月	開発時、ほぼ常に使用。
インフラ	Docker Compose	2年7ヶ月	開発時、ほぼ常に使用。
インフラ	Kubernetes	10ヶ月	GKEを用いたstaging環境、本番環境の構築経験あり。
DevOps	GitHubActions	1年	複数のプロジェクトでの使用経験あり。
クラウド	GCP	2年7ヶ月	CloudFunctions, CloudRun, CloudStorage, KubernetesEngineなどのマネージドサービスの使用経験、サービスアカウントの作成、WorkloadIdentityの、使用経験あり。

4.2. ソフトスキル

スキル	経験期間	備考
教育	3ヶ月	メンターとして、質問しやすい空気作りや、自発的な思考を促せるよう意識しながら、

スキル	経験期間	備考
		後輩エンジニア教育をした経験がある。
調整力		複数のプロジェクトで、企画サイド、PMなどとの調整を経験。
リーダーシップ	3ヶ月	リードエンジニアとして、計3人のチームを率いて開発を進めた経験あり。

5. 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