

# 1 Resume

## 1.1 Table Of Contents

- Resume
  - Table Of Contents
  - 自己紹介
    - \* 興味関心
    - \* 性格
      - ・ 16personalities
  - 経歴一覧
    - \* 株式会社 Speee(2022 年 8 月~)
      - ・ 広告配信プラットフォームの開発
    - \* 株式会社良品計画 (2022 年 8 月~10 月)
      - ・ EC サイトのバックエンド・フロントエンド開発
    - \* 株式会社クラフトマンソフトウェア (2022 年 5 月~7 月)
      - ・ 大手メーカー様の内製 Android アプリのビルドパイプライン刷新
    - \* マキヤマブラザーズ株式会社 (2022 年 5 月~7 月)
      - ・ 自社 Web アプリケーションの開発
    - \* 株式会社ビットエー (2019 年 8 月~2022 年 5 月)
      - ・ 広告配信の最適化の基盤システムの開発・運用
      - ・ AI/VR を活用した不動産仲介プラットフォーム開発
    - \* 株式会社スタッフサービス (エンジニアリング事業本部) (2018 年 7 月~2019 年 7 月)
      - ・ 衛生画像の分析支援 Web アプリケーション開発
      - ・ 製薬開発支援 Windows アプリケーション開発
    - \* FIT 株式会社
      - ・ 太陽光発電パネルの統計情報管理 Web アプリケーション
  - できること
    - \* Backend
    - \* Infrastructure
    - \* Frontend
    - \* DevOop
    - \* Others
  - 今後の課題
  - 希望内容
    - \* 積極的にやりたいこと・できること
    - \* そこそこやりたいこと
    - \* 興味はある
    - \* できなくはないが積極的にはやりたくない
    - \* やりたくないこと
    - \* 待遇
      - ・ 最低希望年収
      - ・ その他 (MUST 条件)
      - ・ その他 (WANT 条件)

## 1.2 自己紹介

インターネットでは「かなで」という名前で活動しています。

2018 年 4 月より Web のエンジニアとしてキャリアを開始しています。

主たる領域はサーバサイド開発とクラウドを使ったインフラ構築、多少は Web フロントエンド開発の心得があります。

TypeScript や Golang と GCP、Terraform でのサーバサイド・インフラ開発が得意です。

twitter

GitHub

LAPRAS

Scrapbox

はてなブログ

### 1.2.1 興味関心

- Golang
- Rust
- GraphQL
- gRPC
- k8s
- 分散システム
- Observability
- Qwik
- 決済
- 広告
- 人事・組織
- セキュリティ

### 1.2.2 性格

#### 1.2.2.1 16personalities 管理者

## 1.3 経歴一覧

技術スタックは私が関わった範囲のみ記載しており、そのソフトウェアが内包している技術全てではない可能性があります。

### 1.3.1 株式会社 Speee(2022 年 8 月~)

#### 1.3.1.1 広告配信プラットフォームの開発

- 技術スタック
  - Backend
    - \* Go

- \* Python3.8
- \* unittest
- \* pandas
- \* MySQL
- Infrastructure
  - \* AWS
    - ・ Athena
    - ・ ECR
    - ・ S3
    - ・ Sagemaker
    - ・ ECS
- DevOps
  - \* Terraform
  - \* GitHub Actions
  - \* Airflow
  - \* Circle CI
- Others
  - \* Git

### 1.3.2 株式会社良品計画 (2022 年 8 月~10 月)

#### 1.3.2.1 EC サイトのバックエンド・フロントエンド開発

- 業務内容

2 つの業務内容があります。

1 つ目は Elasticsearch のパフォーマンス改善の調査とパフォーマンス改善をするための修正実装のプロトタイピングを行いました。

2 つ目は Web フロントエンドの TypeScript 化と Storybook の導入、カタログ作成を行いました。これにより

- 技術スタック
  - Backend
    - \* Java8
    - \* Spring Boot
    - \* PHP
    - \* Lumen
    - \* Go
    - \* MySQL
  - Frontend
    - \* JavaScript
    - \* TypeScript
    - \* React
    - \* Redux-Saga
    - \* Ant Design
    - \* Storybook
    - \* Jest

- Others
  - \* Git

### 1.3.3 株式会社クラフトマンソフトウェア（2022 年 5 月～7 月）

こちらは業務委託で参画させていただいております。

#### 1.3.3.1 大手メーカー様の内製 Android アプリのビルドパイプライン刷新

- 業務内容

ビルドパイプラインのインフラで使用している AWS の構築を担当しました。

- 技術スタック
- Infrastructure
  - AWS
    - \* ECR
    - \* S3
- DevOps
  - Docker
  - Docker Compose
  - Terraform

### 1.3.4 マキヤマブラザーズ株式会社（2022 年 5 月～7 月）

こちらは業務委託で参画させていただいております。

#### 1.3.4.1 自社 Web アプリケーションの開発

- 業務内容

マキヤマブラザーズ株式会社様が提供している複業サービス「DeLMO」の開発に携わりました。

- 技術スタック
- Backend
  - Go
  - MySQL
  - gqlgen
  - SQLBoiler
- Frontend
  - Next.js
  - Ant Design
  - Apollo Client
- DevOps
  - Docker

- Docker Compose
- Others
  - CodeCommit

### 1.3.5 株式会社ビットエー（2019 年 8 月～2022 年 5 月）

ビットエーではサーバサイドからインフラの設計・運用、フロントエンド開発とフルスタックに開発業務に携わりました。

SES であることから実際にお客様のエンジニアと会話したり非技術職の方ともコミュニケーションを取って業務を進めておりました。入社当時は課題感がありましたが、現在は指摘されることはないので非技術職とのコミュニケーションは問題ないかと思っております。

また開発業務以外では、主に知見共有と採用業務を行いました。

社内で技術知見を広めるために勉強会の主催や LT 会での多数登壇をしました。勉強会は普段業務では直接関わらないが現代の基礎となっている技術について取り上げています。現在はマスタリング TCP/IP を題材にネットワークについて勉強をしています。

採用業務では実際に社員・業務委託の面接に複数回出席して応募者様のスキルをヒアリングしたり会社とマッチングするかを会話しました。採用面接以外でも毎日良さそうなエンジニアの方を見つけては GitHub の実装を見て実際に声をかけたりプログラミングテストの採点を行なっておりました。

開発業務のみならず、会社が楽になるように日々動いています。

#### 1.3.5.1 広告配信の最適化の基盤システムの開発・運用

- 案件クライアント

美容系専門の広告代理店

- 期間

2021 年 1 月～2021 年 12 月

- 開発人数

2～5 人（途中で増加）

- 技術スタック
  - Backend
    - TypeScript
    - Node.js
    - Express
    - Jest
    - Puppeteer
  - Frontend
    - TypeScript
    - Next.js
    - React

- Jest
  - Ant Design
  - styled-components
  - Adobe XD
- Infrastructure
  - Google Cloud Platform
    - \* IAM
    - \* Cloud Run
    - \* App Engine
    - \* BigQuery
    - \* Cloud SQL
    - \* Cloud Pub/Sub
    - \* Cloud Tasks
    - \* Dataflow
    - \* DataStore
    - \* Cloud Storage
    - \* Container Registry
    - \* Cloud DNS
    - \* Cloud Build
  - MySQL
- DevOps
  - Docker
  - Terraform
  - GitHub Actions
- Others
  - Git
  - GitHub
  - Jira
  - Confluence
- 開発内容

開発全体としてはスクラム開発の手法を一部採用して Git-flow で開発をしました。1 週間を 1 スプリントとして、スプリントの最初に Sprint Planning をしてスプリント最後に Retrospective をして KPT について話し合いました。新たに発生したタスクに対しては story point を planning poker で見積もりをしました。

当初は自分と弊社の別メンバーのみで開発をしました。メンターをしていた常駐先クライアントの新卒 3 名を開発に加えて最終的には 5 名で開発をしました。

主に 2 つの開発をしました。

1 つは常駐先クライアント様の社内で広告運用に使用している社内 Web アプリケーションの機械学習を使った予算配分の最適化機能の開発です。

もう 1 つは自社顧客向けの効果測定と広告配信の最適化の基盤システム開発です。

社内 Web アプリケーションは自分がアサインされる以前に弊社の別メンバーが開発していたアプリケーションにエンハンスで新規のページから作成をしました。

構成は Backend が TypeScript+Express で、Frontend が TypeScript+Next.js で Cloud Run にデプロイしています。

初めて実務で TypeScript や Express、Next.js や GCP を使いましたが、アプリケーションの仕様を含めてキャッチアップを行い、スケジュール通りに新機能をリリースできました。

基盤システム開発は自社顧客向けに Facebook Conversion API を利用した広告配信の最適化と効果測定を目的としています。要件定義から開発運用を行いました。

立ち上げ時は自分を含め 2 名で開発をしました。プロパーではなかったですが、システム要件の策定からクラウドアーキテクチャの設計、開発と運用まで含めて任せていただきました。

現在ではリリース済みで、いくつかの企業様に導入していただいております。

この開発では、多くの GCP サービスを使い倒す機会を得ることができたこと、実務未経験でしたが IaC に Terraform を自分から導入し、インフラを安全に管理できるようにしました。

この 2 つの開発以外の業務では、広告運用チームとそのチームの担当顧客様との Facebook Conversion API の導入相談を受けたり、常駐先クライアント様の新卒のメンターを行いました。

開発面では IaC の旗振り役をやったりテストコードの推奨、アプリケーションや Terraform の CI/CD 自動化を導入しておりました。

#### 1.3.5.2 AI/VR を活用した不動産仲介プラットフォーム開発

- 案件クライアント

某外資系コンサルティングファーム

- 期間

2019 年 10 月～11 月 2020 年 4 月～12 月

- 技術スタック
- Backend
  - PHP7.4
  - Python3.6
  - Laravel
  - PHPUnit
  - SendGrid
  - Splunk
- Frontend
  - JavaScript
  - A-Frame
  - Vue2
  - Vuex
  - Jest
  - Sass

- Maps JavaScript API
  - Geocoding API
  - Places API for Web
  - Figma
- Infrastructure
  - MySQL
  - SQL Server
  - Azure
    - \* Virtual Machine
    - \* Azure Functions
    - \* Azure SQL Database
    - \* Azure Storage
    - \* Azure DevOps
    - \* Traffic Manager
    - \* Azure CDN
- DevOps
  - Docker
  - Docker Compose
  - Circle CI
- Others
  - Git
  - GitHub
  - AirTable
  - Trello
  - Kibela
- 開発内容

途中参画にはなりますが、入社して初めての案件でした。

Laravel を使ったバックエンド開発と Vue.js を使ったフロントエンド開発、詳細設計とテストを行いました。

Laravel と Azure は初めてだったためキャッチアップを行い、既存メンバーよりも詳しくなりました。

また以下のような実績があります。

- テストコードの導入
- Clean Architecture like な設計導入
- Bus Event を Vuex に置き換え
- Atomic Design の導入

まずバックエンドのフロントエンドともにテストコードがなかったため、テストコードの追加とテストコードを書く意識を作りました。テストコードを書くことで実装の理解が早まり、CircleCI や Azure Pipelines との相乗効果からデグレを未然に検知できるようになりました。結果として在籍している間は PoC もしましたが障害ゼロを達成できました。

そして設計指針を導入して実践しました。当初は Laravel の Model と Controller それぞれにビジネスロジックが混在しており、Fat Controller と Fat Model が入り混じった状態でした。ここに永続化層として



Repository 層、ビジネスロジックを記述する Service 層を追加、外部とのやり取りは Infrastructure 層に、Controller 層は HTTP リクエストを受け付けて Service 層に受け渡し、Service 層から受けた結果を HTTP レスポンスとして返すだけにしました。

設計指針を作ったことで全体の設計に一貫性が保たれ、自然とモジュールの責務について考えることができるようになりました。指針もお互いシステム設計に詳しいわけではないので、深入りしすぎることはせず、共通理解として作成して本来のプロダクト開発に専念しました。

フロントエンドでは 2 つあります。まず Bus Event を Vuex に置き換えました。

当時は Bus Event でコンポーネントの状態の受け渡しを行っていました。これを Vuex に置き換えたのですが、本来 Vuex で扱うべきではない状態も入り込んでしまい、一部は成功ですが一部は理想とは程遠い状態となってしまいました。あらかじめ現在の state を洗い出した上で、Vuex に持たせるべき state は何か、他はコンポーネントの依存関係を整理して解決できないかなどを考えるべきだったと思います。

またフロントエンドも設計指針がなかったので Atomic Design を導入しました。これも一部では設計指針として分かりやすかったのでよかったのですが、安易が導入は微妙だったという結論に至りました。Atomic Design は本来 UI デザイナー向けの設計手法であり、エンジニアが導入するなら当時導入していた Figma のコンポーネントから Atomic Design を踏まえて設計するべきでした。当時は Atomic Design に準拠したデザインになっておらず、その結果 atoms や molecules、organisms の粒度が曖昧になってしまい混乱が起きました。

現在もフロントエンドのコンポーネント設計は模索中ですが、安易に Atomic Design を選ぶべきではないという教訓を得ました。

### 1.3.6 株式会社スタッフサービス（エンジニアリング事業本部）（2018 年 7 月～2019 年 7 月）

#### 1.3.6.1 衛生画像の分析支援 Web アプリケーション開発

- 案件クライアント

大手日系メーカー子会社

- 期間

2018 年 10 月～2019 年 7 月

- 技術スタック
  - Backend
    - Java 8
    - Spring Boot
  - Frontend
    - JavaScript
    - Angular 1.x
  - Others
    - Git
    - BitBucket
  - 開発内容

Spring Boot でのバックエンド開発と Angular でのフロントエンド開発、開発後のテストで社内でのみの総合試験を行いました。

開発では元々画面の描画完了まで 10 秒かかっていた処理を 2 秒まで短縮したこと、総合試験で不要ログ削除ツールのバグを発見するなどをしました。

また総合テストではテスト項目以外でも自分で考えてテストを行い、非常にクリティカルなバグを発見し報告しています。

#### 1.3.6.2 製薬開発支援 Windows アプリケーション開発

- 案件クライアント

中小 SIer

- 期間

2018 年 7 月～2019 年 9 月

- 技術スタック
  - Application
    - C#
  - Others
    - SVN
    - Visual Studio
    - Windows
  - 開発内容

C#での Windows アプリケーションの開発をしました。

実装前にテストケースのマトリクスを提出する必要がある、ここでソフトウェアテストの観点やテストの意義について学びました。

#### 1.3.7 FIT 株式会社

##### 1.3.7.1 太陽光発電パネルの統計情報管理 Web アプリケーション

- 期間

2018 年 4 月～6 月

- 技術スタック
  - Backend
    - C#
    - .NET MVC
  - Frontend
    - Angular 1.x
  - Infrastructure

- PostgreSQL
  - IIS
- Others
  - SVN
- 開発内容

太陽光パネルを設置している顧客向けに発電量や CO2 削減量などを確認できる Web アプリケーションを開発しました。

エンジニアとして初めての業務で、最初のリリースは 4 月中と決められていたため頑張って開発をして期日に間に合わせました。

PostgreSQL 以外は初めてでしたが、業務中にキャッチアップを行って先輩に質問しながら開発を進めました。

リリース後はエンハンスでチャート画面の拡張などを行いました。

## 1.4 できること

### 1.4.1 Backend

- Laravel8 での API 開発
- PHPUnit を使ったテスト
- TypeScript/Express での API 開発
- jest を使ったテスト
- supertest や MSW を使った HTTP テスト
- Python での API 開発
- unittest や pytest を使ったテスト
- Go での API 開発
- gotest を使ったテスト

### 1.4.2 Infrastructure

- RDB でのテーブル設計
  - 物理設計はあまり詳しくありません。
- GCP でのインフラ構築
- BigQuery でのデータ操作
- Terraform での IaC
  - GCP は (k8s 周りを除いた) 一般的に使われるリソースは一通り経験しています。
  - AWS はあまり詳しくないです。

### 1.4.3 Frontend

- React or Next.js & TypeScript でのフロントエンド開発
- 1 からの簡単な環境構築

- webpack は全然分かりません。
- Vue.js でのフロントエンド開発
- jest を使ったテスト – mock を使うことができます。
- playwright や puppeteer などでの E2E テスト実装
- storybook でのカタログ作成とインタラクショナルテスト

#### 1.4.4 DevOop

- GitHub Actions や CircleCI での CI/CD 構築
- renovate 導入
- Docker & Docker Compose での環境構築

#### 1.4.5 Others

- 基本的な Git 操作
- デジタル広告の知識

### 1.5 今後の課題

- 大規模サービスの知見
  - スケーラビリティを考慮した設計
  - Observability
- AWS でのインフラ構築・運用
- 低レイヤの知識
  - 主にデータベースについての深い知識
- k8s
- アルゴリズム
  - AtCoder 緑相当
- 並行処理・並列処理
- チームリード経験

### 1.6 希望内容

#### 1.6.1 積極的にやりたいこと・できること

基本はサーバサイド~インフラにかけてを全体的に関わっていきたいです。

- (Web に限らず) サーバサイド開発
  - 最近は Golang を触ってますが、静的型付け言語なら基本は大丈夫です。
- IaC ベースのインフラ設計・構築と運用
  - メインは GCP ですが AWS でも対応します。
- 開発環境の効率化・自動化
  - Four Keys+信頼性の計測と可視化

\* エンジニアの生産性を可視化し、チームの生産性向上の指標とするのとエンジニアの評価を適正化する。

- トイルの撲滅
  - これまで様々な手作業を自動化してきています。
- データ基盤構築・運用
  - バッチを極力使わない、リアルタイム性と幂等性、各ユースケースに沿ったテーブル設計を追い求めたいです。
- 社内開発基盤（Developer Experience 改善みたいな）
- TDD ベース開発
  - テストコードを当たり前に実装する
  - TDD・BDD は設計手法であるという文化を醸成していく。
- レガシー業界をエンジニアリングで改善していく

#### 1.6.2 そこそこやりたいこと

- エンジニア採用
- レガシーシステム改善
- React/Next.js での Web フロントエンド開発

#### 1.6.3 興味はある

- BtoB Saas
- レガシー産業
- フロー効率開発
- 科学的なエンジニア採用

#### 1.6.4 できなくはないが積極的にはやりたくない

- Vue/Nuxt.js での Web フロントエンド開発
  - メインでなければ大丈夫です
- ピープルマネジメント・プロジェクトマネジメント
  - 苦手意識はありますが、部分的であれば挑戦したいと思っています

#### 1.6.5 やりたくないこと

- Web フロントエンドのマークアップ/スタイリング
  - 単純に知識不足です。直近での個人的な優先度も低いです。
- 社内独自フレームワーク
  - 良い思い出がないです。

#### 1.6.6 待遇

##### 1.6.6.1 最低希望年収 700 万

#### 1.6.6.2 その他（MUST 条件）

- フレックスタイム制導入
- リモート勤務導入
  - － 東京在住ですがフルリモートの場合はオフィス所在地は問いません。
  - － 都内であれば週 2 日が限度です
- SES/受託開発はすみませんが現在検討しておりません。

#### 1.6.6.3 その他（WANT 条件）

- フルフレックスタイム制導入
- フルリモート勤務導入
- 各種手当て
  - － 例）病床休暇、住宅補助