

Webフロントエンドと アーキテクチャ事情の持論を喋る

～我々は JavaScript エンジニアではないという強い気持ちを添えて

@ahomu (Ayumu Sato) | TechFeed Experts Night vol.4 2022.09.21

@ahomu

- TechFeed Official Expert No.25
- Web Frontend Engineer, vPoE
 - Offers | overflow, inc.
- Author of #超速本
 - w/@1000ch



前回のおさらいと今回のこと

TechFeed Conference 2022

- SPA vs MPA の構図はもう無理
 - CSR と SSR の局所使い分けが実状
 - フレームワークサポートも多様化
 - もはや設計具体ではなく偏った思想
- 設計のキモはどこにあるのか
 - HTML、キャッシュ、JavaScript
 - 色んなアプローチがあるよね
 - 今日はこっちを主に再編してお届け

検索… (⌘ + K)

SPA/MPA 議論の俯瞰と 現代における設計のポイント

Web フロントエンドの設計に銀の弾丸はありません!!

@ahomu (Ayumu Sato) | 2022.05.14 TECHFEED CONFERENCE

連載: TechFeed Conference 2022

SPA/MPA 議論の俯瞰と現代における設計のポイント
む) — TechFeed Conference 2022講演より

【書き起こし】Webフロントエンドの設計について基本的な観点のおさらいから近年のSPA/MPAにまつわる
について5分でどこまで圧縮できるかチャレンジします。

JavaScript

<https://techfeed.io/entries/62a8534290ee1f2ca98b1a2a>

今回のアジェンダ

- フロントエンドとは
- フロントエンドアーキテクチャとは
- アーキテクチャ設計の要点
- 実例を交えたシミュレーション
- まとめ

ではやっていきましょう

Frontend ?

ユーザーとサービスを UI で繋ぐために
HTML 生成処理を整える技能領域

OR/AND

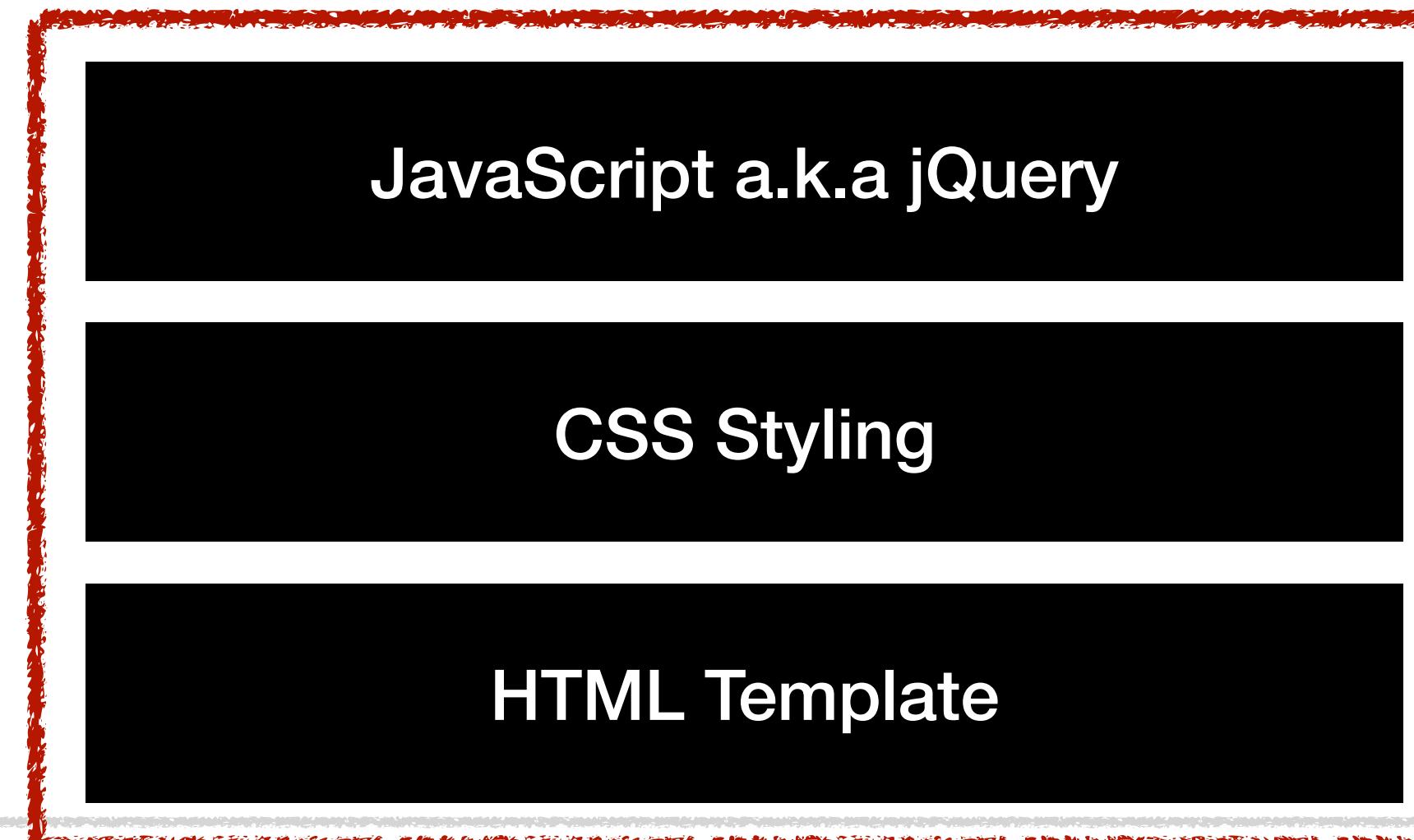
システムとデザインの境界で責任を持ち
ユーザーに届ける品質を司る職能

Decays without doing anything

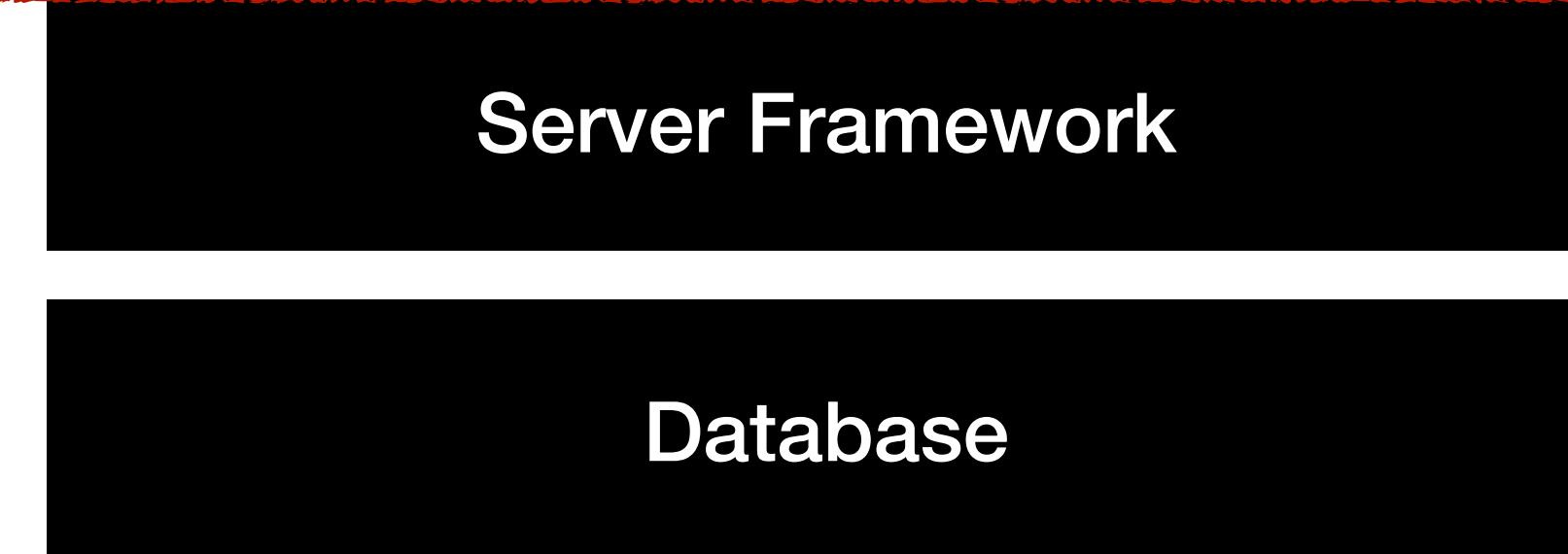
- 通信環境、デバイス、ユースケース、デザイントレンド etc
- すべて時間の進み、技術の歩みと共に変化し続けている
- クライアントサイドはその変化を直接その身に受ける
- なにもしなければユーザーエクスペリエンスも開発体験も朽ちていく
- だからこそ、フロントエンド技術は How をアップデートし続けている

Transitions of Frontend

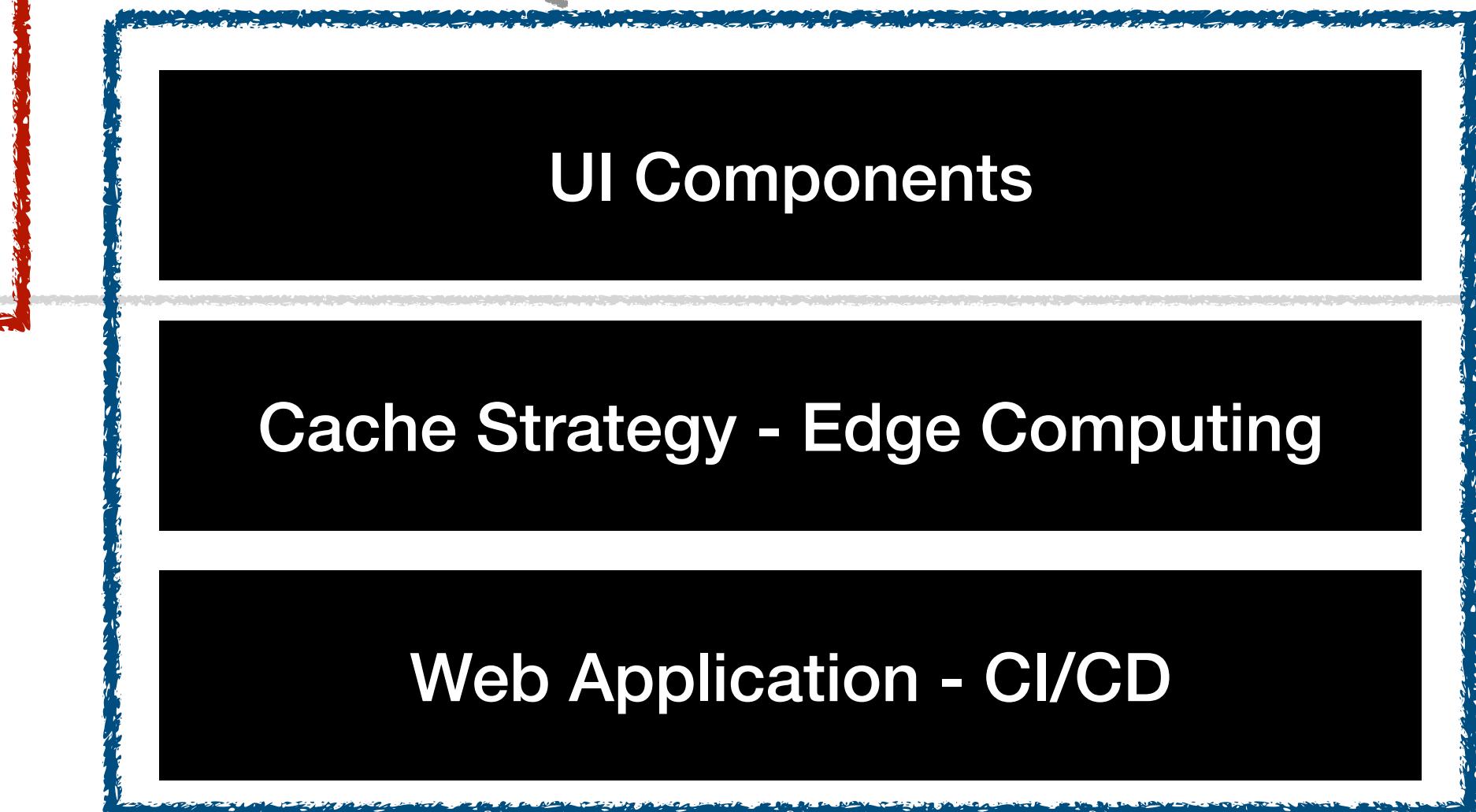
Client Side



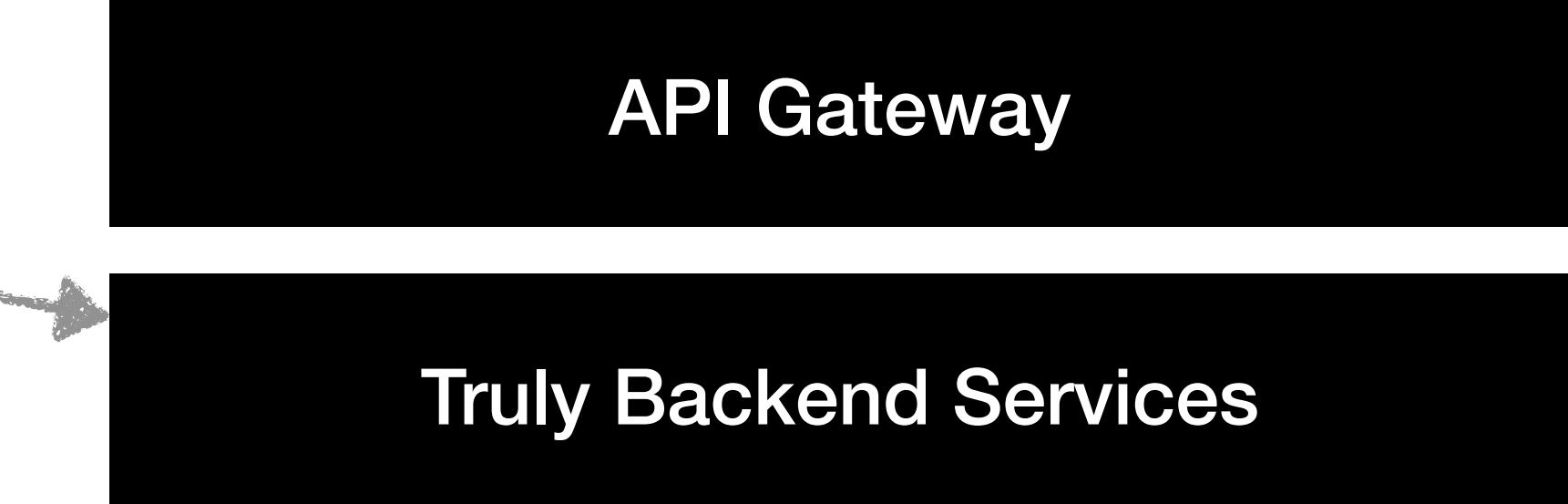
Server Side



Node.js の台頭、フルスタックフレームワーク普及
いわば JavaScript エンジニアを経て
Web アプリケーションエンジニアへの変化



バックエンドエンジニアが提供する価値の中心も
Web アプリケーションの後方にシフトしている
モバイルアプリの存在も API 前後の線引きに影響？



Web Frontend Architecture

- ソフトウェアにとって性能（速さ）は価値そのもの
- アーキテクチャは性能と向き合っている
- 一方で、開発体験/効率の名の下に保守性や生産性とも向き合っている
- 性能と開発体験の天秤を揺らし続けているのが近年のフロントエンド

Modern History

(記憶が定かでないので順不同)

- 各種ライブラリの台頭によりコンポーネント志向開発が本格的に普及
- Node.jsを中心としたビルドプロセスとエコシステムが重層的に発達
- 1つのコードベースでServer / Clientに対応する実装(SSR + CSR)
- Server → Clientで状態を引き継ぐためHydrationが誕生
- JavaScriptですべてを解決する開発者体験の御旗(e.g. CSS in JS)
- サバクラ共通化されたコードベースにnpmでライブラリがマシマシ

Seriously bloated JavaScript

- 肥大化の一途を辿ったクライアントサイドで実行される JS
 - ファイルサイズによる Networking コストの問題は当然ある
 - しかしディスクキャッシュ経由でも生じる Computing コストのほうが深刻
- JS を Web の CO₂ と喩えられるのも言い得て妙
 - 開発者体験を支えるため JavaScript に依存しがちな Web 開発の現状
- Hydration 課題はきっと Yosuke Furukawa or/and りい-san が喋る

設計の要点

どこで HTML を生成するか
どのようにキャッシュ戦略を描くか
どれくらい JavaScript を載せるか

どこで HTML を生成するか

- サーバーサイド
 - 特に静的なコンテンツならサーバーサイド + CDN が最速で HTML を返すのが最強
- クライアントサイド
 - とはいえ、インタラクティブなサービスはクライアントサイドでも生成が必要になる
- エッジサイド
 - 近年、CDN～クラウド業者を中心にエッジで諸々を処理できる環境が整いつつある

どのようにキャッシュ戦略を描くか

- キャッシュ自体はあらゆる箇所に差し込む余地がある
 - サーバーアプリケーション（データベースとかオンメモリとか）
 - ミドルウェア（nginx proxy cache とか、Varnish とか）
 - CDN（キャッシュだいすきコンテンツデリバリーネットワークさん）
 - ブラウザ（ローカルキャッシュ、ServiceWorker CacheStorage とか）
- 鮮度 & 整合性と戦いつつも HTML は CDN に食わせたいのが本命
 - CDN は転送距離の課題も解決するので頭ひとつ抜けている
 - ストリーミング動画ファイルもゲームのパッチファイルも CDN が大好きサ

どれくらい JavaScript を載せるか

- JavaScript のバンドルサイズは、クライアントサイドの仕事量と比例
 - 極端な話、JavaScript ゼロで HTML 最速返却しまくるだけで優勝可能
 - 現実にはそもそもいかないのでバンドル分割や減量に励むわけである
- ヘビーになりうるパラメータ
 - インタラクションが必要な UI の有無、およびその複雑性
 - CSR の有無
 - フルスタック SPA フレームワークの(予防的)導入の有無
 - Hydration の有無

**提供物の理想状態にあわせて
これらの要点を総合的に判断する**

シミュレーション

Offers Magazine

- 静的なオウンドメディア
 - リアルタイム性が乏しい
- インタラクティブ性は極低
 - ログイン依存処理もほとんどない
- どういうHowがあるだろうか
 - HTML は SSR ? 事前静的生成 ?
 - キャッシュは CDN に置けそう ?
 - CSR + Hydration は本当に必要 ?

The screenshot shows the homepage of Offers Magazine. At the top, there's a navigation bar with links for 'Offersとは', '副業案件を見る', '副業事例を見る', '採用担当者様はこちら', 'ログイン', and '新規登録はこちら'. Below the navigation is a search bar. The main content area features a 'PICK UP!' section with three articles. The first article is about a digital minister CTO, the second is about a company SOWOW CTO, and the third is about a medical SaaS developer. Below this is a large banner for 'Offers' with the text '週1/フルリモートで副業を始めるならOffers' and a call-to-action button '今すぐはじめる!'. To the right of the banner is a 'GitHub連携' button. The bottom section has a '新着記事' (New Articles) grid. One article is about a medical SaaS developer using Offers. Another article discusses ML engineers being recruited through Offers after three interviews. There are also sections for 'FOR WORKER' and 'FOR RECRUITER'. On the far right, there's a 'ランキング' (Ranking) section with a laptop icon.

副業・兼業に役立つ
Offers Magazine

Offersとは 副業案件を見る 副業事例を見る 採用担当者様はこちら ログイン 新規登録はこちら

副業・フリーランス プログラミング インタビュー デザイン お金 採用・組織

PICK UP!

デジタル庁 CTO / 共性デジタル庁 CTO 藤本氏に聞く、行政DXの面白さや開発組織の展望。複業を決めた理由とは？

式会社ソウゾウ CTO 名村卓【未来】エンジニア個人や開発組織がスケールするために今必要なこと

医療・介護系SaaSの開発組織をOffersで構築へ！エピグノが実践する業務委託の採用手法とは？

Offers
週1/フルリモートで副業を始めるならOffers

今すぐはじめる！

まずはGitHubを連携する！

新着記事

もっと見る >

FOR WORKER 副業したい方はこちら

FOR RECRUITER 採用担当者様はこちら

ランキング

Offers 候補者検索

- ログイン前提のサービス
 - HTMLはキャッシュしようがない
- 検索など動的なUIが多い
 - SPAっぽくする余地がある
- どういうHowがあるだろうか
 - HTMLのキャッシュは無理そう？
 - 割りきってSPAっぽく構成する？
 - 肥大化しうるJSをどう最適化する？

The screenshot shows the 'Offers' candidate search interface. The top navigation bar includes links for '求職者検索' (Candidate Search), '自動リストアップ' (Automatic List Up), 'メッセージ' (Messages), '採用管理' (Recruitment Management), '求人・案件一覧' (Job/Case List), a notification bell icon, and user profile icons.

The main area is titled '候補者一覧' (Candidate List) with a note '(候補者の1-25件を表示中 / 合計24246件)'. It features a search condition sidebar on the left with fields for '年齢降順' (Age Descending) and a 'リセット' (Reset) button. The sidebar also lists saved search conditions:

- エンジニアスキル: HTML
- エンジニアスキル: CSS
- エンジニアスキル: JavaScript
- エンジニアスキル: Webpack
- エンジニアスキル: TypeScript
- 専門職: フロントエンドエンジニア

Below the sidebar is a 'この条件で検索' (Search with these conditions) button. The main list displays five candidate profiles:

求職者名	マッチしたスキル	所持スキル	特徴タグ	SNSつながり	タグ
デザイナー ID: 80 94歳 未設定 良い仕事があれば 今は考えていない	CSS 50 Web Design 50 JavaScript 50 ビジネス思考 50				+ タグを追加
エンジニア ID: 44 92歳 未設定 今すぐ働きたい 今は考えていない					+ タグを追加
エンジニア ID: 50 92歳 未設定 今すぐ働きたい 今は考えていない					+ タグを追加
TONY 91歳 未設定 今すぐ働きたい 半年以内に考えて いる	Ruby 50 CSS 50 Sass 50	1年以上 3年以上 3年以上			+ タグを追加
TONY UTM REFERER 91歳 未設定 今すぐ働きたい すぐ転職したい	JavaScript 50 Python 50 Ruby 50 Node.js 50 Go 50 TypeScript 50 Sass 50	1年以上 3年以上 5年以上 3年以上 1年以上 1年以上 1年以上			+ タグを追加

At the bottom right, there are buttons for 'オファー数上限: 0 / 50' (Offer limit: 0 / 50) and '候補者登録' (Candidate Registration).

まとめ

サマリ

- フロントエンドは Web アプリケーション全域が守備範囲になりうる
- 性能と開発体験の間で揺られるなかで設計上の課題も生まれている
- 設計のポイントは、HTML生成・キャッシュ戦略・JavaScript 積載量
- アプリケーションの要件にあわせた設計を！

**When solving a problem of interest, do not
solve a more general problem as an
intermediate step.**

Statistical learning theory by Vapnik, V. N, 1998

ありがとう
ございました！

複業転職マッチングプラットフォーム Offers もよろしくね！