

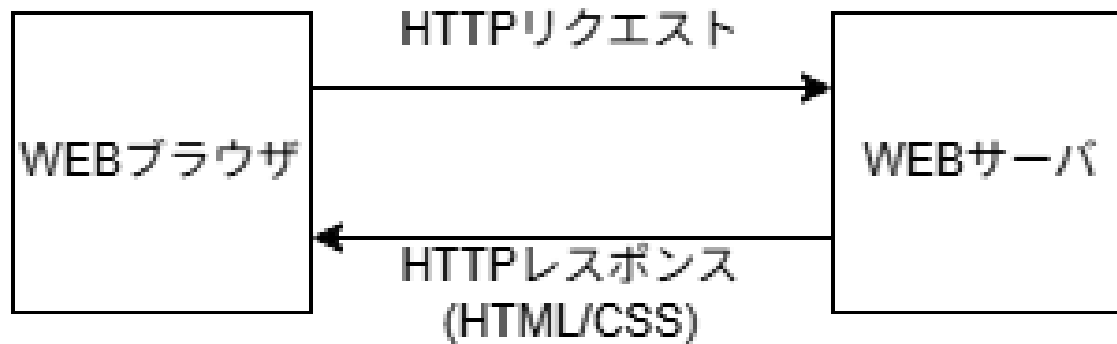
WEB システムの仕組み

～基本的な仕組みをざっくり理解しよう～

今日のお話

- **所要時間:** 約 15 分
- **内容:**
 - 静的 WEB サイト・動的 WEB サイトの基本構成
 - サーバーレス構成のメリットと注意点
 - SPA（シングルページアプリケーション）の仕組み
- **目標:** 基本的な仕組みを理解することで、応用につなげやすくする

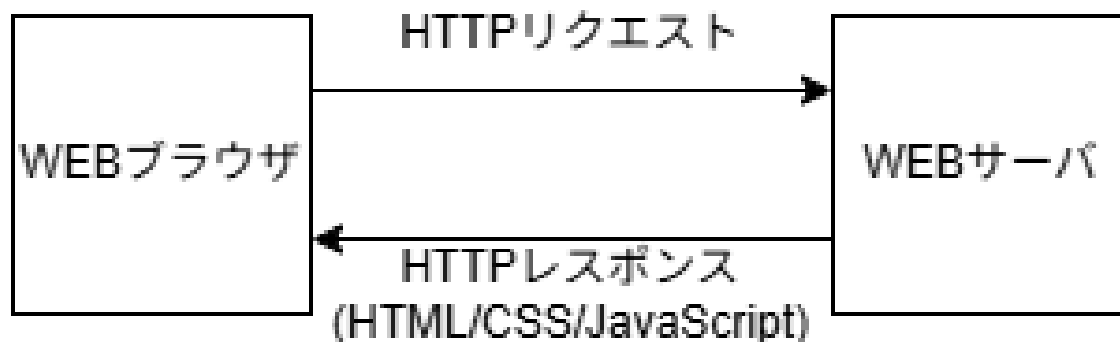
静的 WEB サイト



例：阿部寛のホームページ

- 事前に作成された HTML・CSS ファイルをそのまま返す

静的 WEB サイト（動作付き）



例：(<https://Shingotti.github.io/MyBlog/>)

- 動きはあるが、WEB サーバ側では処理を行わない
- JavaScript はブラウザ側で実行される

動的 WEB サイト



- サーバ側でプログラムが実行される
- ユーザーの入力やデータベースの状態に応じて HTML を生成

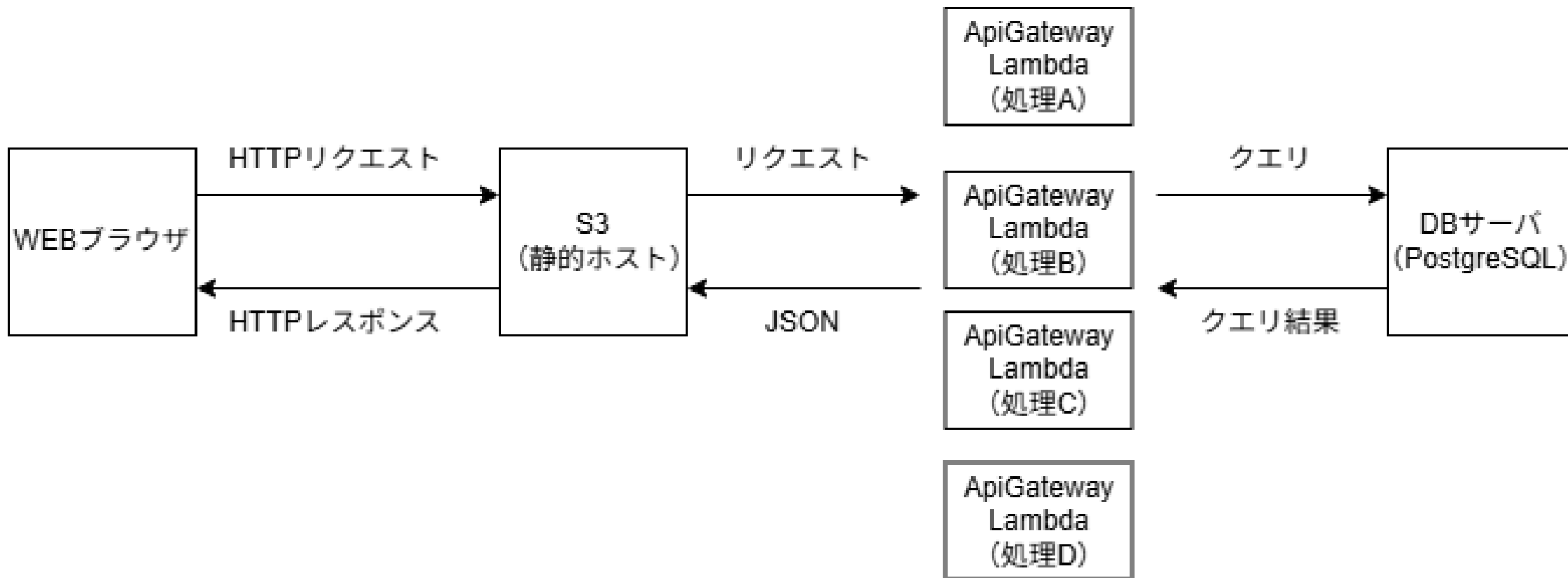
WEB サーバの役割

- HTTP 通信の処理
 - WEB ブラウザと、HTTP リクエスト・レスポンスの送受信
- アプリケーションサーバへの橋渡し
 - 必要に応じて、リクエストをアプリケーションサーバに中継する

アプリケーションサーバの役割

- 業務処理の実行
 - WEB サービスが担っている機能を実行する
- データベース接続・操作
- HTML の動的生成
 - プログラム実行による動的な HTML 作成

サーバレスな構成 (AWS)



- アプリケーションサーバの代わりに、Lambda 関数で作った API を使う
- クラウドが発展したことにより、サーバーを意識しなくてよくなった

サーバレス構成の開発メリットと注意点

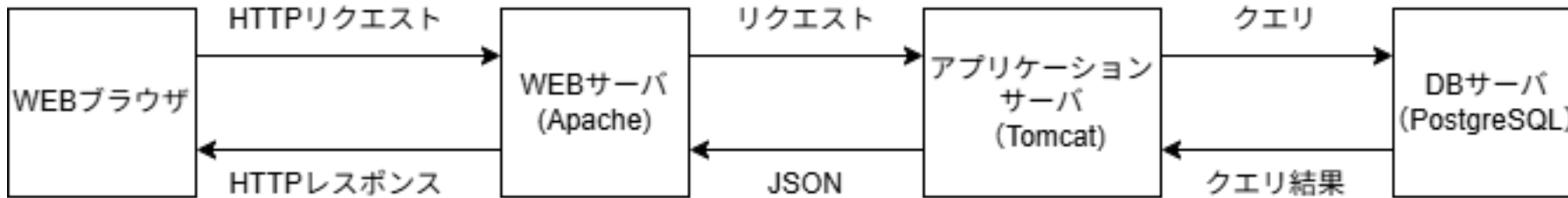
メリット

- **疎結合設計**：機能を API 単位で分離でき、既存コードに影響せず機能追加が可能
- **独立したデプロイ**：API 単位でデプロイできるため、障害時の原因特定が容易

注意点

- **移行コスト**：MVC モデルからの移行は大規模な修正が必要
- **設計の重要性**：設計次第では密結合になるリスクがある

SPA (Single Page Application)



■ SPA の仕組み

- **初回のみ HTML 読み込み**：最初にページ全体を取得
- **JSON データ通信**：以降は JSON でデータのやり取り HTTP まるごととは送らない
- **クライアントサイドレンダリング**：JavaScript でページを動的に更新

従来の WEB アプリケーションとの違い

従来型	SPA
毎回 HTML を取得	初回のみ HTML 取得
サーバサイドで HTML 生成	クライアントサイドで HTML 生成
ページ遷移時にリロード	ページ遷移時もリロードなし

参考書籍・資料

 「[改訂新版] プロになるための Web 技術入門」 ★★★★★

- おすすめポイント: ハンズオンが豊富。TODO アプリを作りながら WEB アプリの作り方が学べる

 「イラスト図解式 この一冊で全部わかる Web 技術の基本」 ★★★★★☆☆

- おすすめポイント: 図解が豊富。かなり初心者向け

ご清聴ありがとうございました！ 🎉