

REST APIに関するE2Eテストの分類と、 それらの自動テストの適用範囲について

藏内 亮

自己紹介

- 藏内 亮
- 株式会社STYLY 2019年入社
- サーバーサイド責任者
- プラットフォーム部 EM



Qिता 表彰プログラム SILVER

10,000Contributionを獲得した記念のバッジです。



自動テストはなぜうまくいかないか？
乗り越えるためには何が必要か？

@kotaichisun
in STYLY, Inc.

Qिता

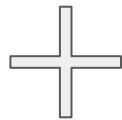
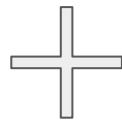
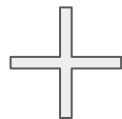
あなたはフロントエンドの何をテスト
したいのか。

@kotaichisun
in STYLY, Inc.

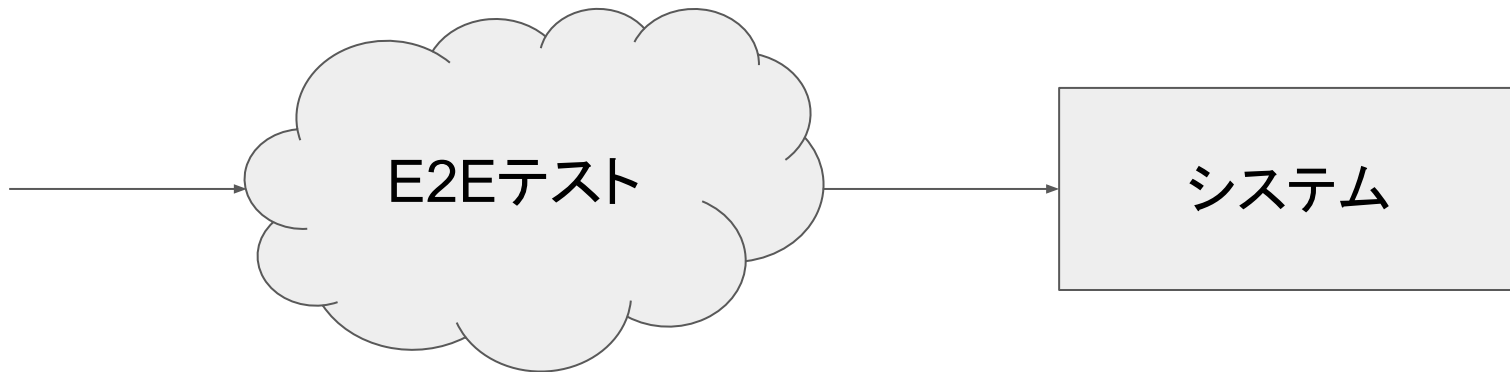
Qिता



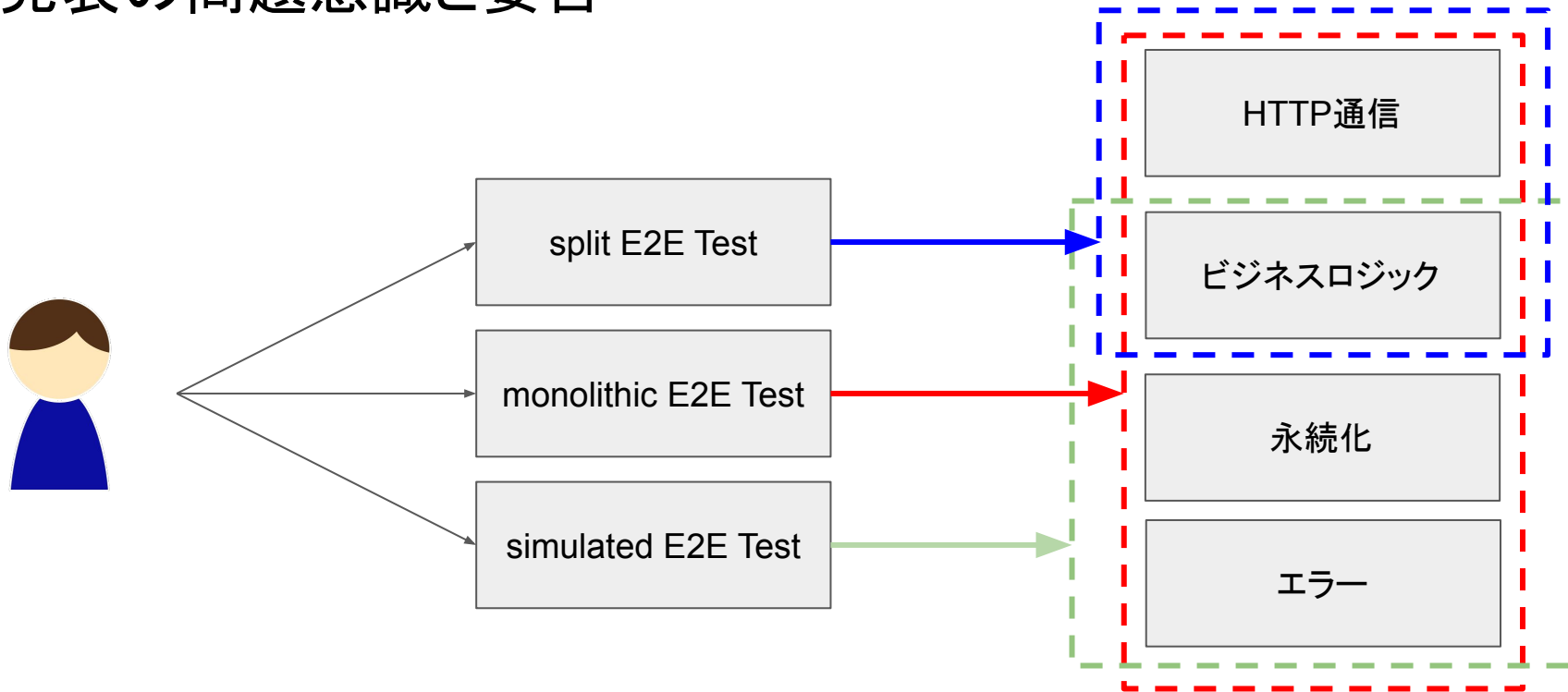
E2Eテスト



それぞれにおいてE2Eテストの定義、テスト可能な範囲が異なる



本発表の問題意識と要旨



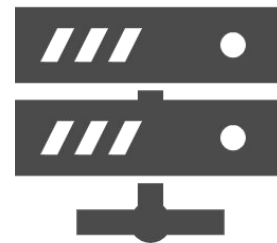
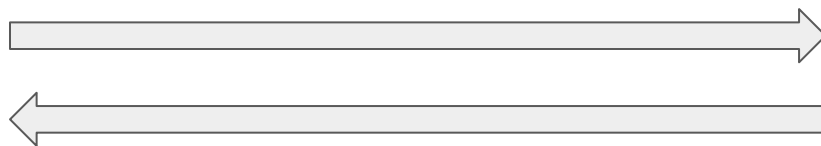
E2Eテストの種類と適用範囲を把握することで、適切な手法を選択できるようにすること。

E2Eテストとは



クライアント

GET /users/1

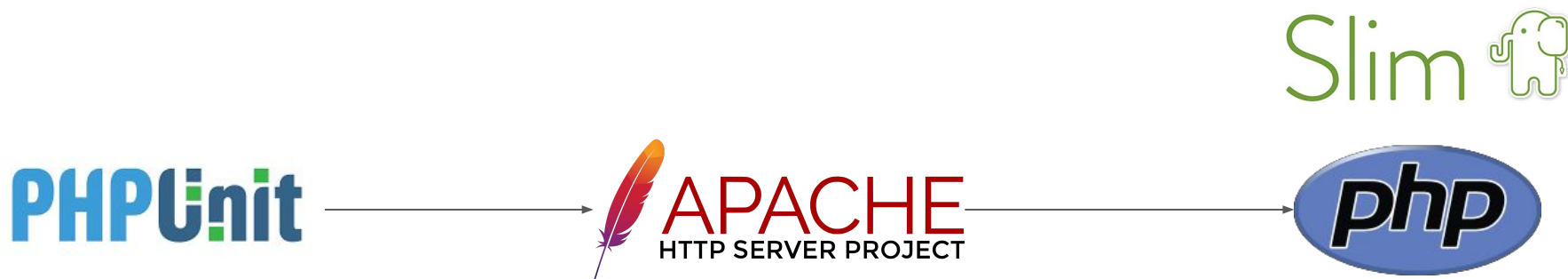


APIサーバー

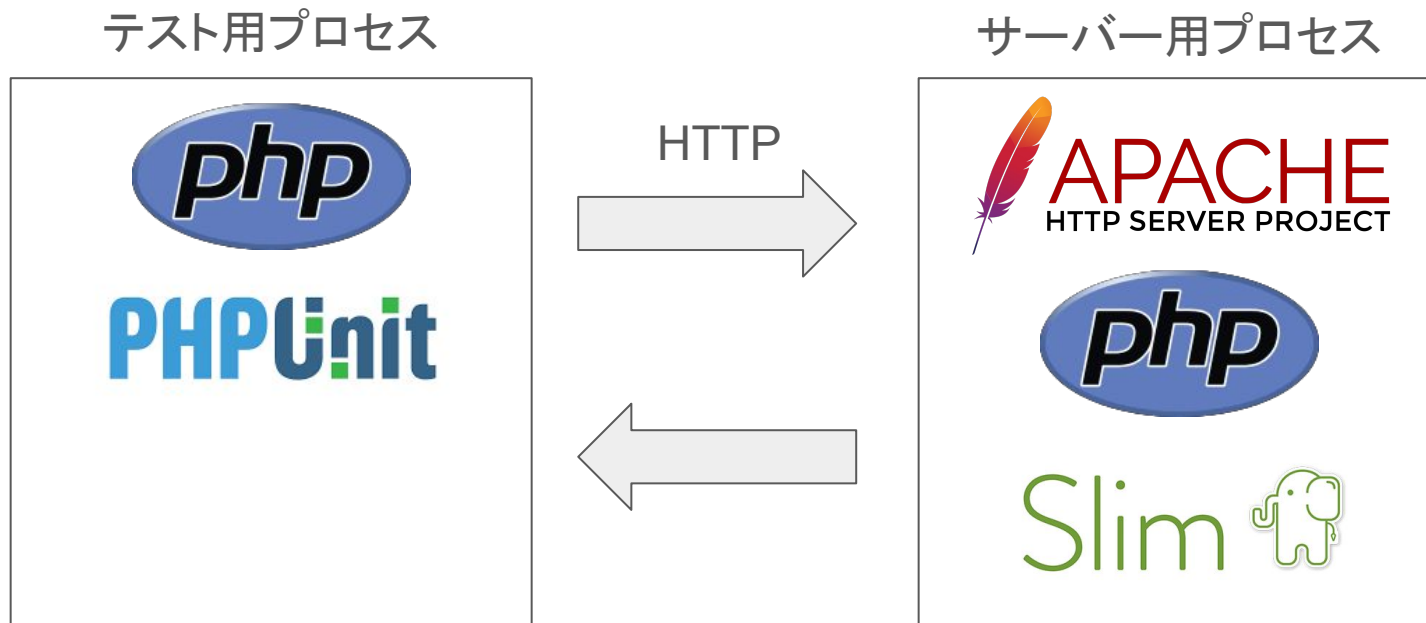
```
{  
  "id" : 1,  
  "name" : "kurauchi"  
}
```

curlなどのHTTPクライアントから、HTTP通信を介して呼び出されたREST APIの機能的な側面の動作が、仕様と一致することを確認するためのテスト

PHPを用いたREST APIのE2Eテストの構成例



split E2E Test



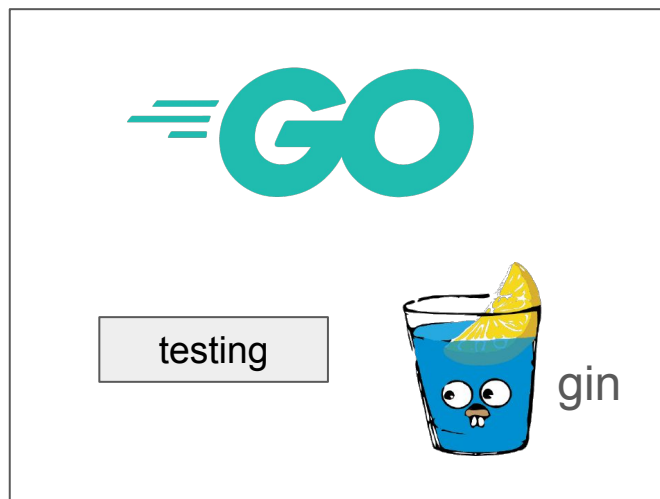
テスト用プロセスとサーバー用プロセスが分離したHTTP通信を介したE2Eテスト
(一般的に認知されているE2Eテストの形式)

Go言語を用いたREST APIのE2Eテストの構成例



monolithic E2E Test

テスト用プロセス



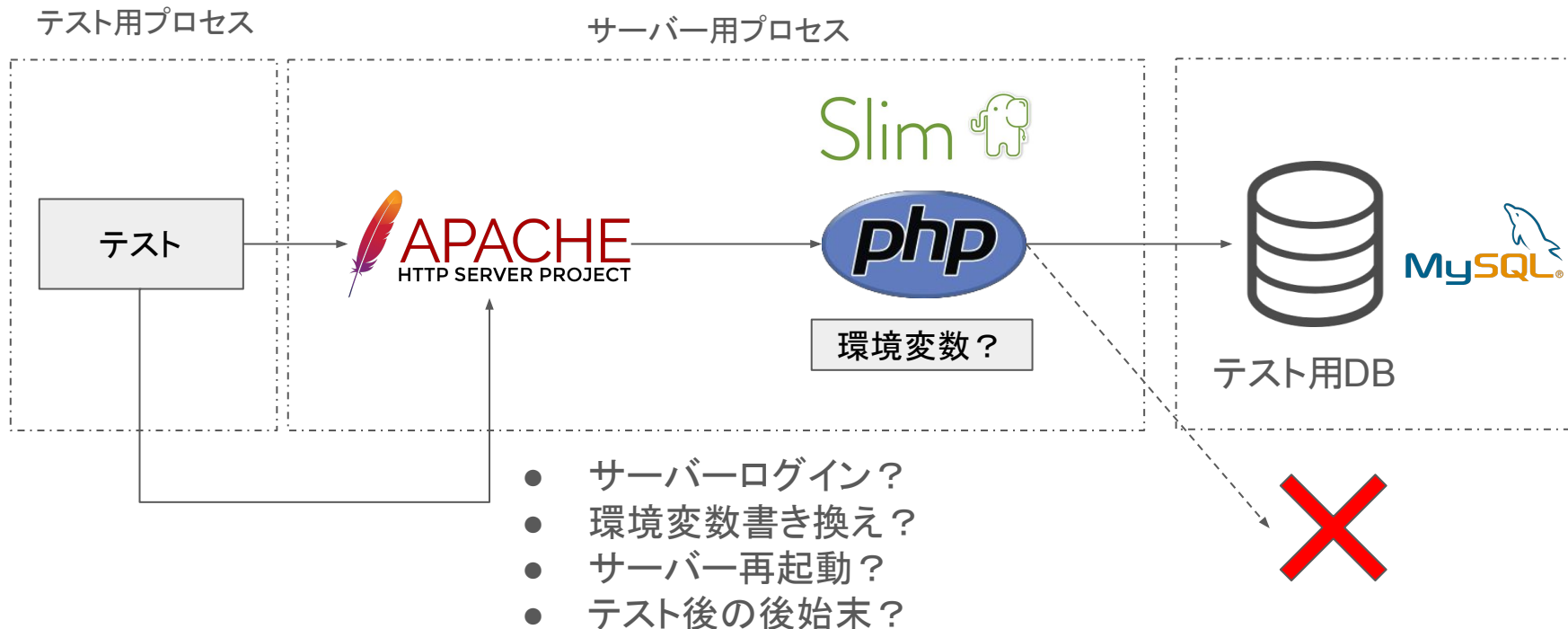
テスト用プロセスとサーバー用プロセスが同一でHTTP通信を介したE2Eテスト

DBの接続できなかったときのテスト



間違ったホストに接続したときに500エラーになるか確認したい。

split E2E Testの構成の場合



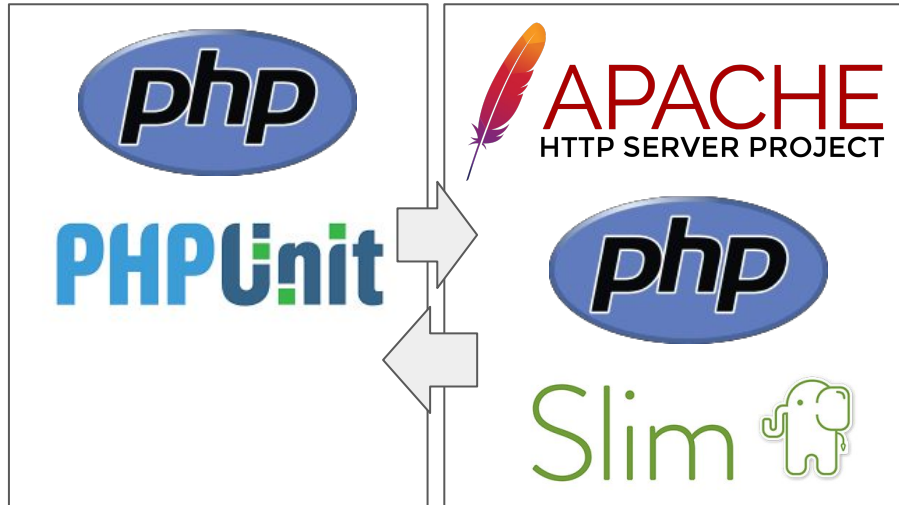
変更の反映・環境の冪等性の確保が困難

split E2E Testとmonolithic E2E Testの比較

split E2E Test

テスト用プロセス

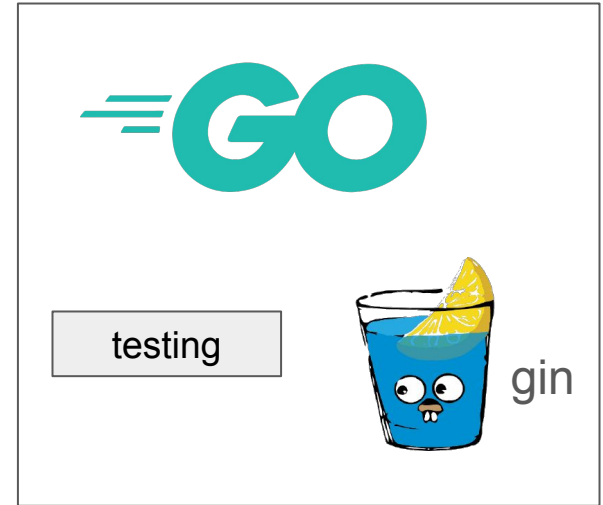
サーバー用プロセス



テスト用プロセスがサーバー用プロセスと分離しているため、APIサーバーの接続先のDBのホスト名変更の設定変更が難しい

monolithic E2E Test

テスト用プロセス



テスト用プロセス内でHTTPサーバーを起動するので、APIサーバーの接続先のDBのホスト名変更の設定変更が可能

monolithic E2E Testの必要条件

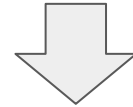
```
// Test for POST /user/add
func TestPostUser(t *testing.T) {
    router := setupRouter()
    router = postUser(router)

    w := httptest.NewRecorder()

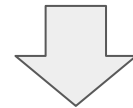
    // Create an example user for testing
    exampleUser := User{
        Username: "test_name",
        Gender:   "male",
    }
    userJson, _ := json.Marshal(exampleUser)
    req, _ := http.NewRequest("POST", "/user/add", strings.NewReader(string(userJson)))
    router.ServeHTTP(w, req)

    assert.Equal(t, 200, w.Code)
    // Compare the response body with the json data of exampleUser
    assert.Equal(t, string(userJson), w.Body.String())
}
```

Go言語は言語機能として
HTTPサーバーが稼働可能



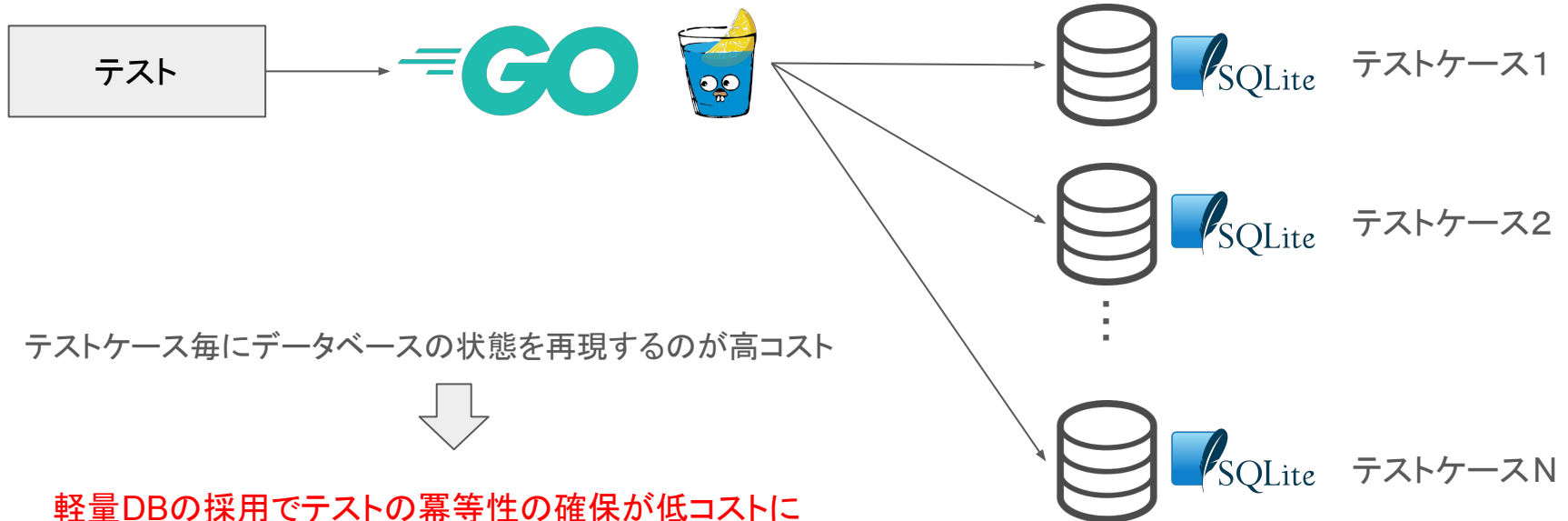
任意のタイミングで
任意の設定の
HTTPサーバーを起動可能



APIの接続するDBのホスト名
の変更などが容易

<https://gin-gonic.com/ja/docs/testing/>

言語機能によりHTTPサーバーを起動出来るメリット



テストケース毎にデータベースの状態を再現するのが高コスト



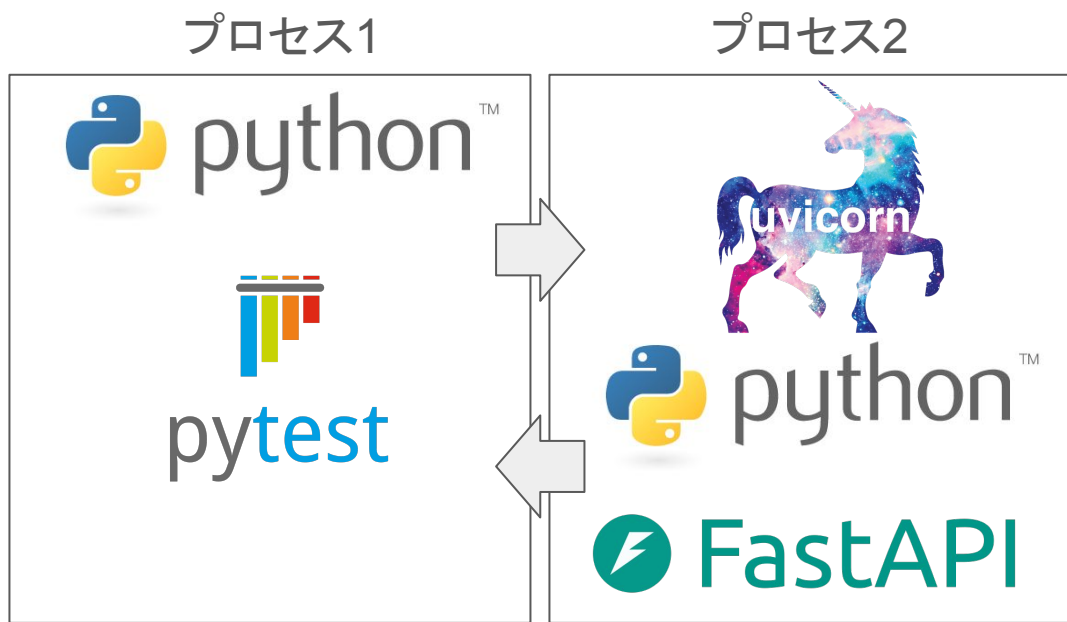
軽量DBの採用でテストの冪等性の確保が低コストに

- データベースの初期化が低コスト
- テストケース毎のデータセットの管理が低コスト

Pythonを用いたREST APIのE2Eテストの構成例



split E2E Test、monolithic E2E Testどちらも出来る



split E2E Test



monolithic E2E Test

FastAPIのテストのチュートリアル

```
from fastapi import FastAPI
from fastapi.testclient import TestClient

app = FastAPI()

@app.get("/")
async def read_main():
    return {"msg": "Hello World"}

client = TestClient(app)

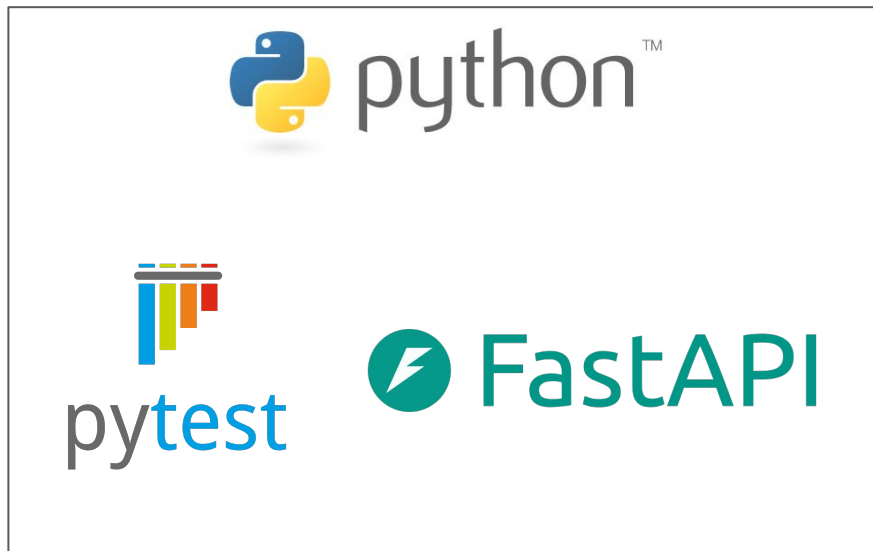
def test_read_main():
    response = client.get("/")
    assert response.status_code == 200
    assert response.json() == {"msg": "Hello World"}
```

HTTPサーバーをテスト時に
起動していない。

<https://fastapi.tiangolo.com/ja/tutorial/testing/#testclient>

simulated E2E Test

テスト用プロセス

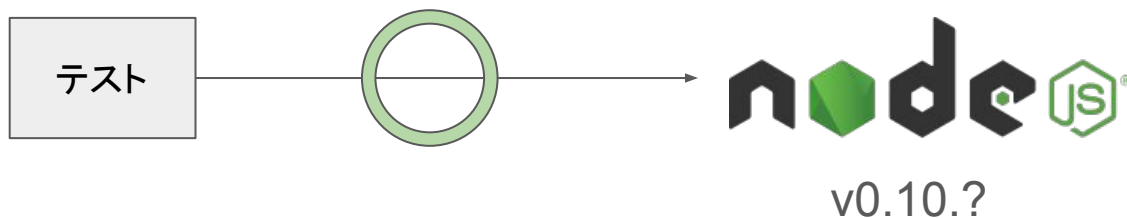


HTTP通信を行わない(HTTPサーバーがない)がE2Eテストと呼ばれるテスト

E2EテストではHTTP通信をしなければならないか？

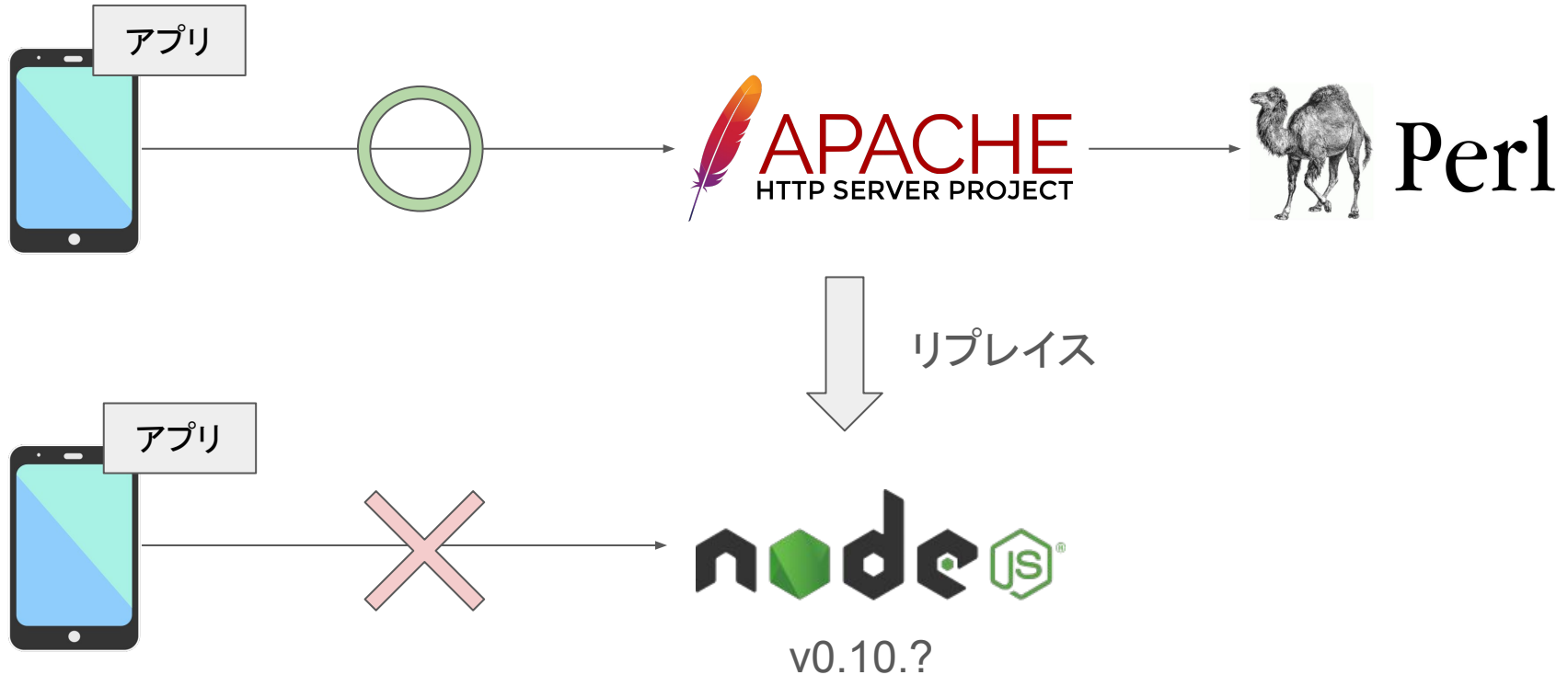


リプレイス



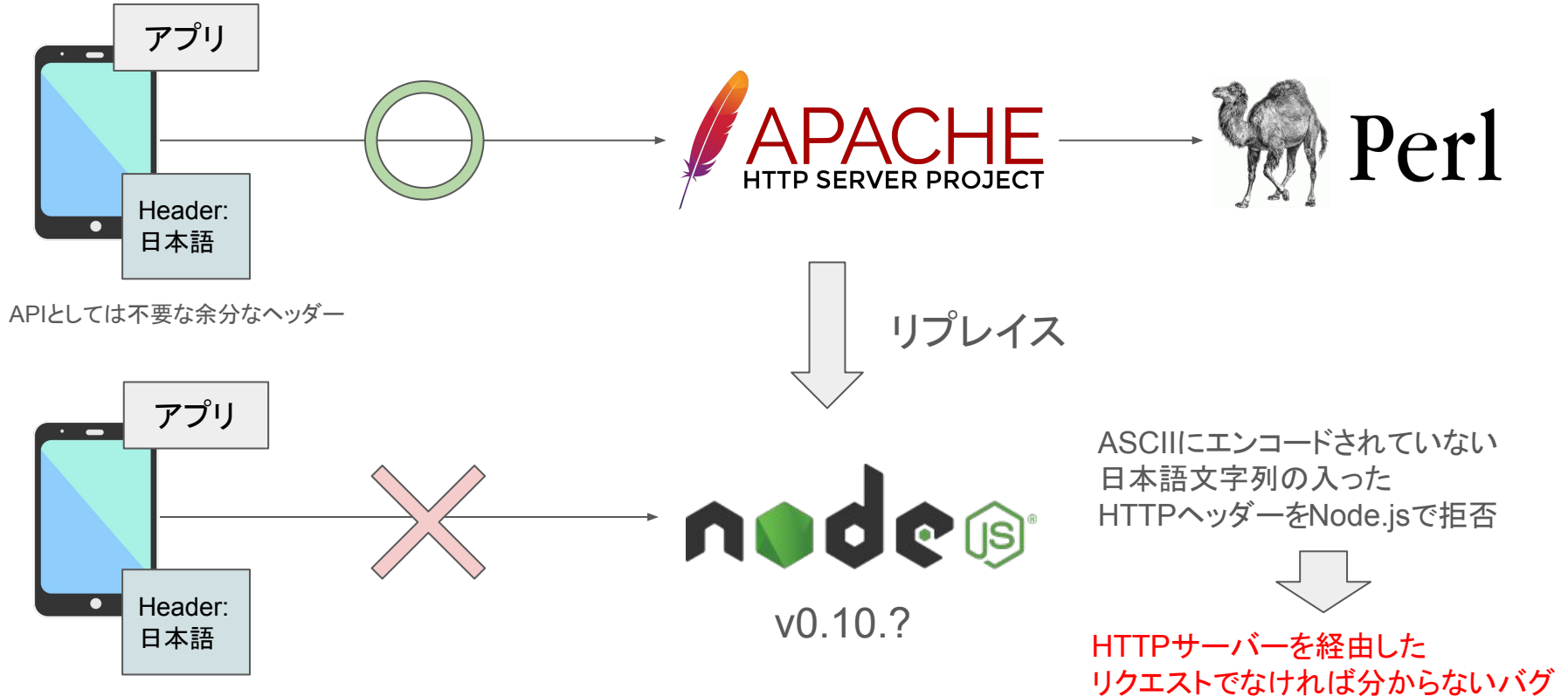
テストからは動作OK

E2EテストではHTTP通信をしなければならないか？



アプリだと動作NG

HTTPサーバーの実装依存の挙動



巨大リクエストボディのE2Eテスト



E2Eテストのアーキテクチャ(HTTP通信の利用)を把握していない場合、NGINXにリクエストが阻まれることが分かりにくい。

→E2EテストでHTTP通信を介しておくほうが本番の再現性は良い。

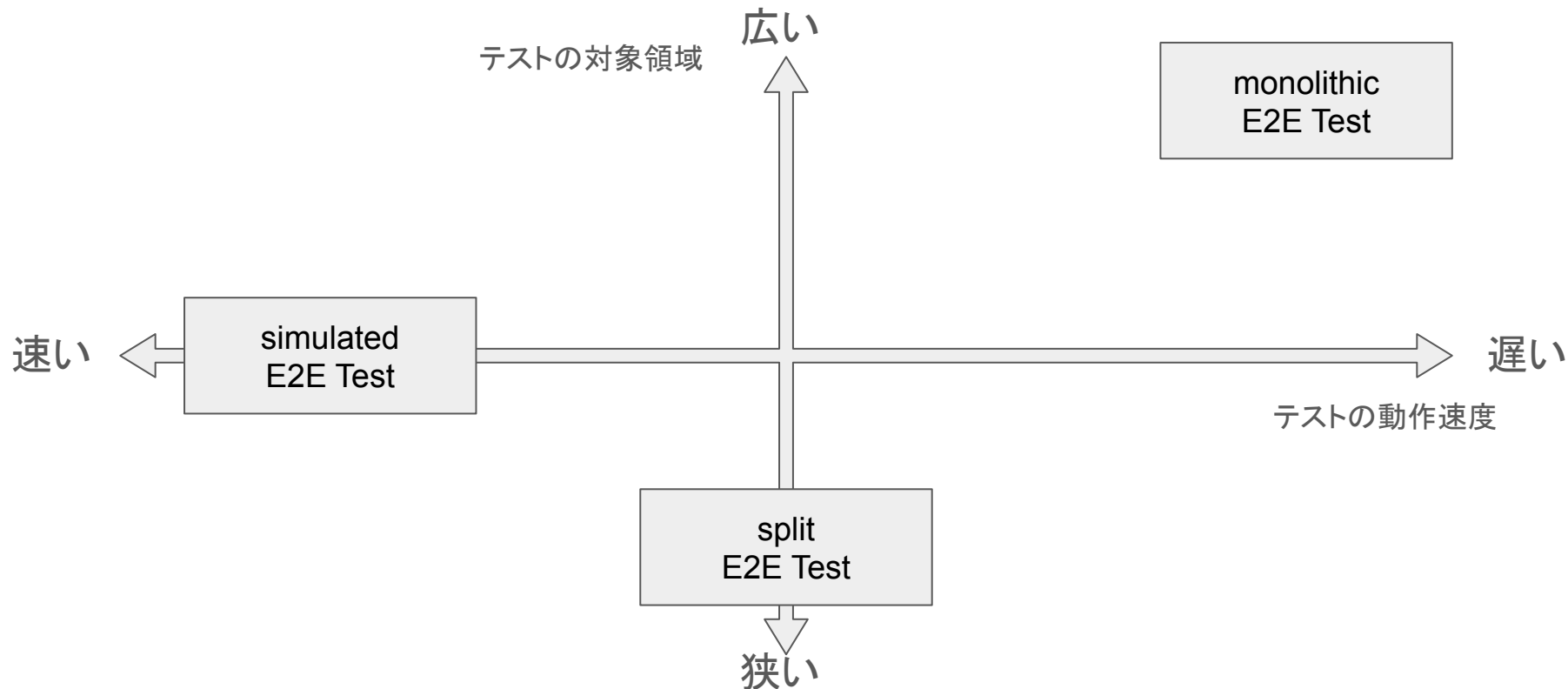
E2Eテストの分類とテスト対象の領域

	業務ロジック	エラー	永続化	HTTP通信	動作速度
split E2E Test	○	×	×	○	普通
monolithic E2E Test	○	○	○	○	遅い
simulated E2E Test	○	○	○	×	速い

※monolithic E2E TestはHTTPサーバーの起動・終了とHTTPリクエスト・HTTPリクエストの処理を繰り返し行っているため、論理的には遅いと考えられる。しかし、データとしての裏付けはない。

一口にE2Eテストといえど、テスト対象領域とテスト速度が異なる。

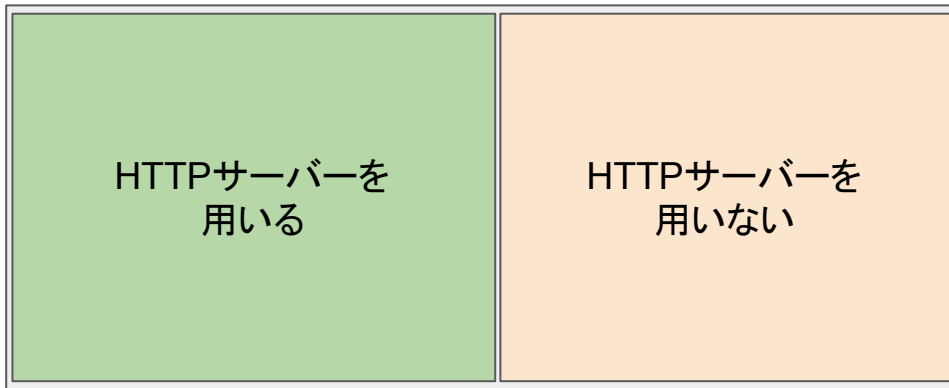
E2Eテストの動作速度とテストの対象領域



E2Eテストの分類について

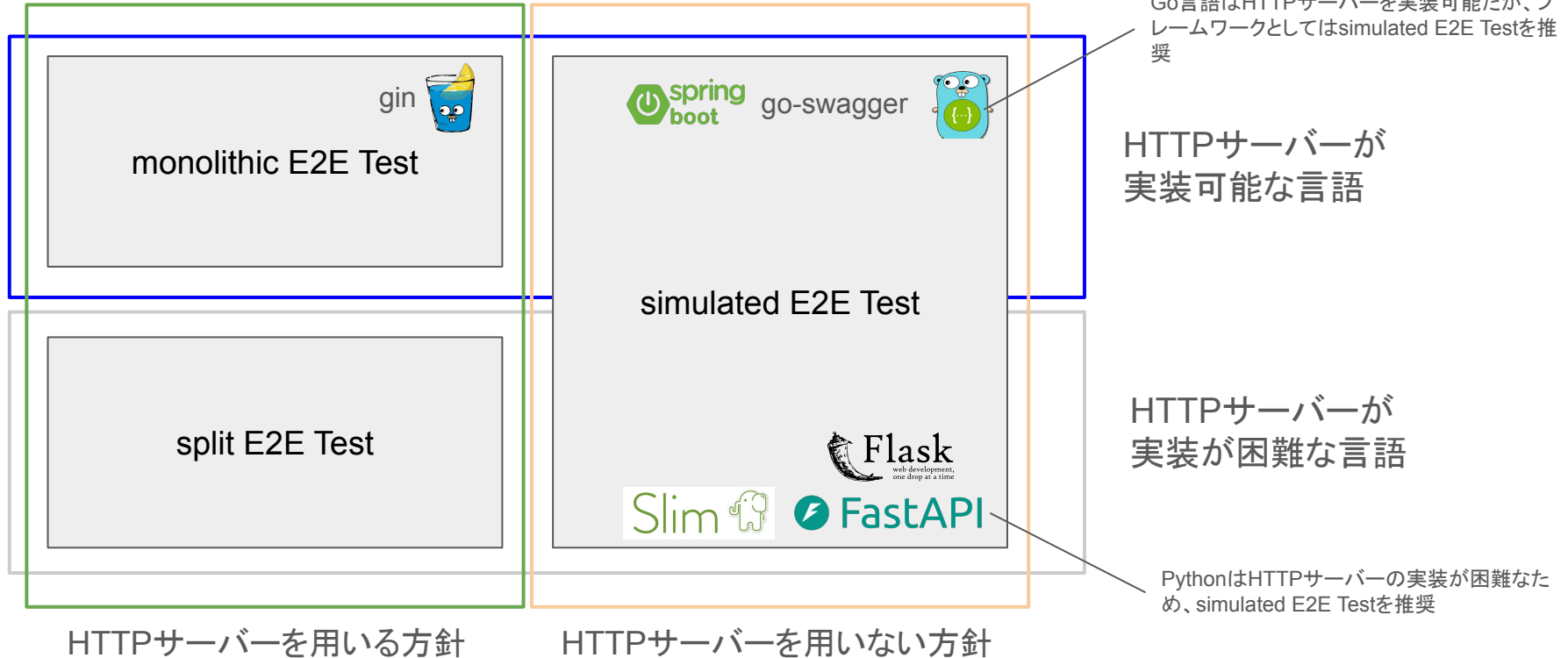


プログラミング言語



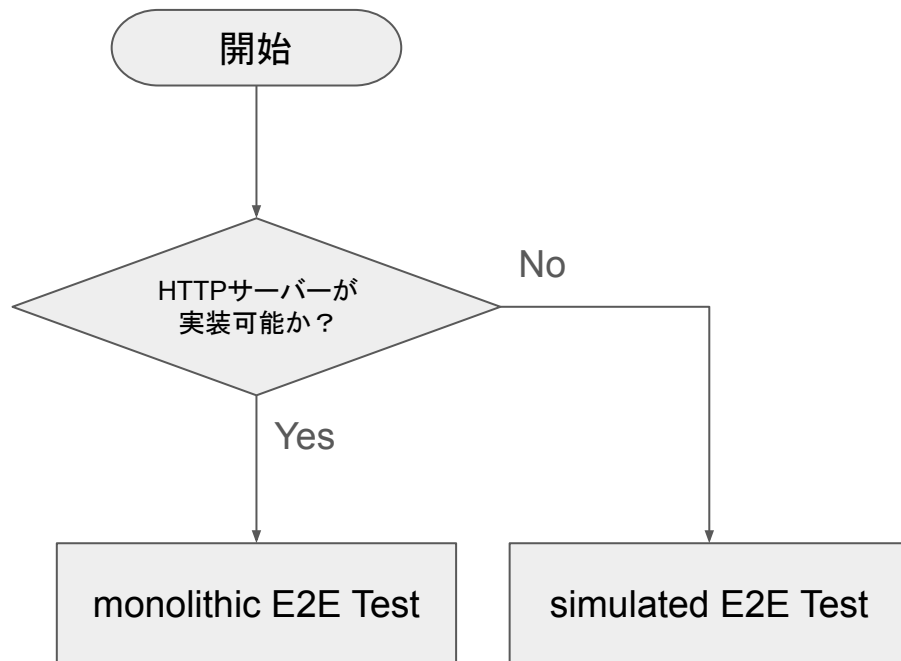
Webフレームワークのテスト方針

E2Eテストの分類について



プログラミング言語の能力も関連があるがライブラリの推奨方針による違いもある

E2Eテストの手法の選び方



monolithic E2E Testには
プログラミング言語のレベルで
HTTPサーバーが実装できることが必須

- Node.js
- Golang
- Java
- C#

monolithic E2E Testしにくい言語

- PHP
- Python
- Ruby
- Perl

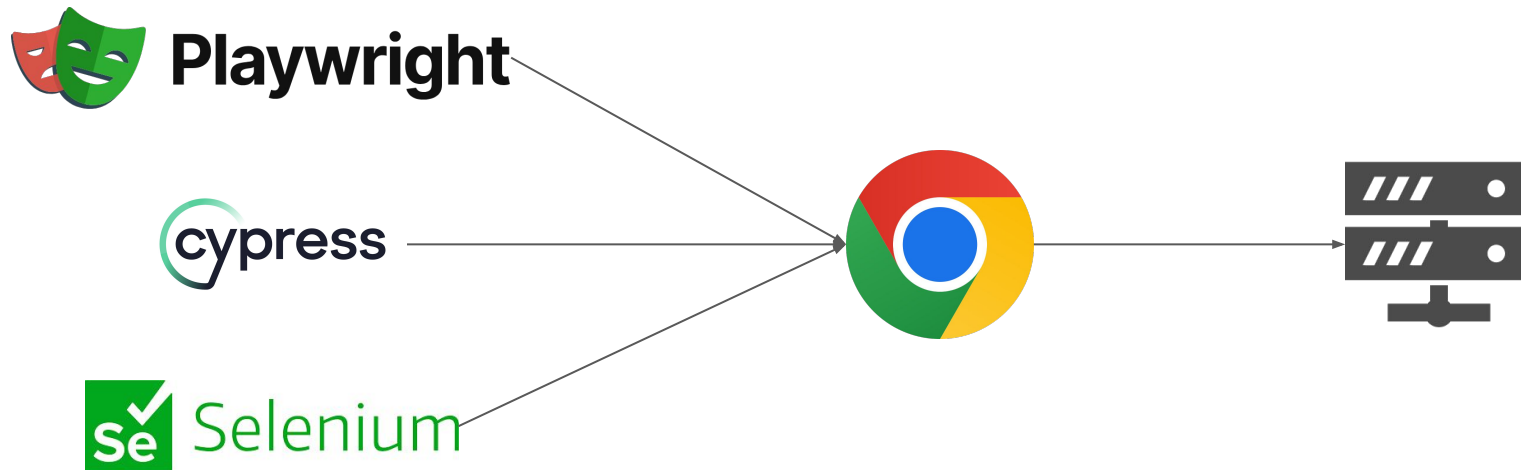
split E2E Testはどのような場合を選ぶべきか？

- split E2E Testをあえて選ぶ必要性はかなり少ない。
- レガシーな環境(LAMP等)を抱えているときに選ばざるを得ない。



1つのAPIに対するテストというより複合したシステムのテストをする場合、採用があるかもしれない。

split E2E Testのよくあるテスト対象



split E2E Test型のアーキテクチャはフロントエンド系に多い。
ブラウザの挙動をテストする必要があるので、テストとブラウザのプロセスが分かれる。
REST APIのテストのアーキテクチャとしては、積極的に採用する印象はない。

まとめ

- E2Eテストは3種類存在する
 - split E2E Test
 - monolithic E2E Test
 - simulated E2E Test
- それぞれのE2Eテストにテスト対象が異なり、テストの速度が異なる
- プログラミング言語でHTTPサーバーが実装可能か。フレームワークのテストはどのようなテスト方針をしているか。を把握し、自分でその適用範囲、テスト可能な領域を把握していることが大切。
- お勧めのE2Eテストの採用方針
 - 1. monolithic E2E Test
 - 2. simulated E2E Test
 - 3. split E2E Test
- monolithic E2E Testはプログラミング言語でHTTPサーバーの実装があることが必須になる。