

# スキルシート 2025.05.26

---

## 基本情報

- 氏名: K・I
- 性別: 男性
- 年齢: 34

## 概要

- 5年間の法人営業経験後、Webエンジニアにキャリア転身。WordPress→Ruby on Rails→Flutter→Golang→AWS(Terraform)と経験を積み、直近約3年間はフリーランスエンジニアとして活躍。
- キャリアスタート時より0->1案件に参画することが多く、その中でDB設計や顧客折衝なども担当。強みとしては留学経験を活かした英語ドキュメントリーディングと新言語/技術のキャッチアップ能力に加えて、元々は営業職だった事もあり、関係者とのコミュニケーションスキルが高いと自負。
- バックエンド・ネイティブアプリ・インフラと幅広く経験してきた知見を活かし、特定の案件に縛られないバリューの提供が可能。

## スキル(実務で使用した技術のみ記載)

### 言語

- Ruby
- Dart
- Golang
- TypeScript
- JavaScript
- Swift
- Kotlin
- Python
- PHP

### フレームワーク・ライブラリ

- Ruby on Rails
- Revel
- Gin
- React
- Next
- Vue
- Flask
- WordPress

### クラウド

- AWS

- VPC
- S3
- Lambda
- EC2
- ECS
- Route53
- ACM
- IAM
- CloudWatch
  - Alarm
  - Dashboard
  - X-Ray
- EventBridge
- Step Functions
- AWS Batch
- SES
- SNS
- ALB
- SecretsManager
- GuardDuty
- CloudTrail
- Security Hub
- Config
- CloudFront
- Aurora
- Auto Scaling
- CodePipeline
- Cognito
- OpenSearch
- Pinpoint
- Athena
- QuickSight
- Redshift

## その他

- MySQL
- PostgreSQL
- SQLite
- Docker
- Terraform
- nginx
- VPS
- OpenTelemetry

## 資格

- 基本情報技術者(2021年5月)
- AWS Certified Solutions Architect – Associate(2021年11月)
- AWS Certified Solutions Architect – Professional(2024年4月)
- AWS Certified DevOps Engineer – Professional(2024年12月)

## 言語

- 日本語
  - ネイティブ
- 英語
  - TOEIC: 755

## 主な職歴 ※降順

### 東北6県に店舗展開するドラッグストアのモバイルオーダーアプリにおけるインフラ構築(2024年5月～現在)

#### プロジェクト概要:

- 自社開発モバイルオーダーシステムの構築

#### 使用技術:

- AWS
- Terraform

#### 担当業務:

- Terraformによる0→1のインフラ設計・構築
- SLO/SLI策定、およびその監視を行うためのメトリクスの設計・実装
- Pinpoint(Golang SDK)によるPush通知機能の実装

#### 発揮したバリュー:

システム要件や仕様が決まっていない中、変更や環境の追加に柔軟に対応できる実装を迅速に遂行。ディレクトリを複製し、2ファイルにまたがる設定値(ドメインやRDSのインスタンスクラスなど)を変更するのみで別の環境を構築できる設計にすることで、環境構築の手間を大幅に削減。約半日でstg環境を立てることに成功。またサービス監視のメトリクスの設計・実装を行い、SLO/SLIの策定に貢献。主にCloudWatch AlarmやDashboardを全てTerraform上で完結させ、リソースの追加が入った際も自動で監視対象に追加される運用を意識。CI/CDに関してはECRへのイメージpushをトリガーとしたCodePipelineを作成することで、アプリケーションチームのデプロイ負担を大幅に軽減。直近ではローンチ後のマーケティングや運用を見据えたBIやトレース、通知の仕組みづくりに着手中。

### 0->1案件特化受託企業でのインフラ/バックエンド開発 (2023年8月～2025年2月)

#### プロジェクト概要:

- Terraformで作成されたAWS構築テンプレートの機能追加・改修
- 新規開発Webシステム案件のバックエンド設計・開発・テスト

#### 使用技術:

- 言語: Golang, TypeScript

- フレームワーク: Revel, Next
- その他: AWS, Terraform, MySQL

#### 担当業務:

- TerraformによるFargate環境の構築
- GHAによるCI/CD
- X-Rayによるトレース
- インフラ費用削減のための構成変更
- Golangによるバックエンド開発
  - Revelと社内のコード自動生成サービスによる設計～開発～テスト。テーブル数にしておおよそ70以上存在するサービスを一人で開発。
  - 他社ベンダーが構築したTypeScript/NestJSで書かれたシステムをGolangで再構築。他社ベンダーからの情報提供が少なく、仕様も不明瞭な中コードを丁寧に読み込み、複数画像を組み合わせた画像生成機能の移植に成功。

#### 発揮したバリュー:

前任のインフラエンジニアが実装したTerraformのキャッチアップを迅速に行い、参画1ヶ月目からリソース作成PR→マージまで成功。複数案件に水平展開され、安定稼働される事を確認。主にCI/CD含む新規のWebホスティング環境構築を担当したが、顧客より別ベンダーが構築したAWSの費用削減依頼を受けた際もMySQL5.7から8系への移行や、それに伴うEC2→Fargateへの構成変更もTerraformにて納期通りに実装。バックエンドに関してはクライアントと付き合いがあった他社ベンダーが突然音信不通となり、Golangへの書き換え難易度が高いとの社内予測がある中、おおよそ2週間で画像生成とバッチ処理周りの実装に成功。同じく他社ベンダーが実装したフロントエンドの軽微な修正も担当し、本番環境では致命的なバグを発生させず、クライアントの希望納期前に納品完了。

### オフショア開発におけるマネジメント/インフラ構築 (2023年8月～2024年3月)

#### プロジェクト概要:

- 日本企業から受注したシステム開発案件における設計・マネジメント

#### 使用技術:

- 言語: AWS, Terraform, PHP, TypeScript
- フレームワーク: Laravel, Next
- その他: MySQL, Docker, nginx, VPS, Tailwind CSS

#### 担当業務:

- AWS/さくらVPSによる検証・本番環境構築
  - CloudFront/RDS/AutoScaling/EC2/ACM/Route53/ALB/S3による検証・本番環境環境構築を複数経験。パスルーティングにより同一ドメインでNextとWordPressを共存させるなど、クライアントの要望を満たす設計・実装に従事。また予算が限られている中、EC2のメモリスワップ機能を利用しメモリリソースを圧迫しないようにするなど工夫。本番環境に至ってはクライアント指定のさくらVPS上に構築する事となったが、SSL証明書の自動更新を含む環境構築を難なく実現。
- docker-composeによる開発環境構築
  - その企業ではこれまでDockerを利用してこなかったが、環境構築作業の効率化と環境の差異によるエラーを抑制する目的で導入。nginx/Laravel/Next/MySQL/LocalstackのDocker開発環境がコマンド一つで起動するよう、Makefileを活用。

- 新規開発案件のDB/アーキテクチャ設計
  - クライアントからヒアリングした内容を受け、DBの設計や技術スタックの選択を担当。Mermaid Chartを導入し、リアルタイムでDB設計の変更がエンジニアに伝わるように工夫。またFlutter/Nextのアーキテクチャ設計を行い、READMEに英語によるドキュメント化を実施。主に環境変数のmockフラグとRepositoryを活用したデータソースの切替や、各ディレクトリの債務をサンプルコードと共に提供。

#### 発揮したバリュー:

初めての英語環境、かつ技術スタックが過去に経験してきたものと大きく異なっていたが、今後の開発も見据えてフロントエンドとインフラの技術刷新を行い、現地エンジニアから高評価を獲得。新規スタックでの開発当初は開発速度が思うように伸びず経営側から不安視されていたが、2週目から速度が上がり、また別の案件でも同じ技術スタックを並行利用する事で社内技術の平準化に寄与。最終的には経営側より「今後も可能な限り同じ技術スタックを使いたい」という評価を獲得。

### CIツール開発・修正 (2022年4月～2023年3月)

#### プロジェクト概要:

- 某大手企業のシステム開発時に利用されている自社CIツールの機能追加、修正。

#### 使用技術:

- 言語: Golang
- その他: PostgreSQL, Bash, Kubernetes, Helm, GAE

#### 担当業務:

- Golangによるバックエンド開発
- GOPATHからMODULEモードへの移行対応
- Goのバージョンアップ(Ver1.12→1.19.5)

#### 発揮したバリュー:

ビルドに4~5分掛かる環境化での実装であったため、実装ミスを極力減らす目的でテスト駆動開発を習得。Golang未経験ながら2ヶ月目以降は1~2日に1PR出せるほどに成長。GOPATHモードとMODULEモードではGAEへのデプロイ方法が異なり、デプロイ時に「プライベートレポジトリのパッケージを特定のパスにダウンロード」「go.modファイルの書き換え」が必要があったが1コマンドで成功するShellスクリプトの実装に成功し、当時のGo最新バージョンへの移行に貢献。

### 契約書管理/社内資産管理アプリ開発(2021年6月～2021年11月) ※正社員

#### プロジェクト概要:

- WebViewを表示させているiOSアプリのフルリプレイス、およびAndroidアプリの新規開発

#### 使用技術:

- 言語: Dart, Swift, Kotlin
- フレームワーク: Flutter
- その他: SQLite

#### 担当業務:

- ネイティブコード(Swift/Kotlin)とOpenCVによるOS毎のカメラ機能開発
  - WeScan(Swift)ライブラリとSmartPaperScan(Kotlin)ライブラリを拡張し、カメラUIの調整、矩形の切取、回転、コントラスト調整、ソート機能などを実装。
- Flutterによるアプリ開発
  - 主にUIやバックエンドとの通信処理などをFlutterで実装。Flutterのコードを簡易化させるため、Swift,Kotlinそれぞれのライブラリで撮影・加工した画像はデータの型を統一した後で、MethodChannelでFlutter側に渡すことでアップロード処理は共通コードで実行することに成功。
- RiverpodとHooksによる状態管理
  - StateProvider,StateNotifierProvider,Hooksを使い分け、StatefulWidgetクラスを使用しないことで最低限のリビルドのみ行われる実装に成功。

#### 発揮したバリュー:

元々iOSアプリでは撮影した画像をメモリに保存する仕様になっていたため、Android版も同じ仕様で実装していたが、端末の一部でメモリのオーバーフローが発生する状態になり、SQLiteを活用する設計に変更。後の調査で元々のiOSアプリでもオーバーフローが発生していた事が分かったが、設計変更が功を奏しバグの解消に貢献。また画像が複数枚表示されるUIではリサイズ済みのサムネイル画像を表示させるなどの工夫も行い、動作のもたつき解消と安定化を実現。さらにOSの差異が発生しない機能は極力Flutterで処理させることでネイティブコードを極限まで減らし、コード量が少なくメンテナンス性の高いコードの記述に尽力。

### ビジネス用途通話アプリ開発(2020年4月～2021年5月) ※正社員

#### プロジェクト概要:

- WebRTCを利用した通話特化型のビジネス用途アプリ開発
- Appleストアへの申請・リリース対応

#### 使用技術:

- 言語: Dart, Swift
- フレームワーク: Flutter

#### 担当業務:

- プロジェクトリーダーとして社内上層部と開発チームのブリッジ役となり、スケジュール調整や仕様を固めるためのドキュメント作成などに尽力。
- Flutter,Swift共に未経験での挑戦となったが、公私共にキャッチアップを行いリリースまで一人で担当。開発だけに留まらず、テストアプリ配布などの付随業務も都度キャッチアップしながら対応

#### 発揮したバリュー:

社内にFlutterに精通したエンジニアが存在せず、かつ日本でのFlutter認知度が低い時期であり情報収集が困難な中でも、英語ドキュメントのリーディングと仮説・検証サイクルを高速に回すことで最終的にはストアへのリリースまで至った。また通話転送機能の実装を行い既存の通話アプリとの差別化を図ったことに加え、iOSの通話履歴との連携にも成功した。この頃から上司からの評価が著しく向上し、査定前に昇給・昇格の打診を受けるまでに至った。

### LINE風チャット&SNS統合サービスのバックエンド開発(2019年11月～2020年4月) ※正社員

#### プロジェクト概要:

- 大手運送業者のドライバー間や本社⇄ドライバー間のコミュニケーション量増加を目的としたアプリ開発
- テキストによるチャット機能に加え、掲示板やポイント交換によるインセンティブ機能を搭載

#### 使用技術:

- 言語: Ruby, JavaScript
- フレームワーク: Ruby on Rails, Vue
- その他: MySQL, Chart.js

#### 担当業務:

- RoRによる0->1のバックエンド設計・開発に従事し、単純なCRUDだけではなくAction Cableによるリアルタイムチャット機能などに挑戦。

#### 発揮したバリュー:

「DB/モデル設計→もう1名のバックエンドエンジニアとFB」を高速で回すことで手戻りが少なく済み、納期の2週間前倒しでの納品に成功。具体的には今までDB/モデル/コントローラーの実装まで行ってからPRを出していたサイクルを、DB/モデル設計時点でドラフトPRを出す方針に変え、レビュー通過後に実装に着手するサイクルに変更。またプロジェクトの途中でフロントエンドエンジニアの手が回らなくなり、2ヶ月間ほどVueとChart.jsによる管理画面作成の実装にも貢献し、プロジェクトを円滑に進めることに成功。APIの仕様はSwaggerで都度共有し、フロントエンドとアプリエンジニアの実装を止めない事を第一に意識。