

2022年 2月 9日

本研究では、以前に作成した Vtuber Music という Web アプリケーションのデータを編集する Wiki として、Vbeing Wiki という Web アプリケーションを構築する。システムの複雑性に立ち向かうために、ドメイン駆動設計(以下、DDD)を活用する。DDD では、システム構築の対象となる領域のモデリングを行い、プログラムを4層に分けることで、従来のアーキテクチャが抱えていた依存の問題を解決する。

DDD では整合性や可読性の向上のためにコードやテストの量が増加するため、基本機能のみの実装でも 60 万字以上のコードを記述する必要があった。DDD は本研究のように一人での開発ではなく、チームでの開発において有用なツールであると考察される。

一方、Next.js を用いて SPA アプリケーションを構築したことで、ページ遷移速度が上がり、ユーザ体験は向上した。

図 1 はインフラの構成図であり、アプリケーションは全て AWS 上に構成されている。左側の Fargate にデプロイされた NestJS アプリケーションがバックエンド、右側の Amplify にデプロイされた Next.js アプリケーションがフロントエンドである。バックエンドへのリクエストは ALB によって複数の AZ にあるコンテナに分散され、可用性を高めている。

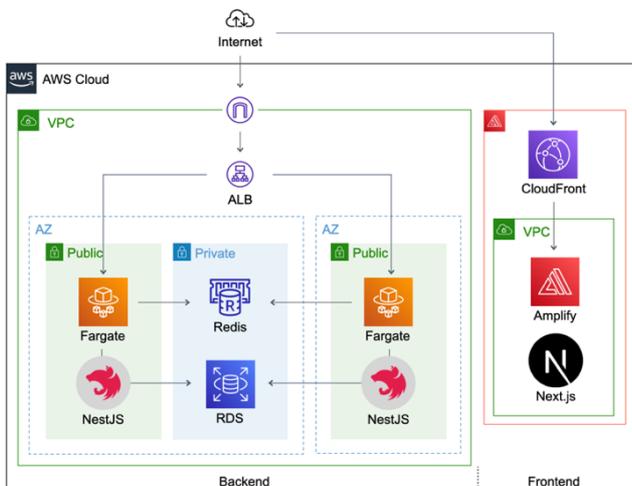


図 1 インフラストラクチャの構成図

本研究で開発した Vbeing Wiki では、ユーザはログインし、各種データを追加・編集できる。編集可能なデータ属性には、Vtuber の名

前・年齢・Twitter ユーザ名などがある。編集ログは属性単位で記録され、編集内容・編集日時・編集ユーザの確認や、変更の取り消しができる。

Vtuber 個別ページの UI を図 2 に示す。Vtuber のデータ属性にはテーマカラーが保持され、個別ページのアクセントカラーに反映される。図 2 はテーマカラーが赤色の例である。



図 2 Vtuber 個別ページのスマホ用 UI

本研究では、表 1 に示す技術を用いることで、ユーザと開発者にどのようなメリットがあるのか考察する。なお、バックエンドとフロントエンドは全て TypeScript でコーディングしている。TypeScript で統一する利点として、学習コストを抑えられる点、型を共有できる点、対応しているクラウドサービスが多い点がある。

表 1 使用した技術・思想一覧

名称	説明
TypeScript	JavaScript のスーパーセット
GraphQL	クエリ言語
NestJS	Node.js フレームワーク
AWS	クラウドサービス
Next.js	React のフレームワーク

Vbeing Wiki は公開していないが、必要な機能を実装したのち公開予定である。