



ELEVENBACK

私たちはなぜ SPA で開発するのか

2019.11.02 #frontkansai #A5 Takuma HANATANI @ ElevenBack LLC.



# Takuma HANATANI (@potato4d)

## Status

- Senior Front-End Engineer at LINE Corp.
- CEO at ElevenBack LLC.

## Core skills

- Front-End and Cloud Native Engineering
- User Interface and Interaction Design

## OpenSource Ecosystem

- React.js / Vue.js ja document maintainer
- Ex-Nuxt.js ja document maintainer
- Nuxt.js beginners guide (C&R Institute) author



<https://uit.connpass.com/>



<https://uit-inside.linecorp.com/>

東京・福岡で展開中の LINE メンバーが主導のイベント・Podcast コミュニティ「UIT」



LINE UIT室の開発者による「最新のフロントエンド」をキャッチアップできる Podcast

ep.26 非同期処理における Pending と Empty の取り扱いの話



2019/10/17

ep.25 Vue Fes Japan 2019 直前トーク「最近の Vue.js どうですか？」



2019/10/10

ep.24 Firebase for Indies Web-App Developers



2019/10/04

ep.23 UIT meetup 「集まれ！（タブン）実務では使わないフロントエンド芸発表会」 After show



2019/08/29

ep.22 福岡の フロントエンド 生の声 技術顧問とコミュニティ 【Guest: @mya\_ake】



2019/08/16

ep.21 UITエンジニア エディタの流儀 - あるいはただのエディタ愛



2019/08/06

ep.20 Hello LINE! UIT新卒エンジニア座談会



2019/07/04



ex

UIT INSIDE では「Reactの設計」や「現場での非同期処理」など LINE メンバーの生の声が沢山！

Candy

BETA

# Candy

## フリーランス・パラレルワーカー向けの 「気にしない」モバイル会計管理クラウド

<https://candy.ac/campaign/beta>



POWERED BY  
ELEVENBACK

カレンダーや自動化によって会計管理から解放されるクラウド「Candy」ベータ提供中

# 今日の想定聴講者

- 現状 SPA 開発を行っている人
- その中でも、特に技術選定、プロダクトの今後に付き合う機会がある人

Q.

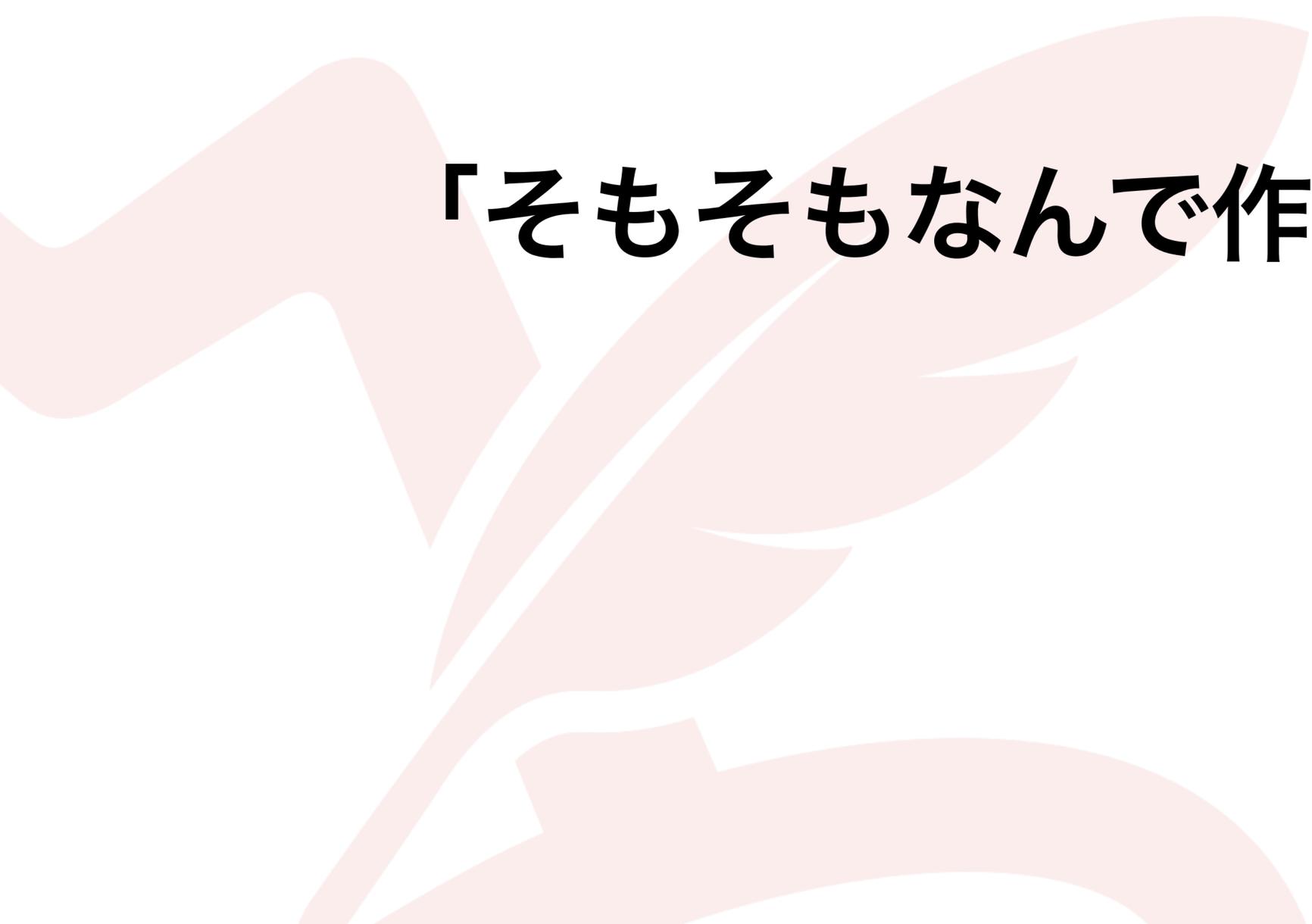
**SPA を作ってご飯食べていますか？**

Q.

**なんで SPA で開発していますか？**

# 今日考えたい3つのポイント

- SPA についての現状の把握
- なぜ私たちは開発する際に SPA を選定しているのか、してしまうのか
- SPA とオーバーエンジニアリング、使い所についての考察

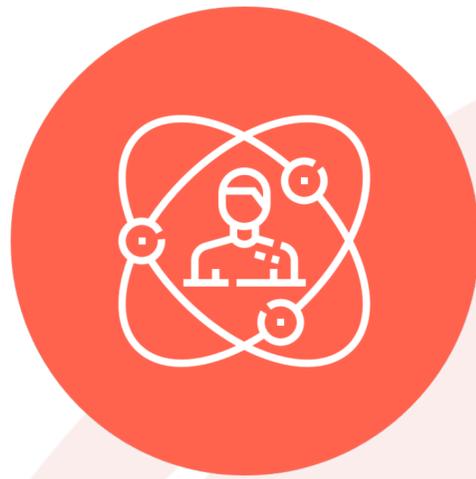
A large, abstract graphic in a light pink color occupies the left and bottom-left portions of the image. It consists of several overlapping, rounded shapes that resemble stylized leaves or petals, creating a layered, organic effect. The text is centered horizontally across the middle of the image, partially overlapping the pink graphic.

「そもそもなんで作ってんだっけ？」

# SPA 選定を構成する要素について

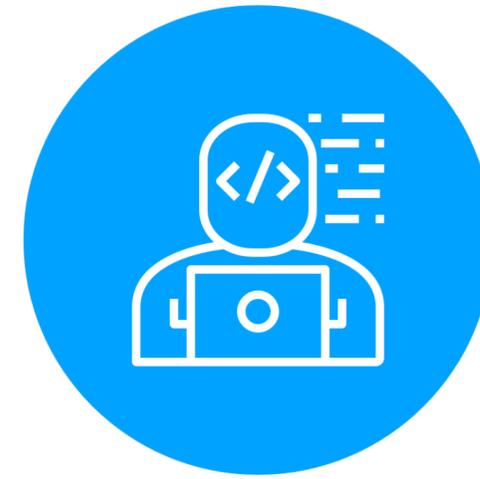
- 「そもそもなんでSPAで作ってるんだっけ？」
  - 「それが普通だから」「デファクトだから」
  - 「jQuery とか Backbone が辛かっただけでしょ」
  - 「アニメーションの実現のために必須じゃない？」
  - 「そもそも 2019 年に SPA 以外作りたくない」
- **全て理由の一端ではあるが、実はもっと簡単に分割できる**

# SPA が選ばれる本質的な理由



## for User Experience

サービスの理想的な体験を Web で実現するため



## for Developer Experience

開発効率を向上、結果的にプロダクトの提供速度を早めるため

# SPA 技術の選定を構成する2つの要件

- **UX 要件による SPA 選定**

- サービスの理想的な体験を Web で実現するために選ぶ必要のある場合に行われる選定

- **DX 要件による SPA 選定**

- 開発の体験を改善し、開発効率を向上、結果的にプロダクトの提供速度を早めるための選定

# SPA 技術の選定を構成する2つの要件

## • UX 要件による SPA 選定の特徴

- UI に関する部分を強化することで総合的なユーザー体験を強化する
- 事業要件と密接に関わるため、必要な要件に合わせて SPA が選定される
  - Web かつネイティブにあるような機能を PWA として提供したい
  - ユーザーの行動による画面遷移をリッチかつシームレスにしたい
  - 外部 API と連携して動的で SEO フレンドリーなサイトを作りたい

# SPA 技術の選定を構成する2つの要件

- DX 要件による SPA 選定の特徴

- 開発者の生産性を高めることによって副次的な作用を得るための選定
  - 最近話題。退職ケア、市場での競争力などメタを大きく含むことに注意
- 必然性に乏しいことが多い、以下のような目的で SPA になることが多い
  - 静的 Web サイトの更新を手動で行いたくない
  - JSX や Vue テンプレート、CSS Modules や Scoped CSS が欲しい

# SPA 技術の選定を構成する2つの要件

- **UX 要件による SPA 選定**

- サービスの理想的な体験を Web で実現するために選ぶ必要のある場合に行われる選定

- **DX 要件による SPA 選定**

- 開発の体験を改善し、開発効率を向上、結果的にプロダクトの提供速度を早めるための選定

# SPA 技術の選定を構成する2つの要件

- **UX 要件による SPA 選定**

- サービスの理想的な体験を Web で実現するために選ぶ必要のある場合に行われる選定

- **DX 要件による SPA 選定**

- 開発の体験を改善し、開発効率を向上、結果的にプロダクトの提供速度を早めるための選定

**チームがどちらの意味で SPA の技術を採用しているかを把握しなければならない**



# UX要件の色が強い事例



# Creating media without an app

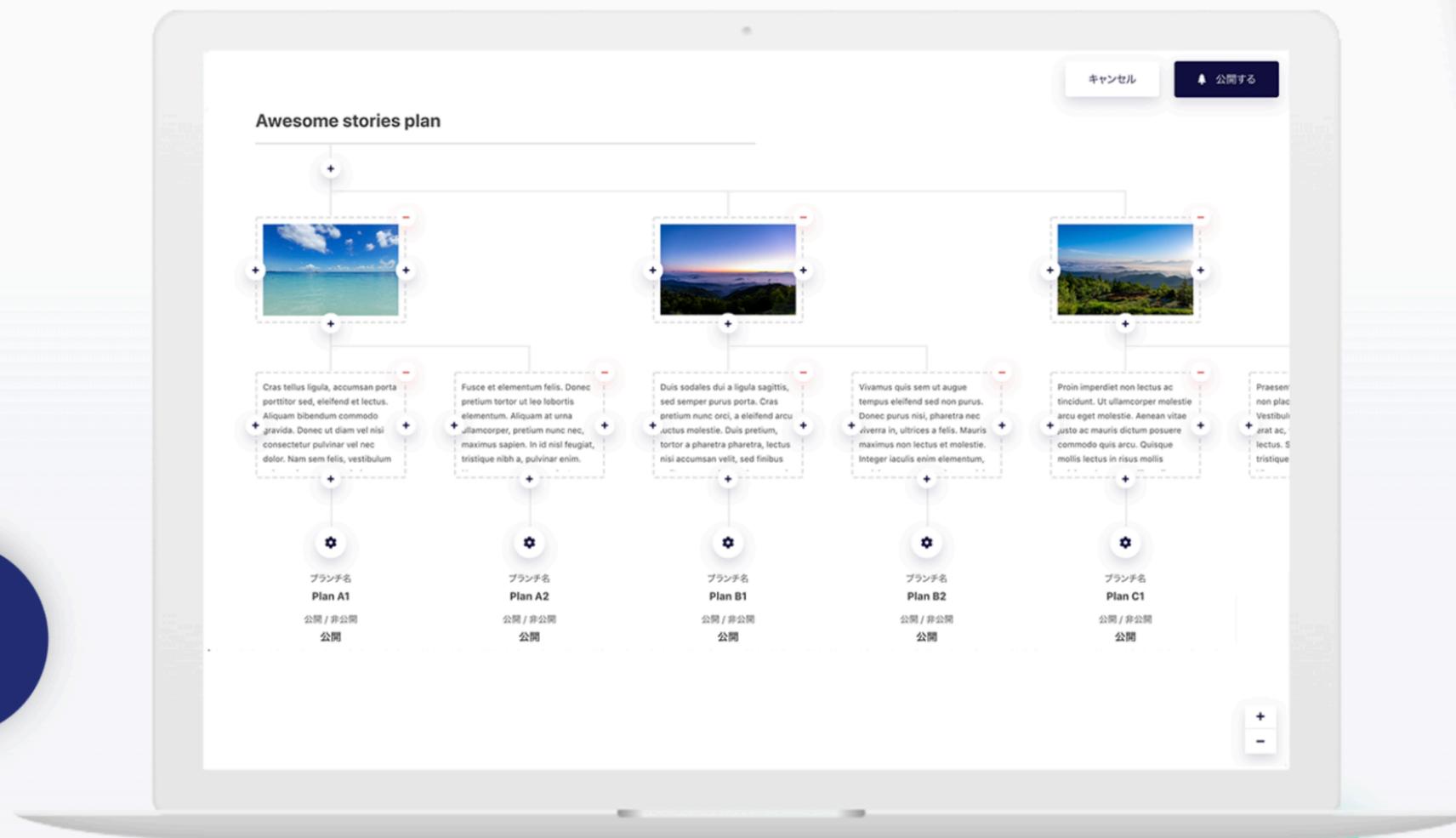


<https://www.youtube.com/watch?v=UTZVXlcUK1w>

国際的な有名な事例でいえば、Instagram も新興国向けの高速かつ軽量な体験のために PWA / SPA を開発した

# ツリー状に書く マークダウン ドキュメント

文章の整理や思考の深掘りをして  
アウトプットの質を上げよう



© 2018.03.19

# Web サイト制作にこそ Nuxt.js がベストマッチである理由

■ Nuxt.js

146 B!ブックマーク 0 シェア ツイート LINE 1 送る Pocket

業務委託でSCOUTERのフロントエンド周りの開発をお手伝いしている、花谷拓磨 (id:potato4d) です。

SCOUTER は、先日自社のコーポレートサイトを Nuxt.js + WP REST API 構成にてリニューアルいたしました。

普段であれば SPA や SSR が必要な高度なアプリケーションなど、主にWebアプリケーション開発の文脈で語られる事が多い Nuxt.js ですが、何故Webサイトであえて Nuxt.js を採用したのか、そしてそのメリットから、実際に利用しての課題点まで、制作を担当した私自ら知見を存分にご紹介したいと思います。

## 新しいコーポレートサイトのご紹介



## 検索

記事を検索



## 最新記事

...が意外に便利だった件

Laravel/Vue.js勉強会#11を開催しました！

macOS Catalinaにアップデートしたらphpenv installでビルドエラーしたのでhomebrew phpつかうようにした

Inversifyを使った、型堅牢なDIコンテナの構築

NuxtMeetUp#9 オールスターズを開催しました

## 月別アーカイブ

▶ 2019 (53)

▼ 2018 (41)

2018 / 12 (9)

2018 / 11 (3)

2018 / 10 (5)

2018 / 9 (4)

2018 / 8 (6)

2018 / 7 (2)

2018 / 6 (3)

2018 / 5 (3)

2018 / 4 (1)

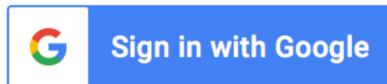
2018 / 3 (3)

<https://techblog.roxx.co.jp/entry/2018/03/19/115229>

SCOUTER(現: ROXX)株式会社の旧 Web サイト。ブランディング要件、SEO、オウンドメディア提供のため採用。

この度はフリーランス・パラレルワーカーのための気にしない会計管理クラウド「Candy」に興味を持っていただきありがとうございます。  
現在 Candy はクローズドベータの利用申請を受け付けております。ご利用をご希望のかたは、下記フォームより申請ください。  
通常当日中～2営業日の審査後、アカウントが発行されます。

## ベータテストに利用する Google アカウント



## 利用者氏名(もしくは屋号)

## ご職業

## 月間の請求書送付数

選択してください

## 日常的に利用する会計書類

- 見積書
- 請求書
- 納品書
- 領収書

## 早期利用希望

- 翌営業日までの利用開始を希望する
- [利用規約](#)・[プライバシーポリシー](#)に同意する

<https://candy.ac/campaign/beta/>

弊社サービス Candy も PWA としての体験の提供や、Bottom Barなどの Native Like な UI のために採用。



# DX要件の色が強い事例

資金の偏りによる  
機会損失のない世界を創造する

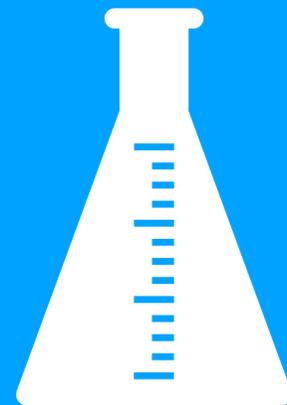
ニュース

[一覧を見る](#)

2019年10月16日 ▶ [【メディア出演】](#) テレビ朝日のバラエティ番組「あいつ今何してる？」に、代表 後藤が出演いたしました

<https://corp.payme.tokyo/>

コーポレートサイトを Nuxt.js の静的サイトで開発した事例。NEWSやバリデーションの更新容易性のために採用。



とある医療系 SPA の開発では「APIファーストで開発されていた」ため Vue.js の SPA を採用。



# SPA 技術をどこまで使うかの技術選定

# そもそも SPA が必要か？必要でも全部必要か？

- そもそも SPA の技術が必要か？
  - これまでの文脈を考えると SPA の技術が必要であることは間違いない
- SPA の技術に付随する技術が全て必要なのか？
  - 例えば React / Vue.js に常に Redux / Vuex は必要か？
    - 認証だけならリアクティブな object、モーダルフラグ制御は portal でも
  - なんでもできる開発環境は必要か？フルスタックである必要はあるか？
- **SPA にするにしても、常に全ての技術を総動員する必要はない**
  - だからといって不足もまずい。**アンダーもオーバーも等しく毒**になる

# EnterpriseQualityCoding / FizzBuzzEnterpriseEdition

👁 Watch 182
★ Star 11.9k
🍴 Fork 542

- 🔗 Code
- 🔔 Issues 279
- 🔗 Pull requests 29
- 📁 Projects 0
- 📖 Wiki
- 🛡 Security
- 📊 Insights

Branch: master ▾

[Create new file](#)
[Upload files](#)
[Find file](#)
[History](#)

[FizzBuzzEnterpriseEdition](#) / [src](#) / [main](#) / [java](#) / [com](#) / [seriouscompany](#) / [business](#) / [java](#) / [fizzbuzz](#) / [packagenamingpackage](#) / [impl](#) /

<b>Tim Kellogg</b> FizzStringReturner may not have copied all characters <span>⋮</span>		Latest commit 16d9d00 on 4 Feb 2017
..		
📁 <a href="#">factories</a>	Fix whitespace	3 years ago
📁 <a href="#">loop</a>	Fix whitespace	3 years ago
📁 <a href="#">math/arithmetics</a>	Merge pull request #234 from askogvold/gradle	3 years ago
📁 <a href="#">parameters</a>	Code style issues - explicit super() call, introduced constants from ...	3 years ago
📁 <a href="#">printers</a>	Increase code coverage (now 91%).	3 years ago
📁 <a href="#">strategies</a>	Fix whitespace	3 years ago
📁 <a href="#">stringreturners</a>	FizzStringReturner may not have copied all characters	3 years ago
📁 <a href="#">visitors</a>	Reverting the revert commit, since it clearly did not revert as inten...	4 years ago
🔥 <a href="#">ApplicationContextHolder.java</a>	Fix whitespace per PR review comments.	3 years ago
🔥 <a href="#">Constants.java</a>	Fix whitespace	3 years ago
🔥 <a href="#">Main.java</a>	Fix whitespace	3 years ago
🔥 <a href="#">StandardFizzBuzz.java</a>	Fix whitespace	3 years ago

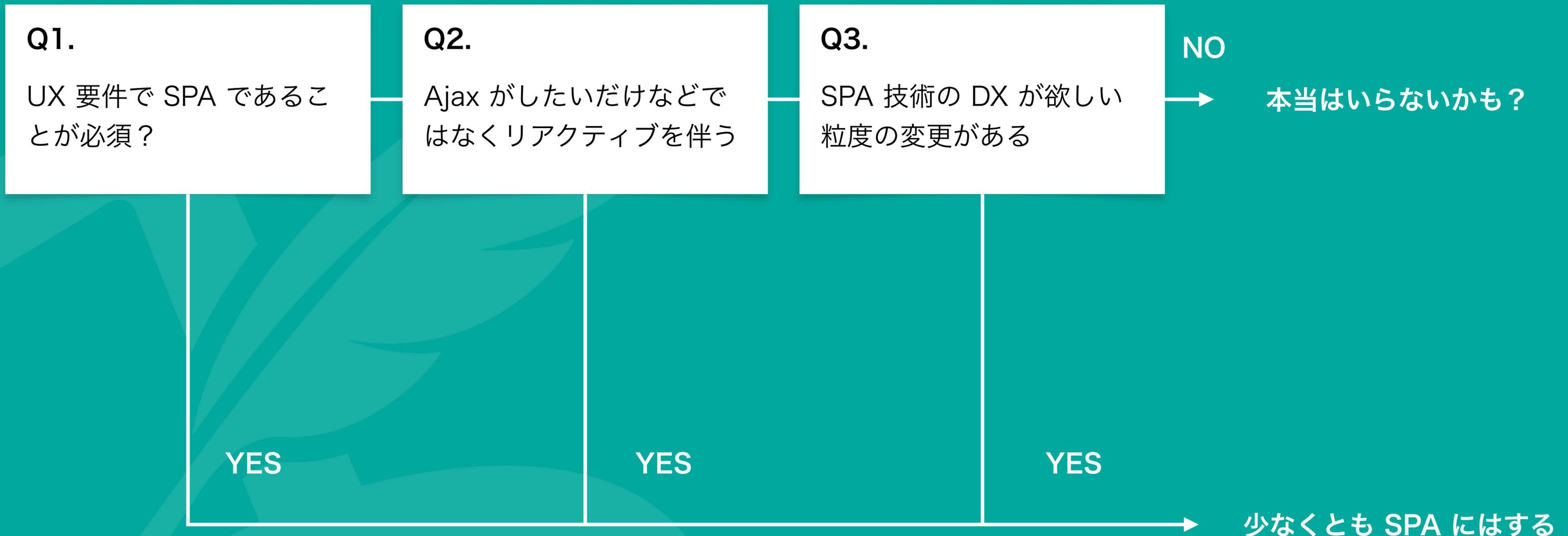
<https://github.com/EnterpriseQualityCoding/FizzBuzzEnterpriseEdition>

目的に対して過度な技術選定を行うと、複雑度が増すだけの「FizzBuzz Enterprise Edition」になってしまう

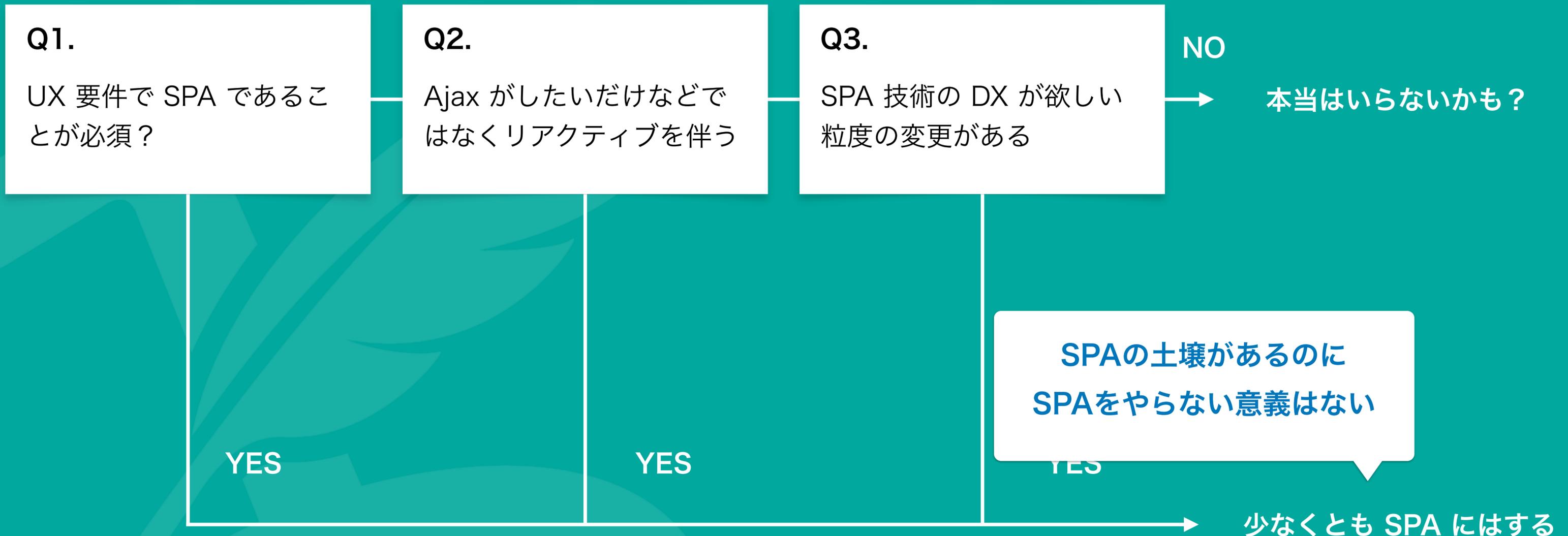


「SPA」と「状態管理」で考えてみる

# その SPA 本当はいらなくても？チャート

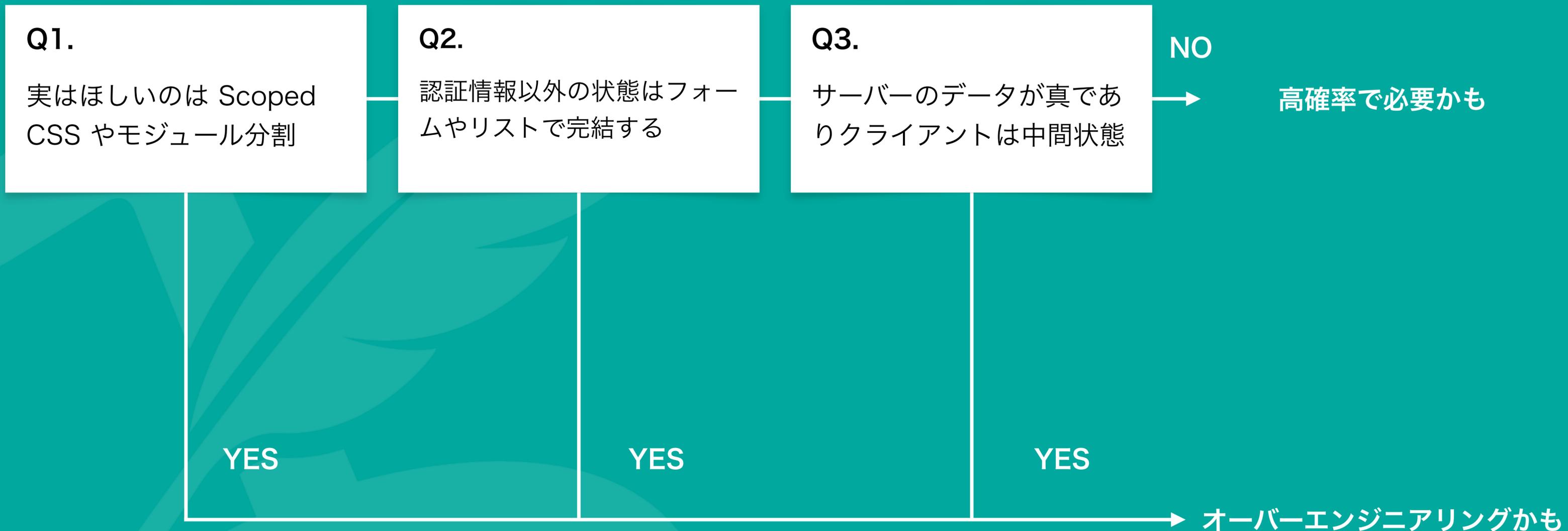


# その SPA 本当はいららないかも？チャート

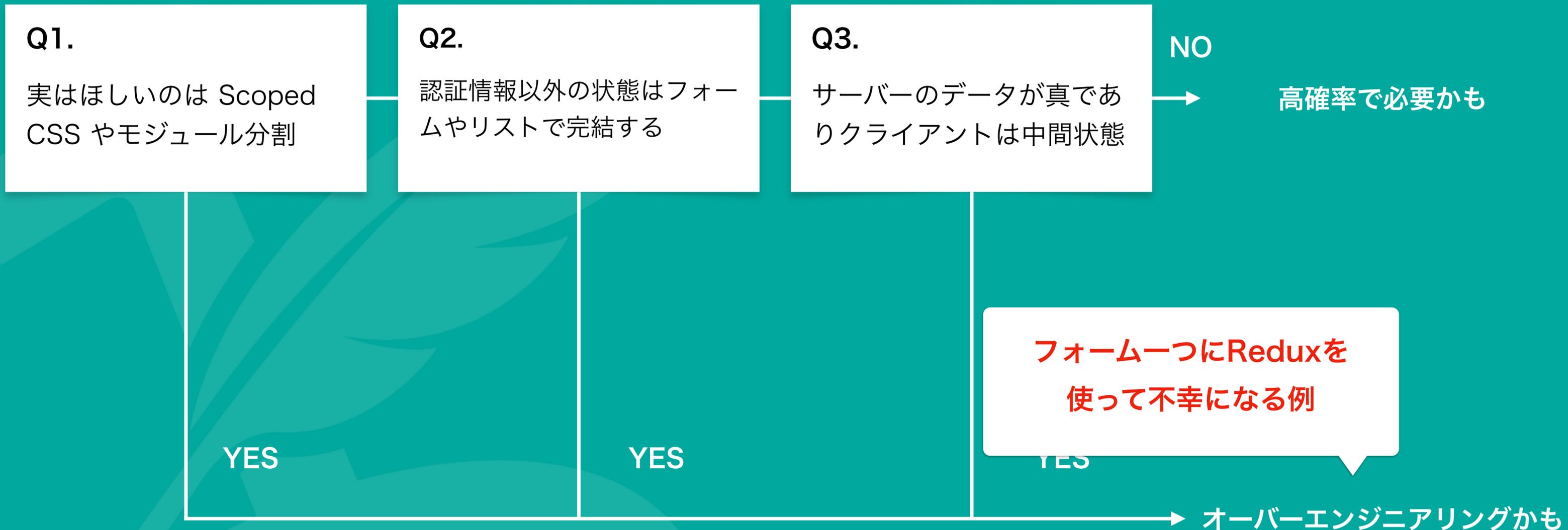




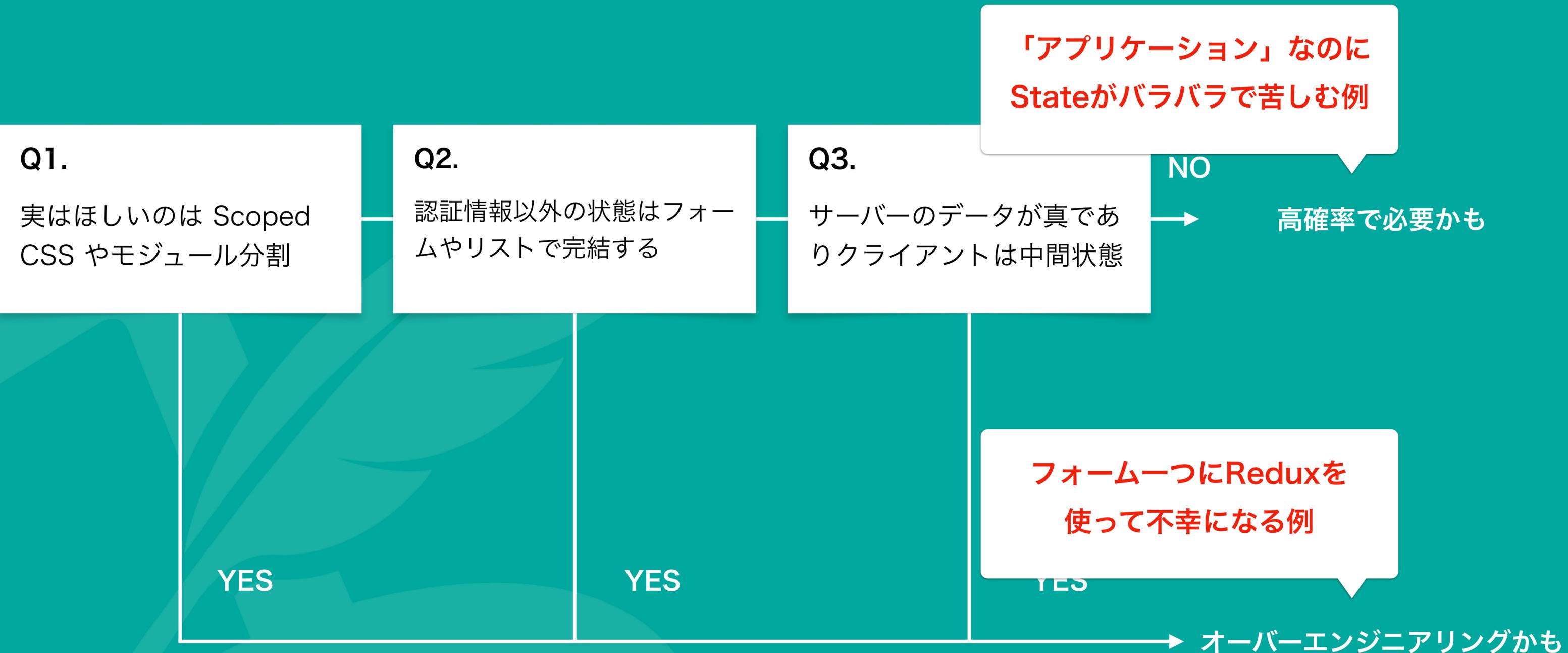
# その State 管理本当はいらないかも？チャート



# その State 管理本当はいらないかも？チャート



# その State 管理本当はいらないかも？チャート



# その State 管理本当はいらないかも？チャート

「アプリケーション」なのに  
Stateがバラバラで苦しむ例

Q1.

実はほしいのは Scoped  
CSS やモジュール分割

YES

Q2.

認証情報以外の状態はフォー  
ムやリストで完結する

YES

Q3.

サーバーのデータが真であ  
りクライアントは中間状態

NO

高確率で必要かも

フォーム一つにReduxを

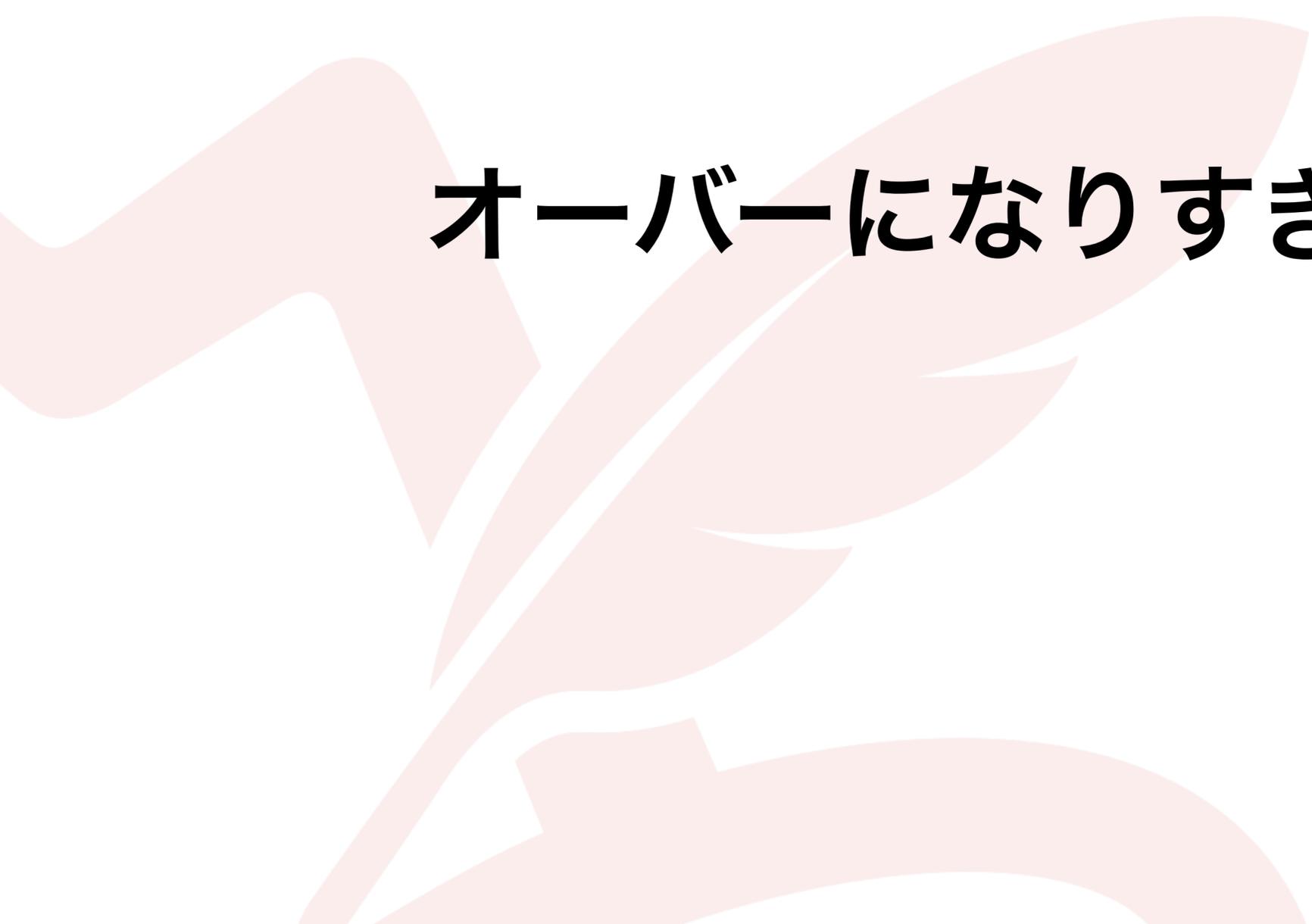
余談：使って不幸になる例

State 管理自体、Store が全てを内包した結果  
ドメイン・ユースケース・Viewなどが混在しやすく、  
将来的にはアンダーにもなりうる中規模向けの施策

かも



常にバランスを考える

A decorative graphic on the left side of the page, consisting of several overlapping, semi-transparent pink shapes that resemble stylized leaves or petals. The shapes are layered, with some appearing in front of others, creating a sense of depth. The colors range from a light, pale pink to a slightly darker, more saturated pink.

**オーバーになりすぎないつまみ食い**

A decorative graphic on the left side of the page, consisting of several overlapping, semi-transparent pink shapes that resemble stylized leaves or petals. The shapes are layered, with some appearing in front of others, creating a sense of depth. The colors range from a light pink to a slightly darker shade.

**NPM のエコシステムを使いたいだけ**

# PARCEL

Blazing fast, zero configuration web application bundler



Get Started

GitHub

<https://parceljs.org/>

設定不要で使えるバンドルツール Parcel

# A modest JavaScript framework for the HTML you already have.

Sprinkle your HTML with controller, target, and action attributes:

```
<!--HTML from anywhere-->
<div data-controller="hello">
  <input data-target="hello.name" type="text">

  <button data-action="click->hello#greet">
    Greet
  </button>

  <span data-target="hello.output">
  </span>
</div>
```

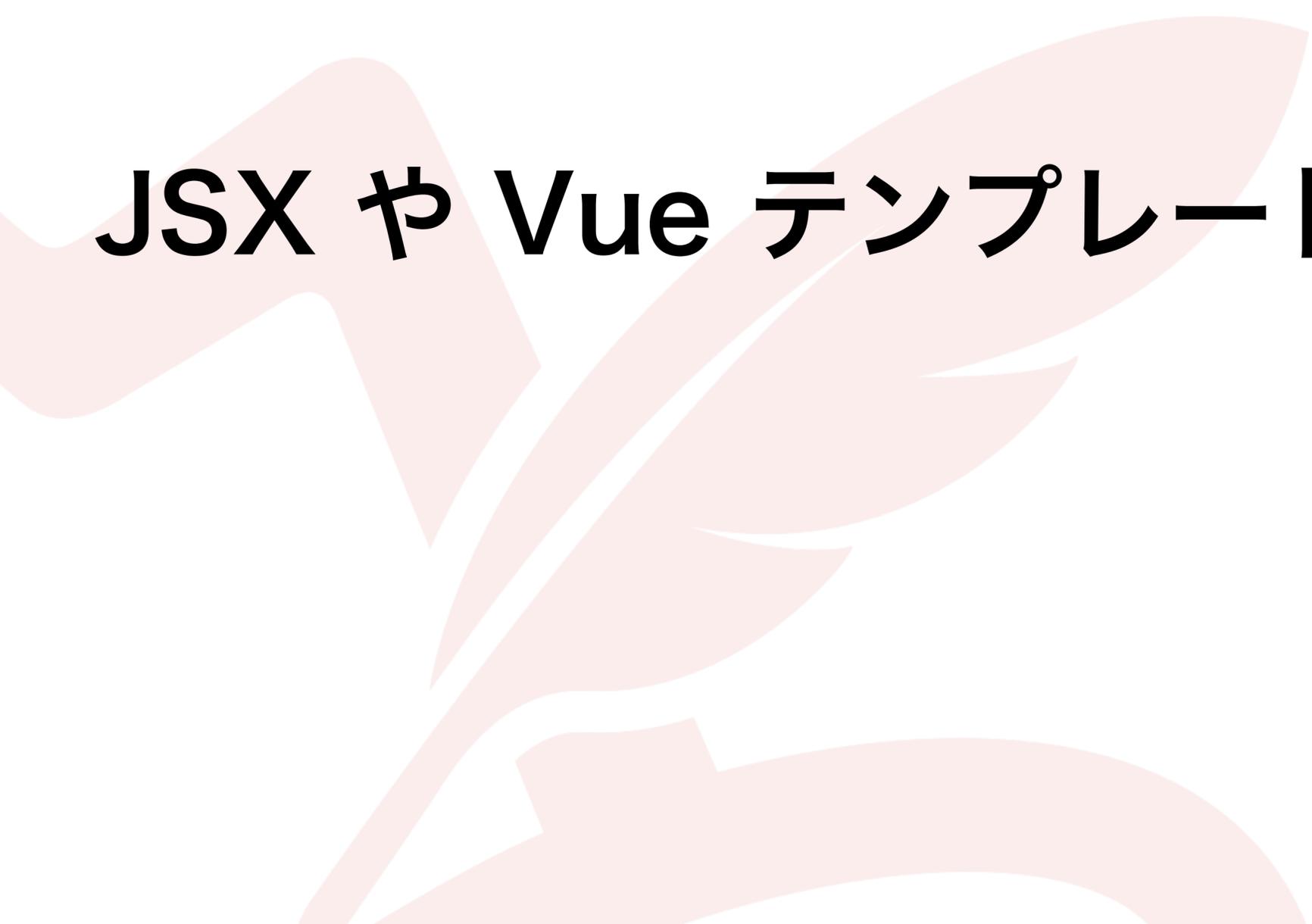
Write a compatible controller and watch Stimulus bring it to life:

```
// hello_controller.js
import { Controller } from "stimulus"

export default class extends Controller {
  static targets = [ "name", "output" ]

  greet() {
    this.outputTarget.textContent =
      `Hello, ${this.nameTarget.value}!`
  }
}
```

<https://stimulusjs.org/>



**JSX や Vue テンプレートで書きたいだけの場合**

## DATA SOURCES

### CMSs

Contentful, Drupal,  
WordPress, etc.

### Markdown

Documentation, Posts,  
etc.

### Data

APIs, Databases, YAML,  
JSON, CSV, etc.

## BUILD



powered by  
GraphQL

HTML · CSS · React

## DEPLOY

### Web Hosting

[AWS Amplify](#), [Netlify](#), [GitHub Pages](#), [Surge.sh](#), [Aerobatic](#), [ZEIT Now](#) & many more

<https://www.gatsbyjs.org/>

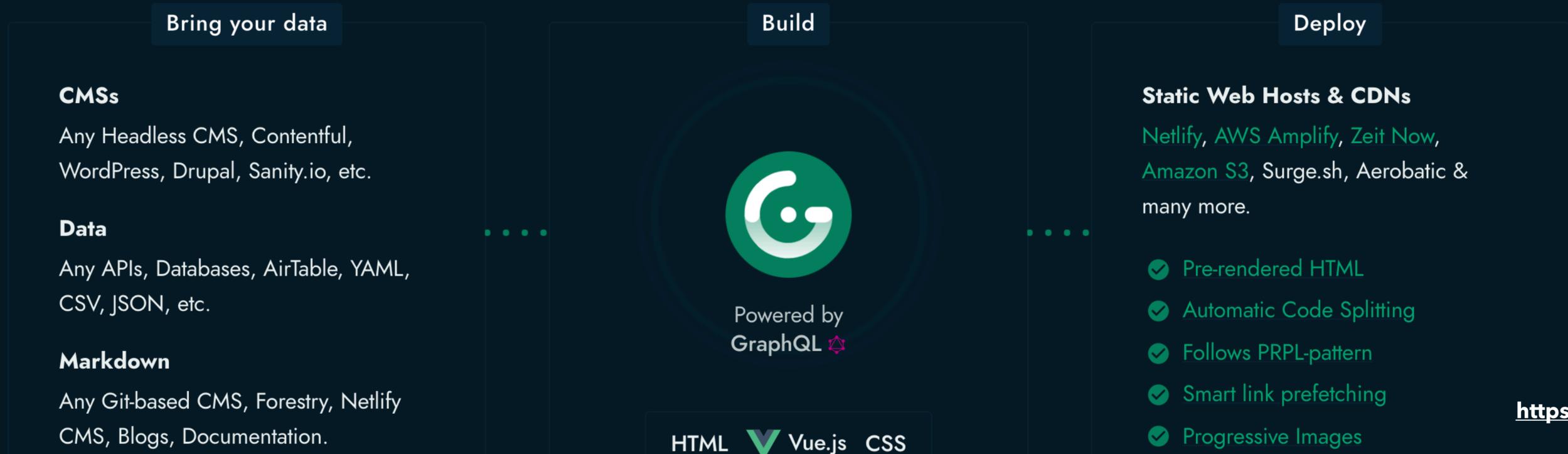
静的サイトの Developer Experience に使われる React 製の静的サイトジェネレーター Gatsby に

# A Vue.js framework for Markdown files

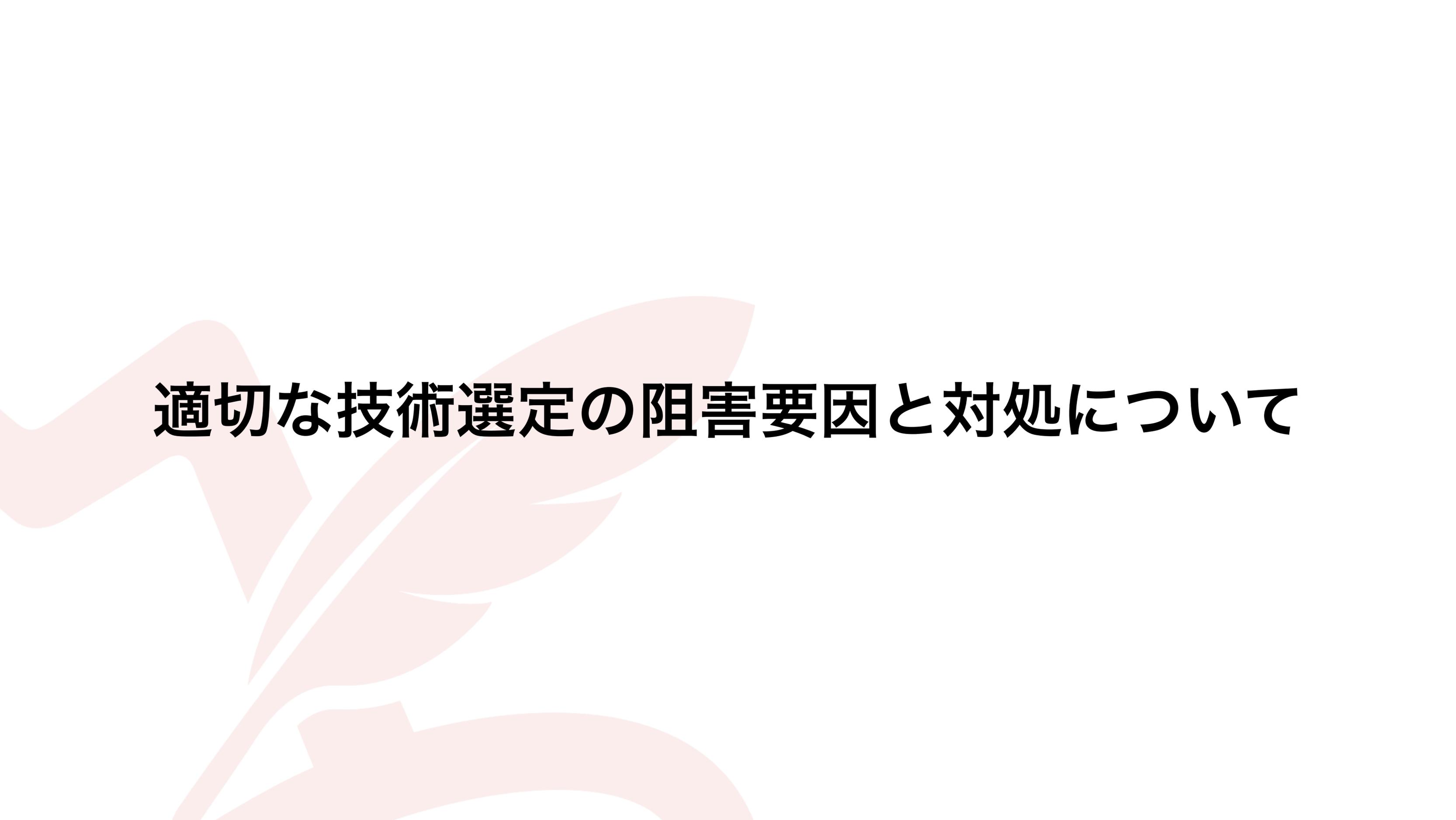
Gridsome makes it easy for developers to build modern  
websites, apps & PWAs that are **fast by default** 🚀

Get started

Open-source MIT Licensed. [GitHub \(v0.7.9\)](#)



その Vue.js 版とも言える静的サイトジェネレーター Gridsome も存在。



# 適切な技術選定の阻害要因と対処について

# 間違った技術選定の原因

- 間違った技術選定とは
  - 要件に対して過度なエンジニアリングが行われていること
  - 要件に対して将来性の無いエンジニアリングが行われていること
  - 速度と品質ではなく、**「要件と想定する最大のアーキテクチャ規模」**が重要
- どこから間違いが起こるのか
  - 意思疎通の PACKET ロスやそもそもの認識の誤り
  - **過ちは必ず現場から発生すること**を認識しておく

“信頼はしてるさ。けど、信用はしてない。確実なプレーを心がけても失敗はある。だからこそベンチがデータを掌握し、勝てる采配を振るわなきゃならねえんだ。”

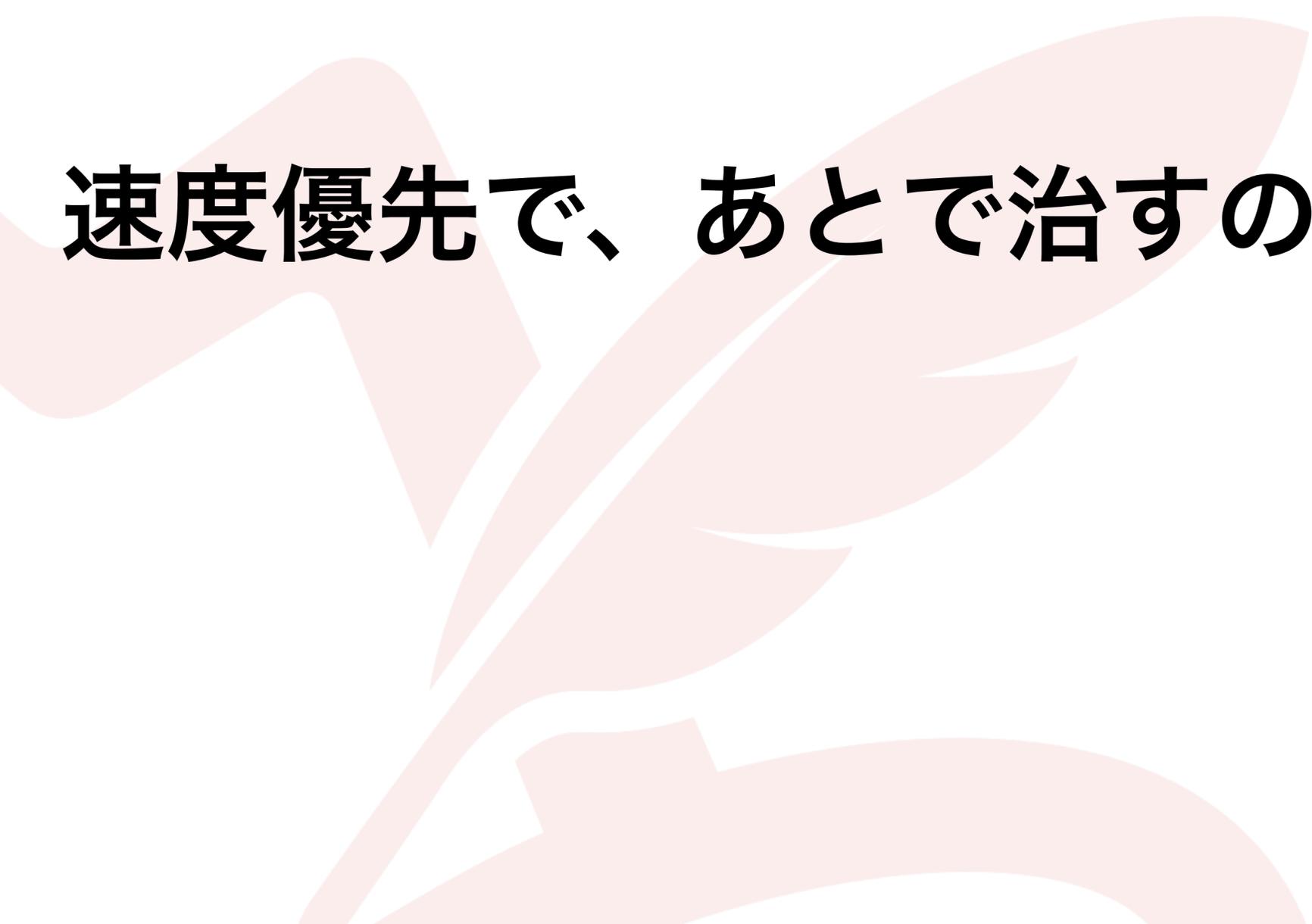
—『ラストイニング』 鳩ヶ谷

# 悪意のある間違った技術選定

- 技術選定を行う際、気をつけなければいけないことに「悪意」がある
  - 本来は悪意は存在しないことが望ましい
  - 一方で、その思い込みによって生まれる悪意がある

# 悪意のある間違った技術選定

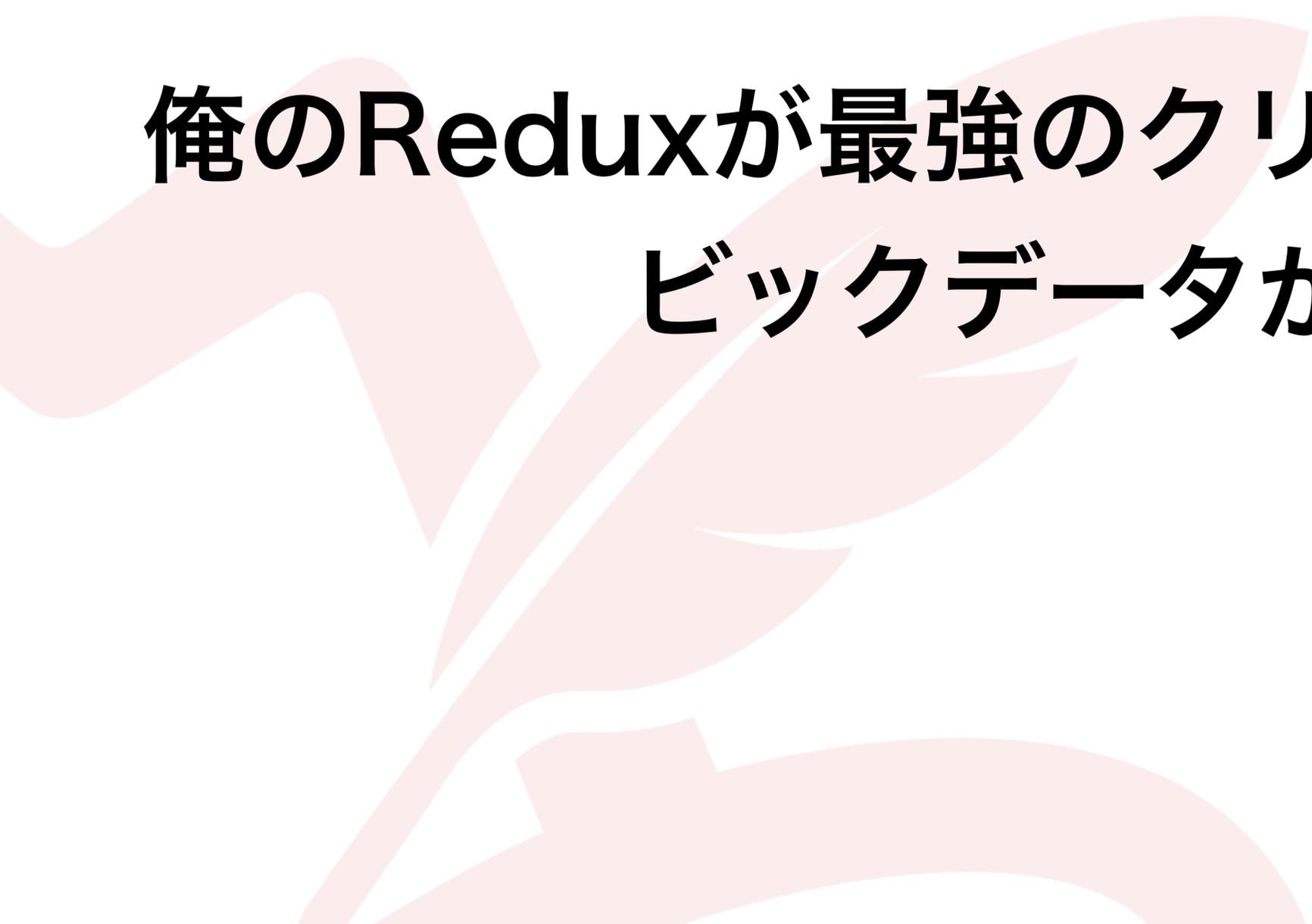
- ・ 属人化した「後から手を入れる」および初期からの「最高の設計」に注意
  - ・ 重厚な負債によって既得権益を狙っていないか
  - ・ ただ一人の思想に依存したアーキテクチャになっていないか
  - ・ プロダクトのためではなく、エゴを優先する「詐欺師」ではないか
    - ・ 「後から返済するんでマージしてください」は本当か
    - ・ 「このアーキテクチャでないと死ぬ」は本当か

A decorative background element consisting of several overlapping, semi-transparent pink shapes that resemble stylized leaves or petals, located on the left side of the slide.

**速度優先で、あとで治すのでマージしてください！**

「本当に返済に責任を持てる？」

「焦げ付かない？」



**俺のReduxが最強のクリーンアーキテクチャで  
ビッグデータがIoTでDDDD**

「なにかの受け売りになっていない？」

「技術的妥当性がある？」

# 悪意のある間違った技術選定

- 故意か過失かを問わず、人は必ず間違いを犯す
  - 人の歴史が、プログラミングの歴史が語ってきた事実
  - 判断できないときは**セカンドオピニオン**に頼るという手段も
    - 事業ではなく**技術専門の技術顧問**はそういうときのためにある
  - 技術選定に関わるときは、**信頼をしつつも言葉の意味を考えることが重要**

A decorative graphic on the left side of the slide, consisting of several overlapping, rounded, pink shapes that resemble stylized leaves or petals, extending from the bottom left towards the center.

**結局どうするのがいいの？**

# 結局どうするのがいいの？

## 1. 現状の結論としては「SPAは使えるなら使え」で間違いない……はず

- ・ 一方でこれからはわからないので、現状での最適解と心得ておく

## 2. SPAの技術は根幹的なものは使っても良いが、補助的なものは適度に

- ・ 少なくとも Flux アーキテクチャは必須ではない

## 3. 将来の負債返済を言い訳に将来性を欠くものはもちろん、その逆も注意する

- ・ アンダーが目の敵にされがちだが、オーバーも十分な危険信号
- ・ 我々にできるのは、要件と規模を常に考え続けること

A large, abstract graphic in a light pink color, resembling a stylized leaf or a flowing ribbon, positioned on the left side of the page. It has several curved, overlapping shapes that create a sense of movement and depth.

まとめ

# まとめ

- SPAの選定は、**現状の開発におけるもっと汎用的で有力な方法論**である
  - デファクトになっているものにはそれだけの理由が存在する
  - 一方で、現状がそうであるだけと認識しなくてはならない
- SPAは「なんとなく」で選定されがちだが、利用価値は2つの軸がある
  - 事業要件、ソフトウェア要件を満たすための **UX 要件**
  - 開発者の生産性向上による間接的な事業の成長のための **DX 要件**

# まとめ

- いつがUX要件であり、いつがDX要件であるか(例)
  - パフォーマンスやインタラクション設計、PWA対応などは UX 要件
  - 技術的妥当性やメンテナビリティなどは DX 要件
- SPA 技術をフル活用すべきか、そうでないのか
  - UX 要件の場合は事業に関わるため、事業的に必要かで判断
  - DX 要件は不足すると将来性を損ない、過度であるとエゴになるため慎重な判断を必要とする。良い塩梅の技術でお茶を濁すことも大切。

# まとめ

- 技術選定を行う立場の人間はどうするべきか
  - 要件に対する **アンダー / オーバーエンジニアリング**は何か
  - 何が必要化を明確に言語化し、常に意識する必要がある
- 成果物の将来を潰さないために
  - それらしい言葉でエゴを満たすエンジニア、不出来を誤魔化すエンジニア
  - **バランス感覚を破壊する詐欺師**に常に目を光らせ、選定者が責任を持つ

A stylized, dark gray leaf graphic is positioned on the left side of the image, extending from the bottom left towards the center. The leaf has a prominent vein structure and a pointed tip.

*“With great power comes great responsibility.”*

–Ben Parker