

# 情報システムの構造

---

藤島拓大

2022/05/16

# 自己紹介

藤島拓大



:tingtt



:tiTakung

趣味: 音ゲー(プロセカとか)

得意:

サーバー関連、DB、  
バックエンド開発、  
フロントエンド開発、UI設計

# 去年の研究

学校配布のPCのWindowsを消してサーバー化したった！



実は・・・

# このプレゼン2回目

1回目は...

わかりにくかったらしい



5月病を乗り越え

ほとんど書き換えた





# 目次

- アプリケーションって何？
- Webサイトってどういう仕組み？
- データはどこに保存するの？
- 僕たちは何を作るの？
- モバイルアプリと機械学習関連で補足

# アプリケーションとは

---

# アプリケーションとは

OS上で動く全てのソフトウェア

- ブラウザ



- Office



- Webサーバアプリケーション



- データベース(DB)



- コマンド



仕組みは同じ

# 課題で作っているのもアプリケーション

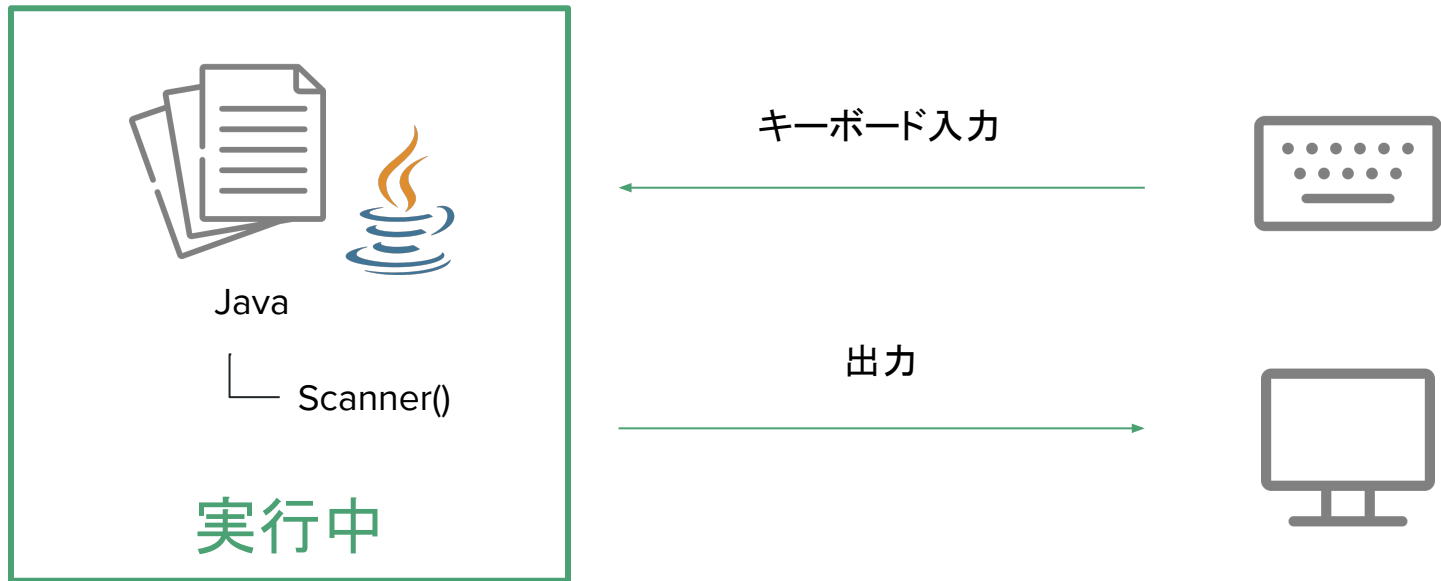


Java

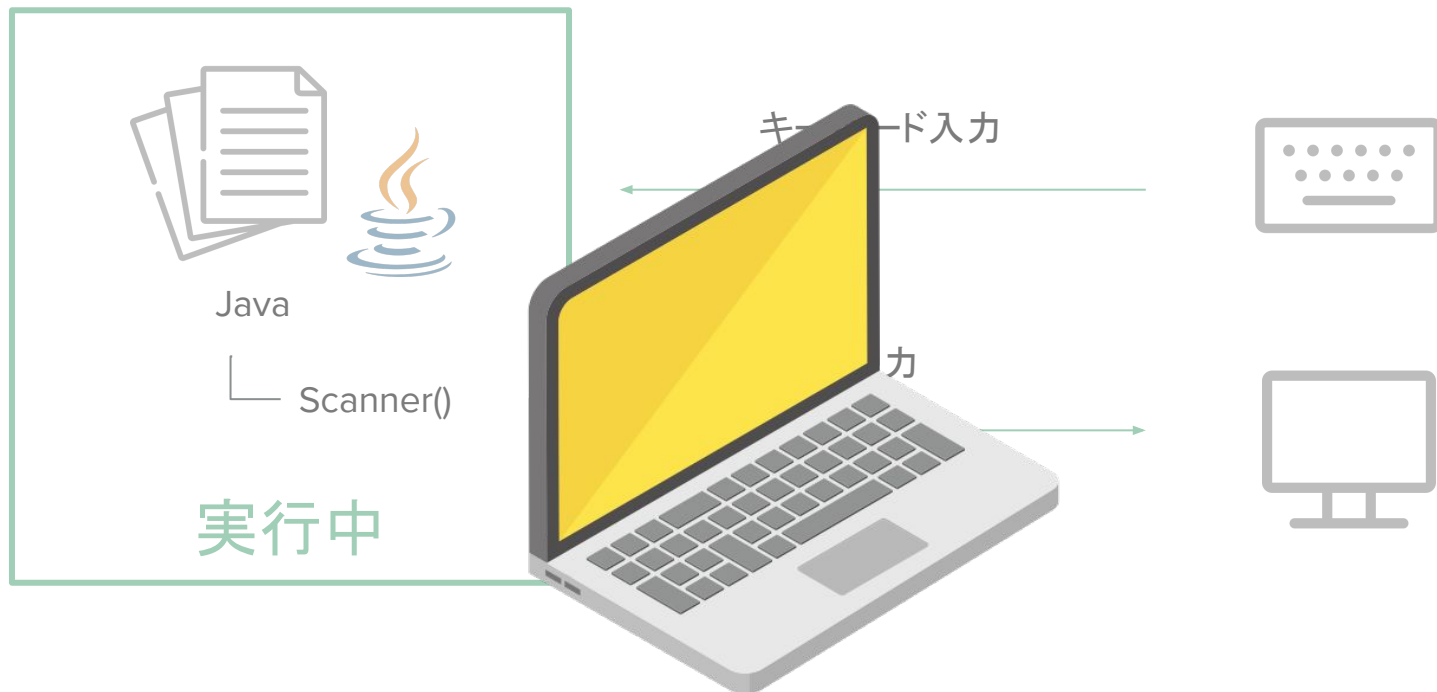
# Webサイト

---

# Javaの課題の例をイメージしてみると

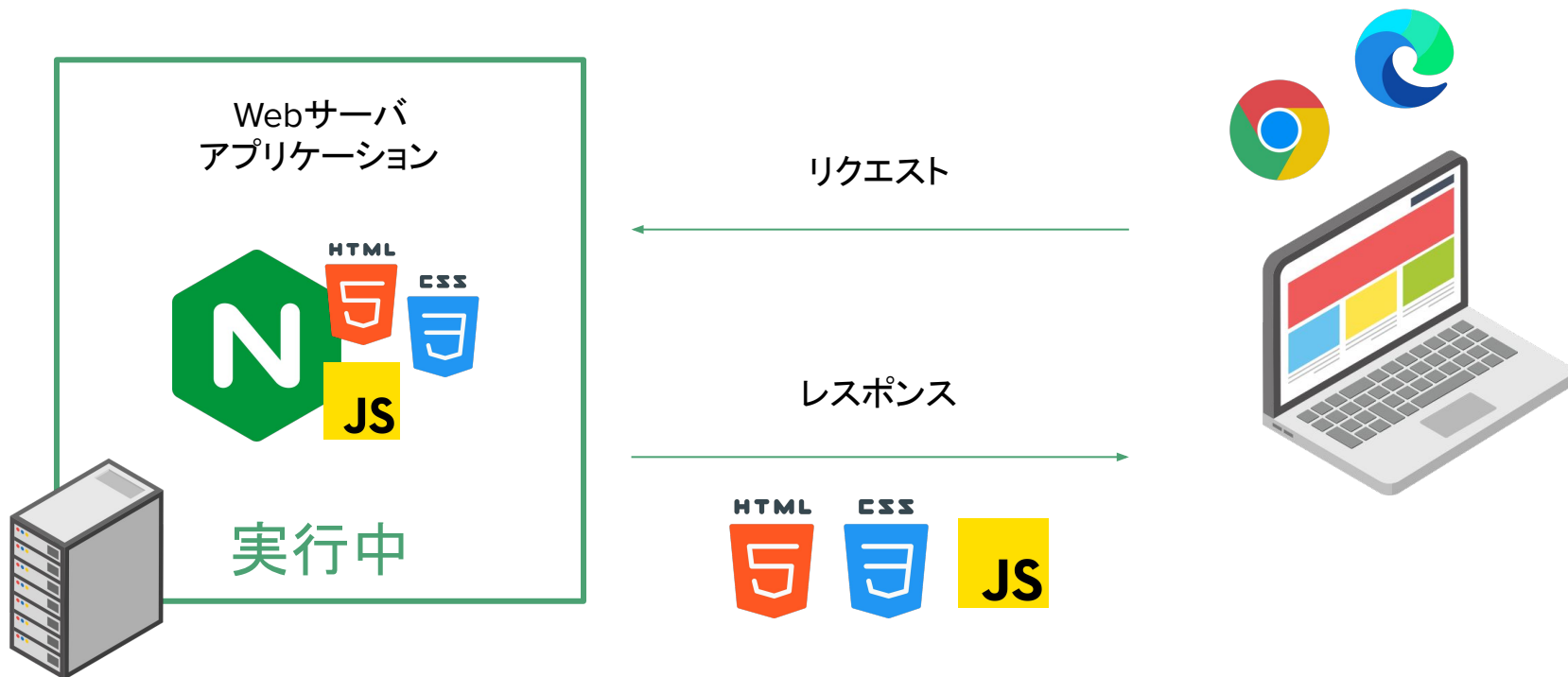


# 端末上で完結

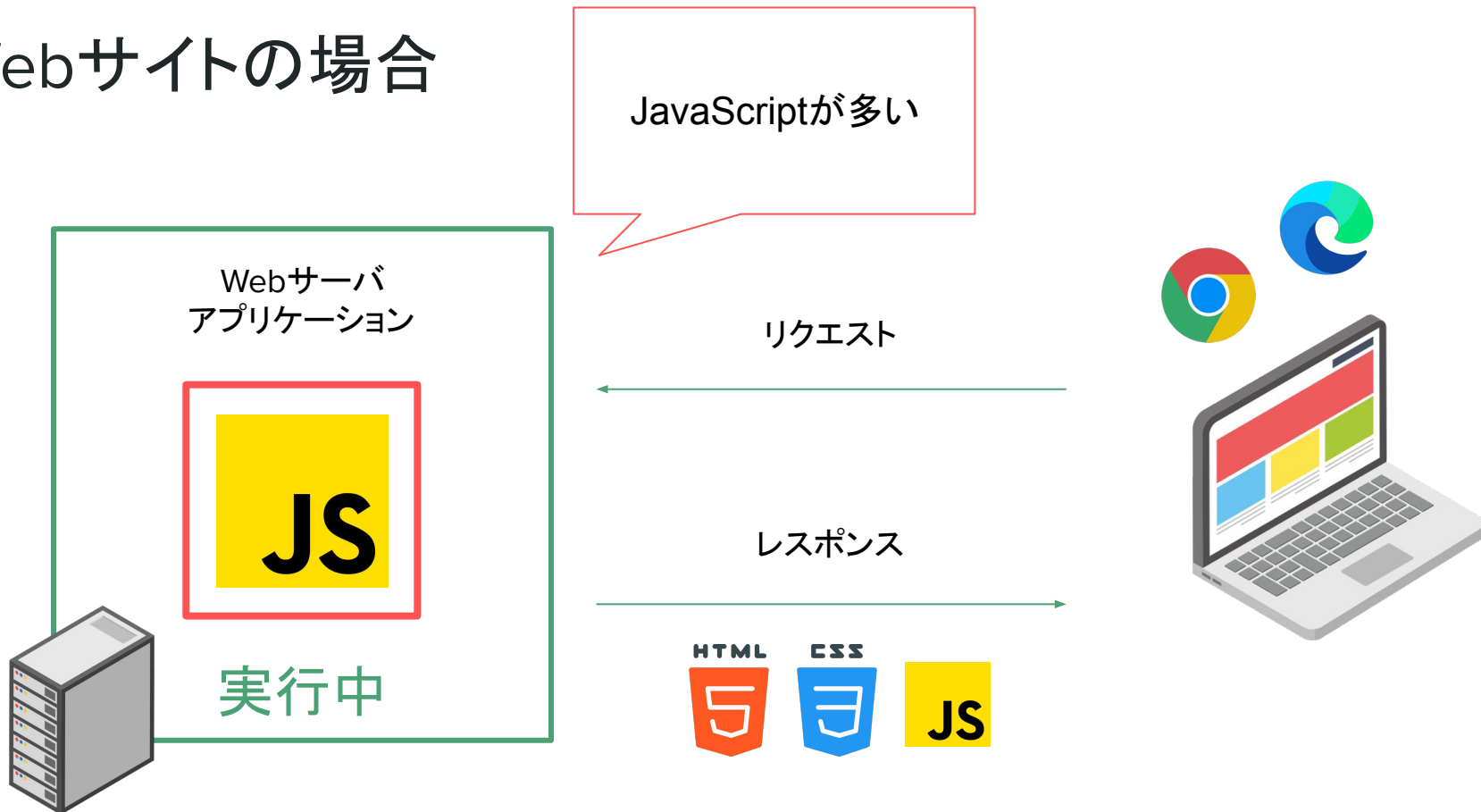




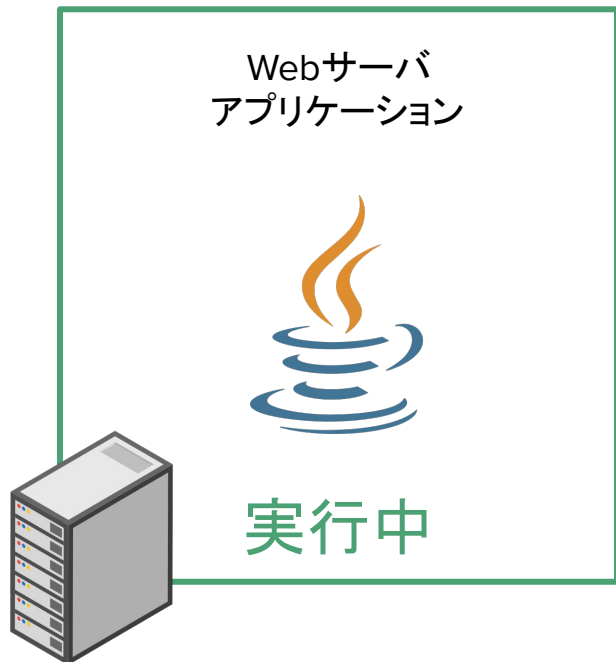
# Webサイトの場合



# Webサイトの場合



# Webサイトの場合



Javaでもできる  
(Apache Struts ?)

リクエスト

レスポンス

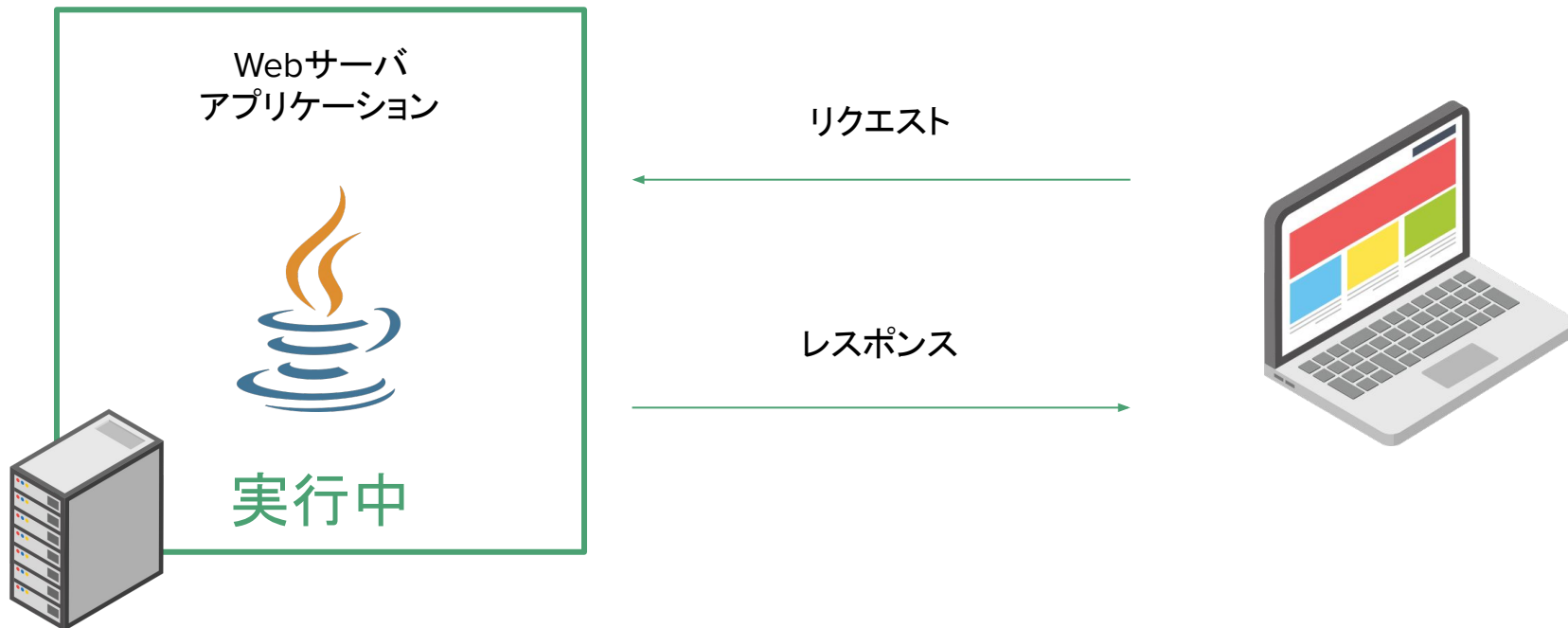


# データの保存

---

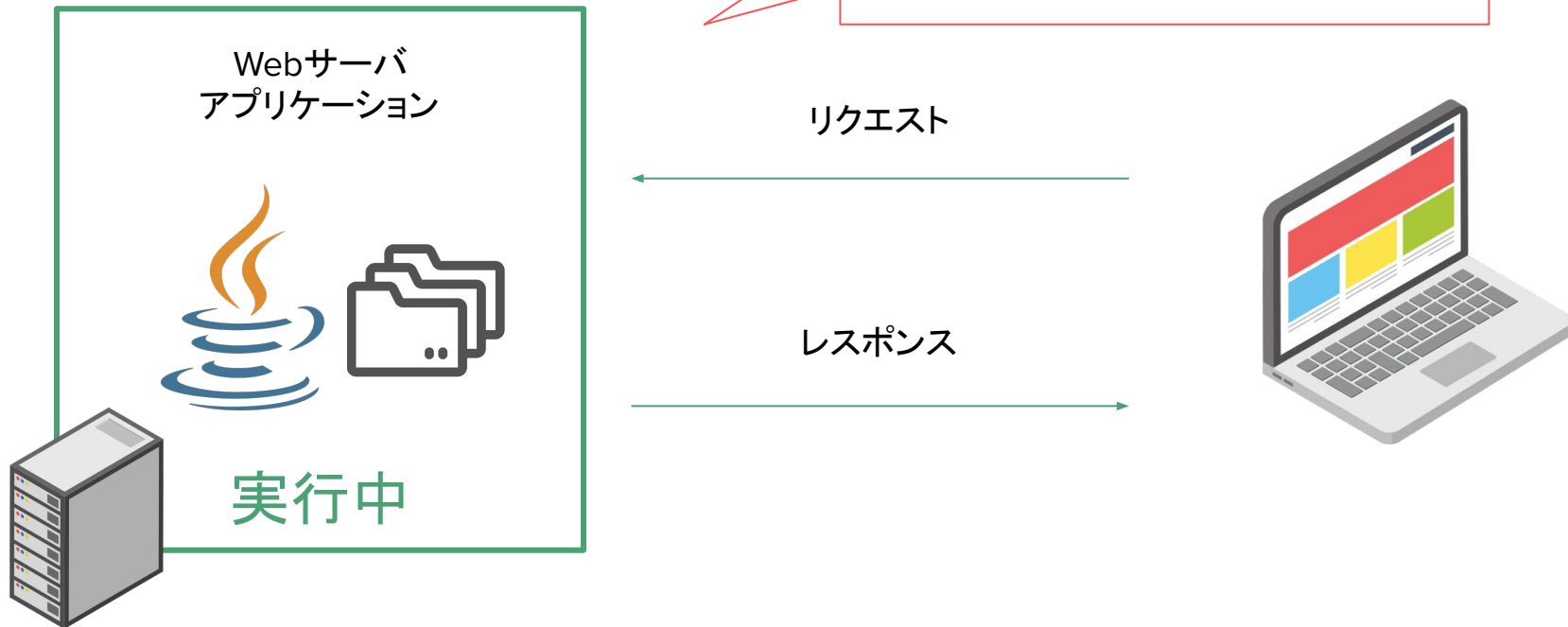
# ユーザーのデータを 扱う場合

# どこにデータ保存するの？



# どこにデータ保存するの？

Webサーバアプリケーションにデータを保存する機能をつける？



それは面倒・・・



# データベース(DB)

DBサーバ  
アプリケーション



リクエスト



レスポンス



Webサーバ  
アプリケーション



リクエスト

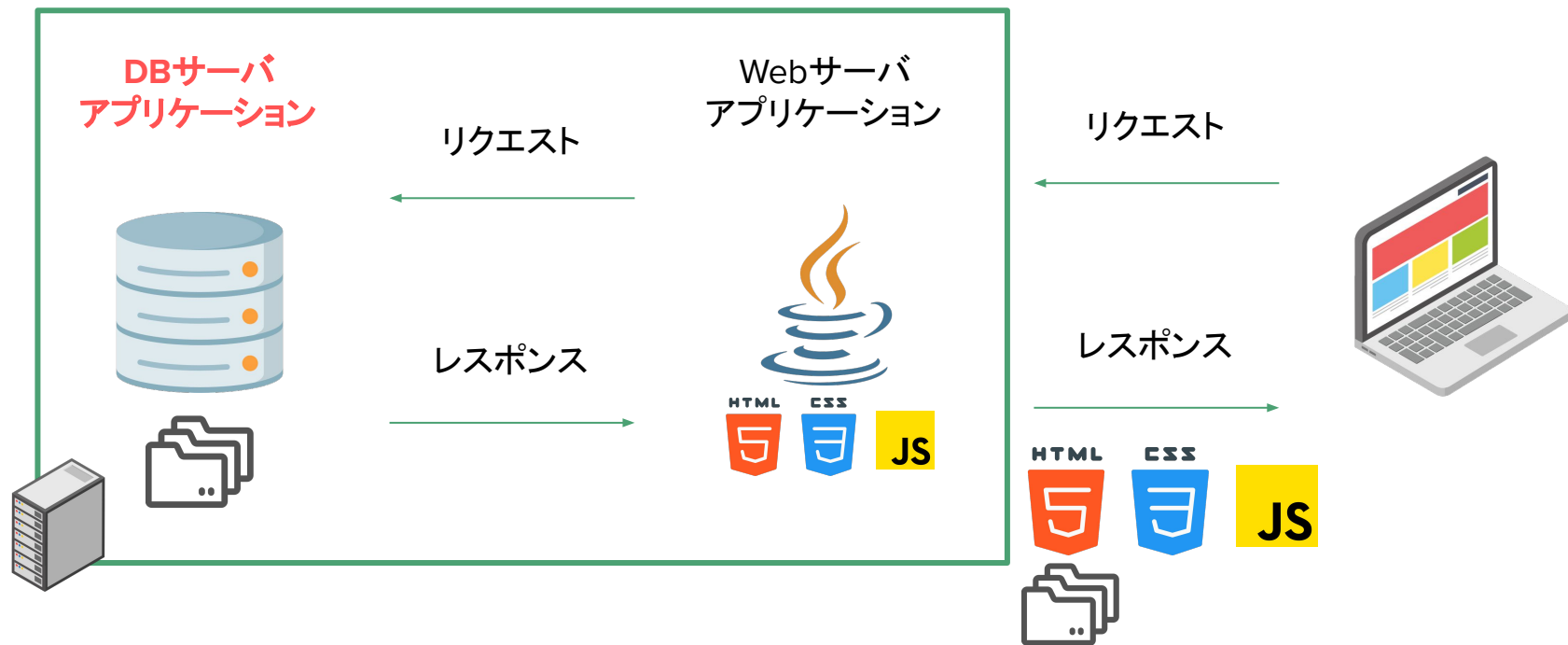


レスポンス



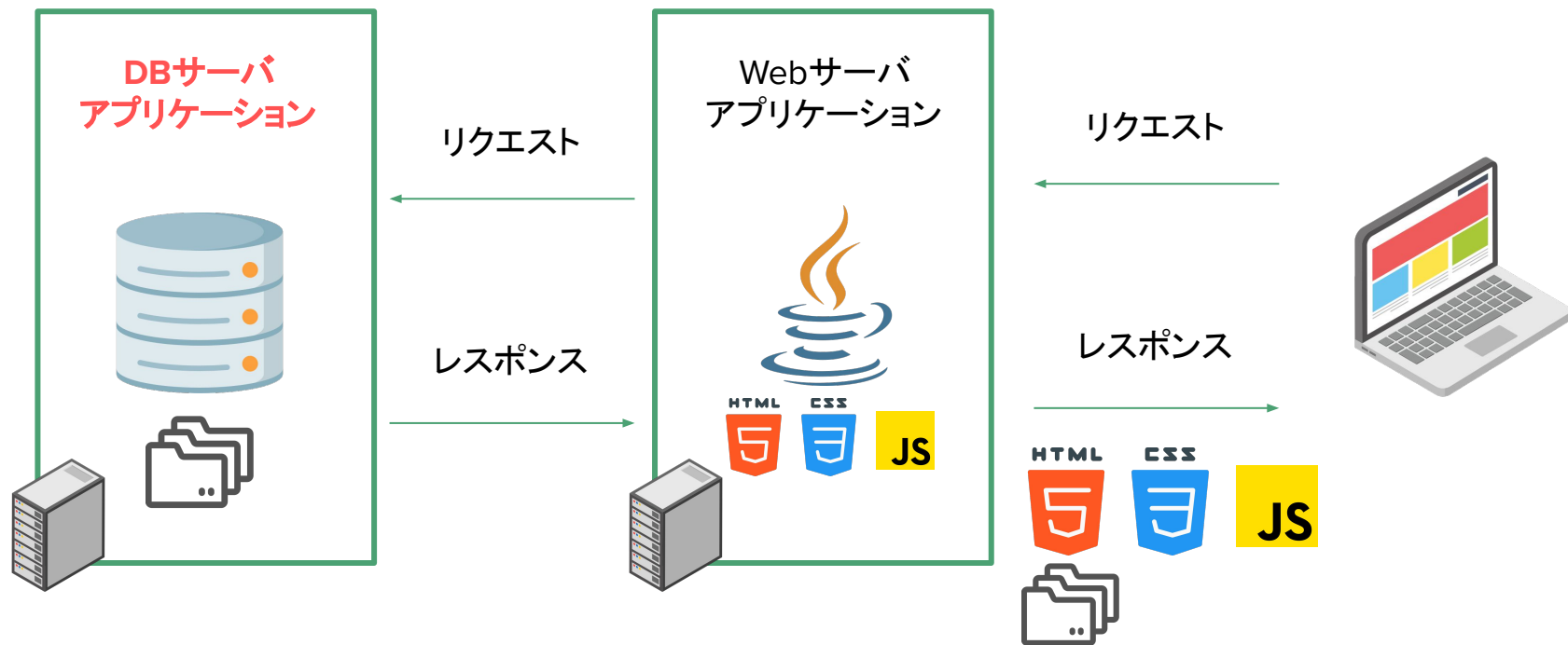
# データベース(DB)

同じサーバに乗せてもOK



# データベース(DB)

別のサーバに分けてもOK

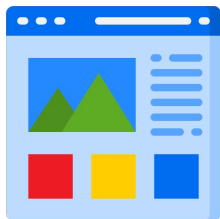


作るには

---



DBに何を保存するか



どんな表示をするか

を考える。

作りたいものを作ろう

# 補足

---

Webサーバ  
アプリケーション

DBサーバ  
アプリケーション



リクエスト

レスポンス



リクエスト

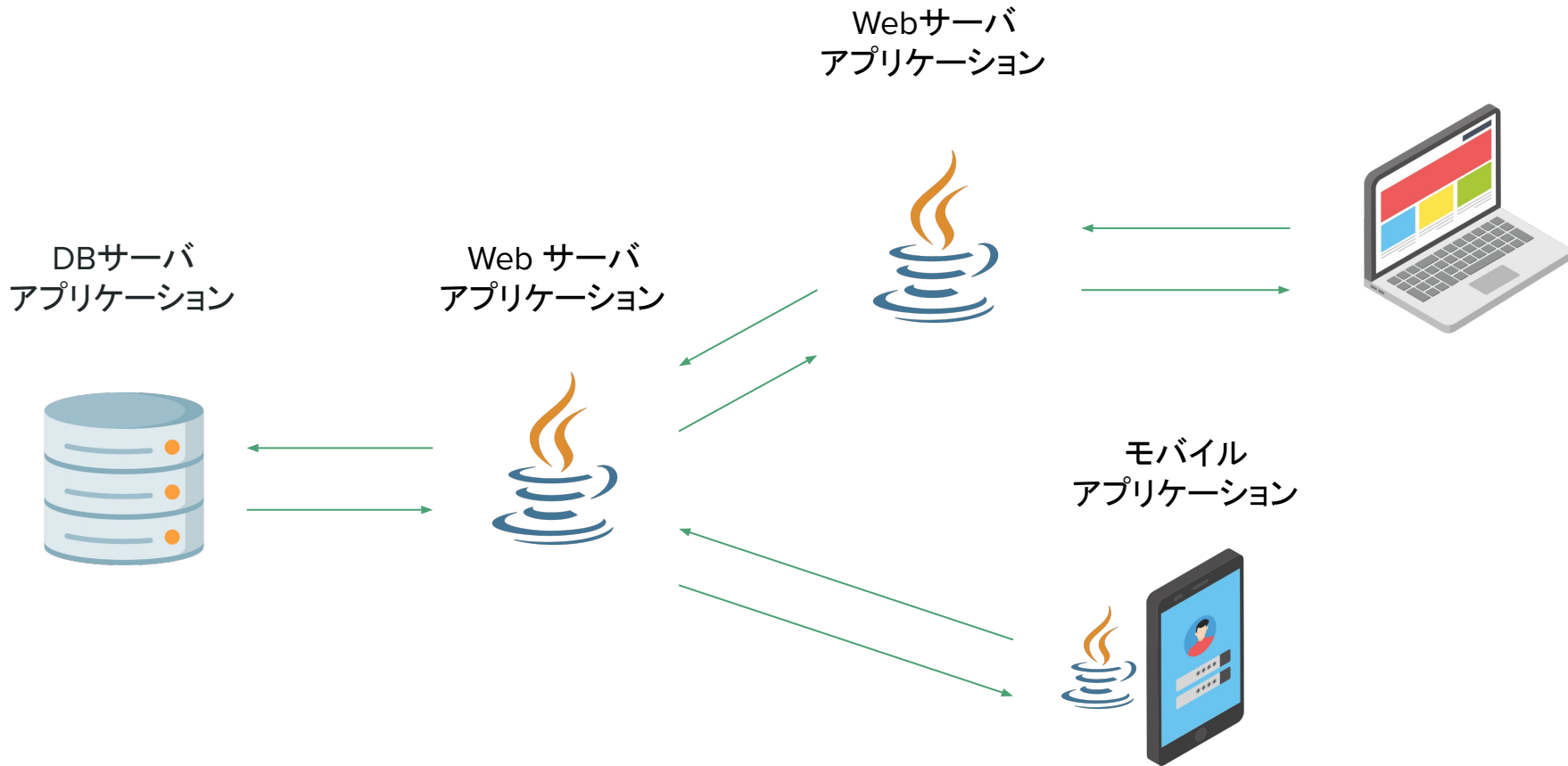
レスポンス

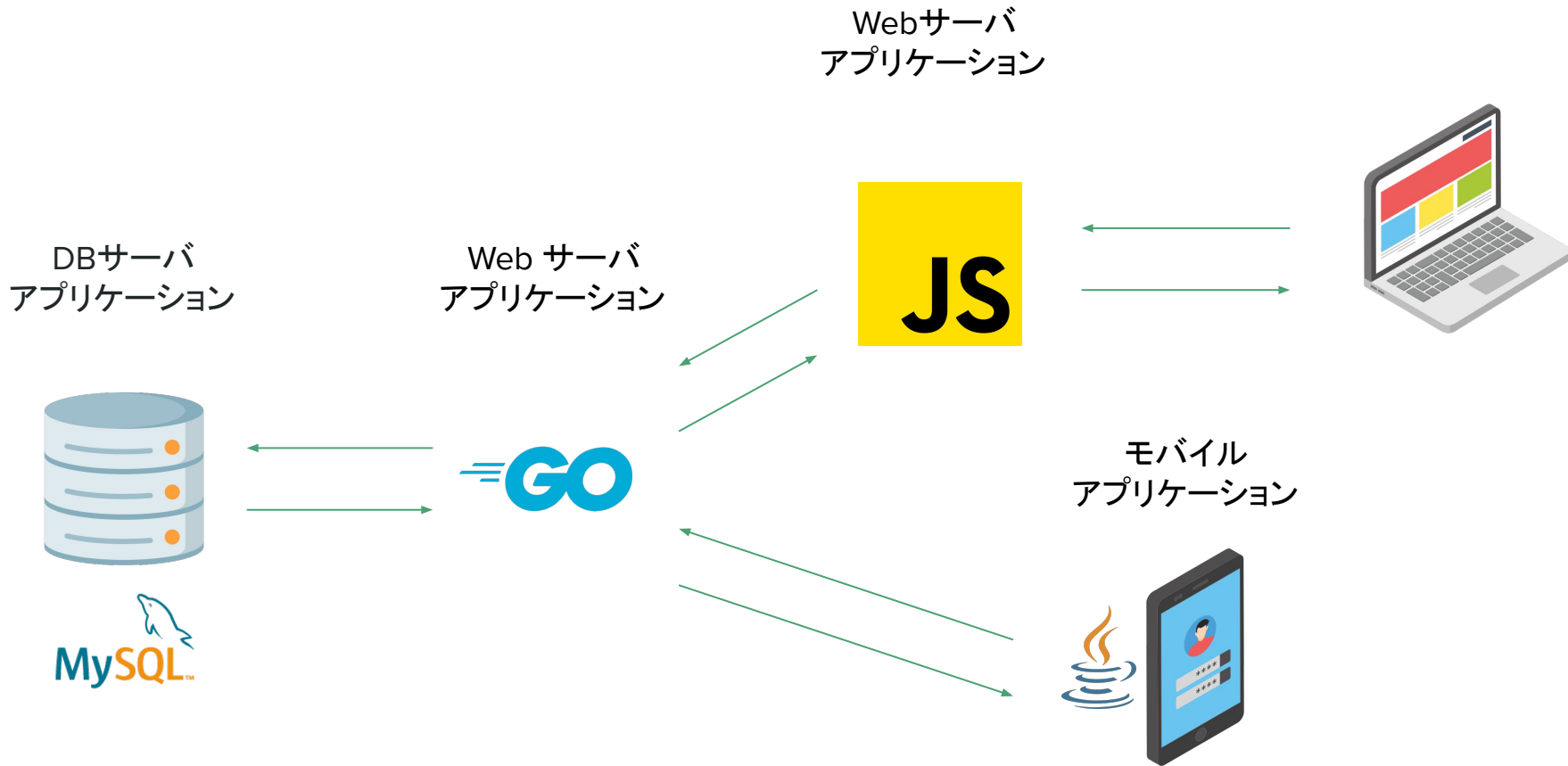
リクエスト

レスポンス









# まとめ

- OS以外は全てアプリケーション
- Webサイトもアプリケーション
- データはDBに保存
- データと見た目を作る！
- モバイルアプリと機械学習も同じ

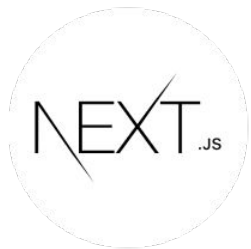
# 得意



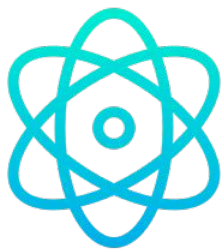
golang



TypeScript



Next.js



React.js



Figma



Git



Docker



Linux

いまは・・・

2台に増えた！

3台にしたい・・・



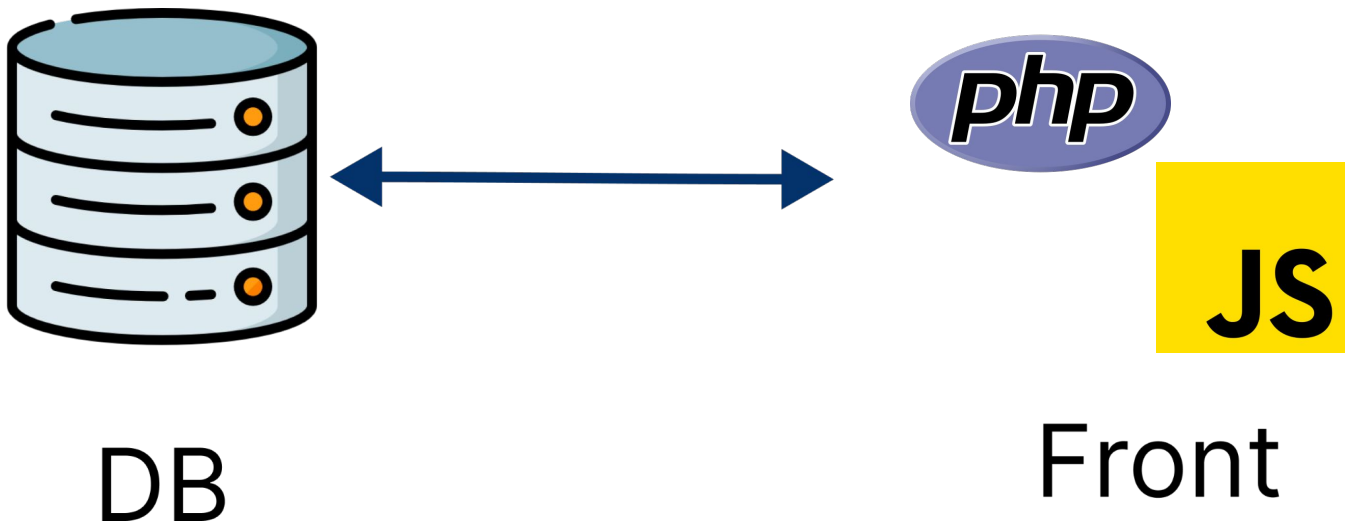
# 目次

- よくある構成
- 行き交っている情報ってどんなの？
- デモ
- 領域ごとの技術紹介

# 構成

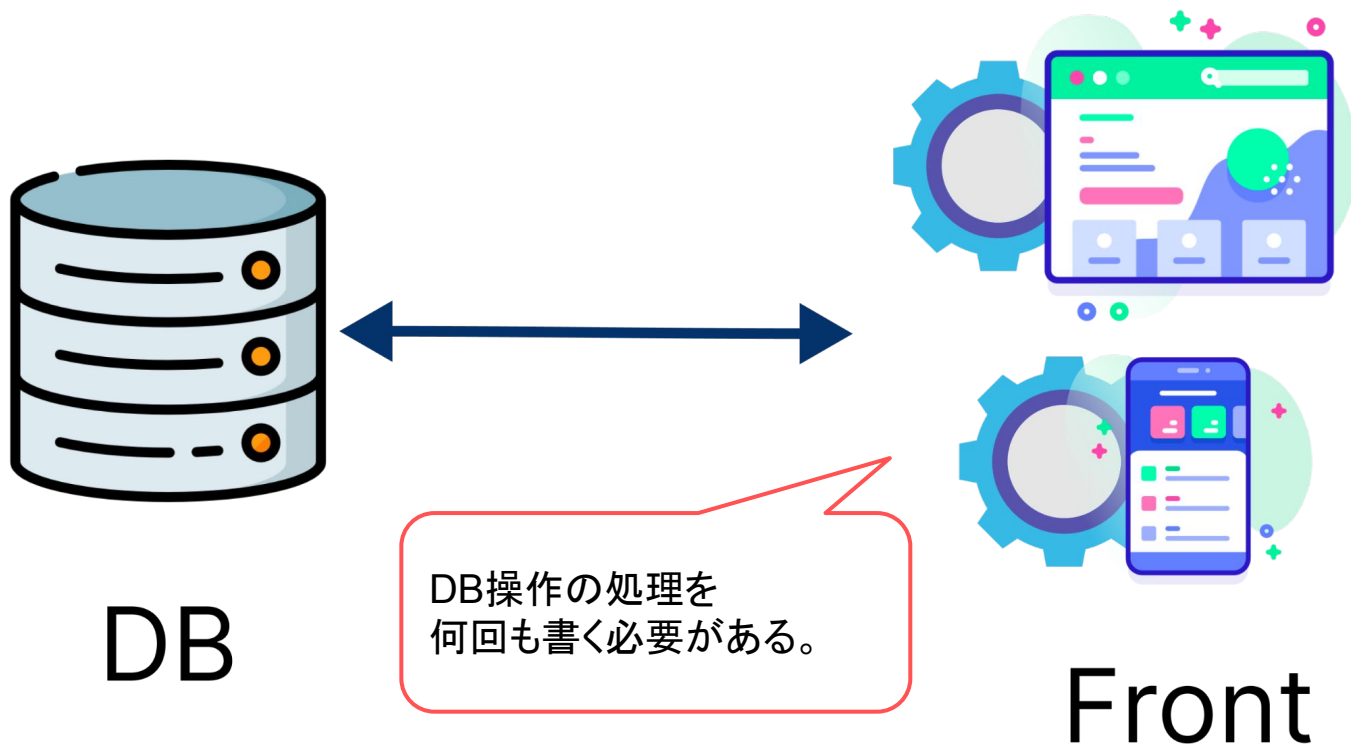
---

## 最低限の構成

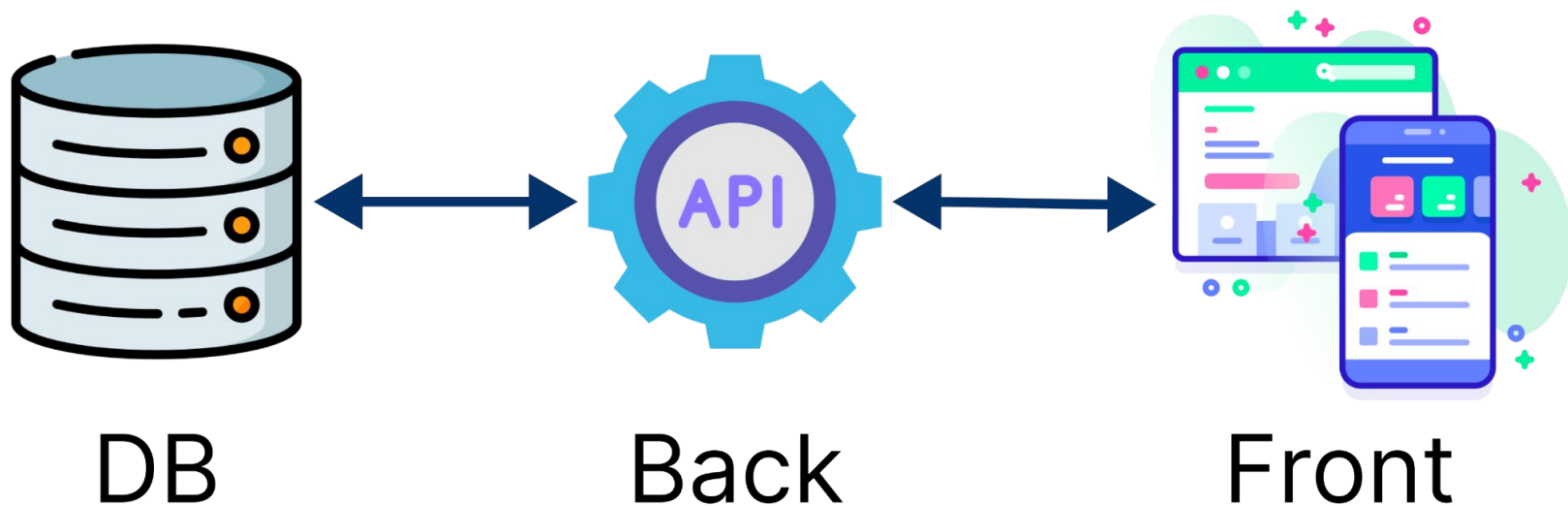




# 最低限の構成



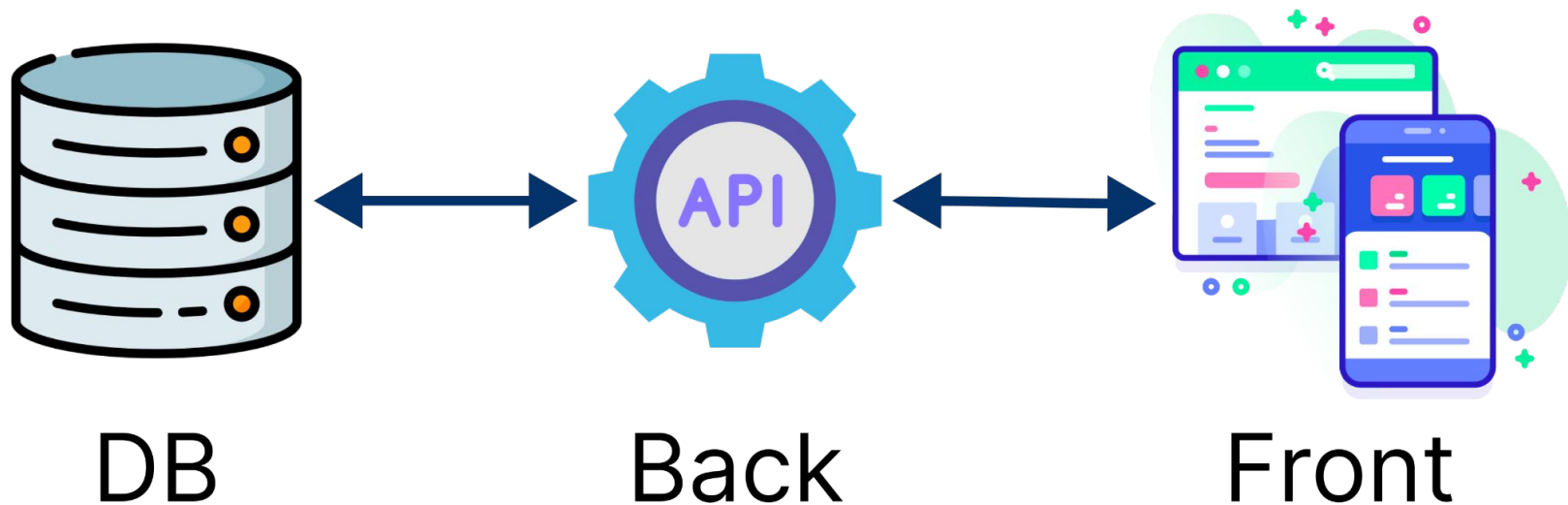
## よくある構成



# 行き交っている情報

---

# 行き交っている情報



DB



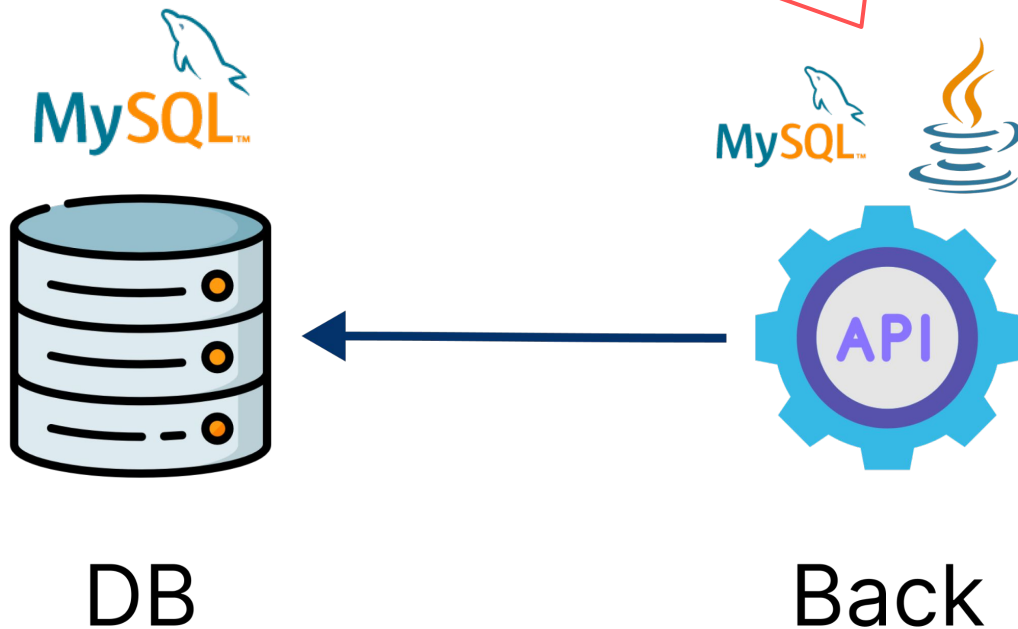
DB

# DB(手動)

id	user_id	name	date	time	execution_time	sprint_id	project_id	completed	repeat_model_id
1	3	ナビゲーションバーの改善	2022-03-21	11:20:00	NULL	2	1	0	NULL
2	3	ユーザー設定のシナリオ書き出す	2022-03-21	NULL	NULL	2	1	0	NULL
3	3	records OAS	2022-03-14	09:30:00	NULL	3	1	1	NULL
4	3	records OAS 続き	2022-03-14	11:20:00	NULL	3	1	1	NULL
5	3	records 実装	2022-03-15	09:30:00	NULL	3	1	1	NULL
6	3	records 実装 続き 1	2022-03-15	11:20:00	NULL	3	1	1	NULL
7	3	records 実装 続き 2	2022-03-16	09:30:00	NULL	3	1	1	NULL
8	3	records 実装 続き 3	2022-03-16	11:20:00	NULL	3	1	1	NULL
9	3	records 実装 続き 4	2022-03-16	09:30:00	NULL	3	1	1	NULL
10	3	projects 親子表示	2022-03-17	09:30:00	NULL	3	1	1	NULL
11	3	pomodoro テスト	2022-03-17	11:20:00	NULL	3	1	1	NULL
12	3	changelog OAS	2022-03-17	09:30:00	NULL	3	1	1	NULL
13	3	changelogs 実装	2022-03-18	09:30:00	NULL	3	1	1	NULL
14	3	changelogをrecordに埋め込み	2022-03-18	11:20:00	NULL	3	1	1	NULL
15	3	queryパラメータの最適化	2022-03-21	09:30:00	NULL	3	1	0	NULL
16	3	モックデータ作成	2022-03-21	09:30:00	NULL	5	1	1	NULL
17	3	モックデータ作成 続き 1	2022-03-22	NULL	NULL	5	1	0	NULL
18	3	モックデータ作成 続き 2	2022-03-23	NULL	NULL	5	1	0	NULL
19	3	定例ミーティング	2022-04-12	NULL	NULL	NULL	1	1	NULL
20	3	定例ミーティング	2022-04-19	NULL	NULL	NULL	1	1	NULL
21	3	定例ミーティング	2022-04-26	NULL	NULL	NULL	1	1	NULL
22	3	定例ミーティング	2022-05-06	NULL	NULL	NULL	1	1	NULL
23	3	定例ミーティング	2022-05-10	NULL	NULL	NULL	1	0	1
24	4	test	2022-05-06	NULL	NULL	NULL	NULL	0	NULL
25	4	test	2022-05-06	NULL	NULL	NULL	NULL	0	NULL
26	4	test	2022-05-06	NULL	NULL	NULL	NULL	0	NULL
27	4	test	2022-05-06	10:00:00	NULL	NULL	NULL	0	NULL
28	4	test	2022-05-08	10:00:00	NULL	NULL	NULL	0	NULL
29	4	test	2022-05-06	10:00:00	NULL	NULL	NULL	0	NULL
30	4	test	2022-05-06	NULL	NULL	NULL	NULL	0	NULL
31	4	test	NULL	NULL	NULL	NULL	NULL	0	NULL
32	4	test	NULL	NULL	NULL	NULL	NULL	0	NULL

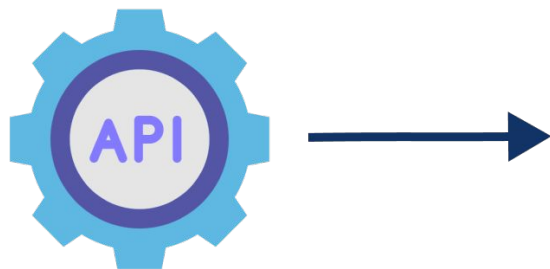
# DB ↔ API

DBクライアントをAPIが間接的に扱う



# API

JSON形式で出力

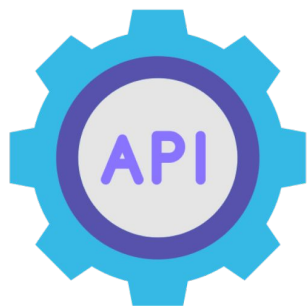


```
{  
  "id": 23,  
  "name": "定例ミーティング",  
  "date": "2022-05-10",  
  "project_id": 1,  
  "completed": false,  
  "repeat": {  
    "until": "2022-08-02",  
    "unit": "week",  
    "days": [  
      {  
        "day": 2,  
        "time": "09:15"  
      }  
    ]  
  }  
}
```

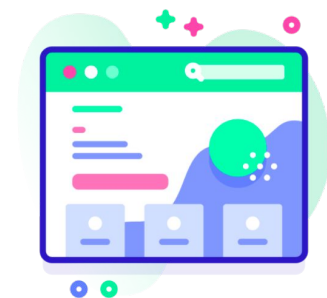
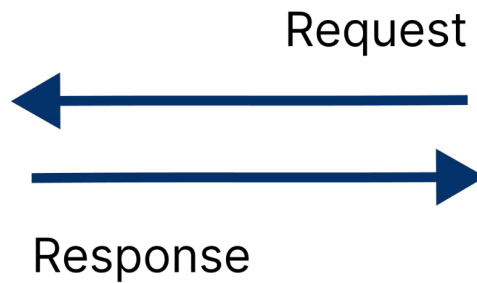


# API ↔ Front

JavaScriptでAPIにアクセスし、JSONを解析



API

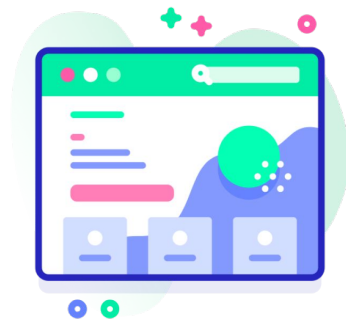
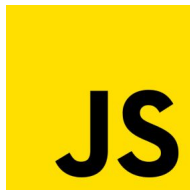


Front

# Front

解析したJSONのデータをもとにGUIを生成

```
{
  "id": 23,
  "name": "定例ミーティング",
  "date": "2022-05-10",
  "project_id": 1,
  "completed": false,
  "repeat": {
    "until": "2022-08-02",
    "unit": "week",
    "days": [
      {
        "day": 2,
        "time": "09:15"
      }
    ]
  }
}
```



デモ

---

# ざっと技術紹介

---

サーバー	DB	バックエンド	フロント Web		フロント モバイル	UIデザイン
 	  	  	  	  	  	  <div>開発 便利ツール</div> 

※ バックとフロントの言語は入れ替わったりもします。