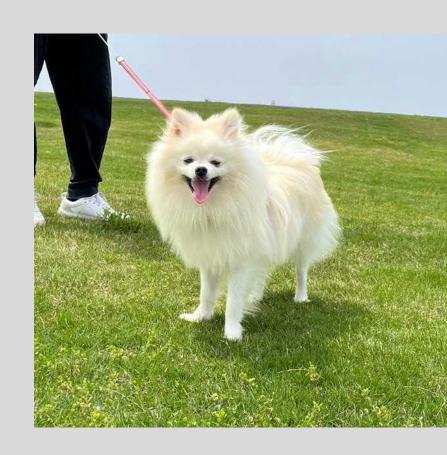
Full TypeScriptだから 実現できる世界線

はじめに

- ・組織と技術選定みたいな話です
- あくまでひとつの例としての考え方です
- 少しでもみなさんのヒントになれば幸いです

自己紹介

- 名前:k-ichirof (X:@k1rof18)
- ・所属:株式会社ユーザベース
 - F#, Go, Scala, TypeScript...
- 経歴:規模小さめの現場を複数経験
 - フロント・バックエンド・インフラの経験あり
- 好き
 - ・ポメラニアン・ゴルフ・漫画(キングダム)



[前提] Full TypeScript とは?

今回は、フロント・バックエンド・インフラに至るすべてを TypeScriptで開発することをFull TypeScriptとします。

Full TypeScriptって良いの?

結論:コンテキストによる!!!

コンテキストを考える



- ・規模小さめの組織
- ジュニアエンジニア多め
- エンジニアがフルスタック的に動く
- 個人開発でも良さそう



- ・大規模組織/採用力がある
- シニアエンジニア多め
- 色んな言語・技術に挑戦したい

Full TypeScriptではない事例

- フロント: TypeScript
- バックエンド: Go
- インフラ: Terraform

- 各チーム1人
 - ・ひとり辞めたら…?
- ・少人数で開発を進めながら、キャッチアップをする
 - ・業務時間で可能…?

Full TypeScriptにした事例

• フロント:React

バックエンド: Express

インフラ:AWS CDK

• その他: pnpm (workspaceモードでモノレポ)

→「圧倒的な」立ち上がりの早さ

・環境構築がカンタン! (コマンド1つでフロント・バックエンドを立ち上げる)

→ キャッチアップの負荷軽減

- 領域に特化したキャッチアップが可能になる
- ・エンジニア歴1年ほどの、バックエンド未経験者が初日からPRを出せた!

▲ フロント・バックエンドでコードを共通化をするのは難しい

- 破壊的変更を防ぐ、責務分割など気をつけないと複雑性を生み出しかねない
- リリースの単位を小さくするのが難しくなるなど...

Full TypeScriptにした結果

- フルスタック的な動きを取る環境を用意しやすい
- 「分業しない」ことでエンジニアとして幅が広がる
- ペアプロがやりやすい
- => エンジニアの成長

- •「人」が抜けた時のリスクを軽減できる
- TypeScriptの求心力(?)による採用力向上
- => 事業の存続・成長

Full TypeScriptだから実現できる世界線とは?

(コンテキストによるが)

エンジニアの成長 事業の成長 を両立 させることができる(可能性が高まる)

ご清聴 ありがとうございました!