Session Title

fake clock microservice
～時刻をハックしてテストする方法～

vvakame / Hiraku Nakano / Hiroyuki Tanaka

vvakame

Merpay Solutions Team

Solutionsチームで働くねこ大好きエンジニア。GoとかGraphQLとかが得意。今回は社内ツール作成第一人者（自称）としてfakeclock改善に乗り出した。

Hiraku Nakano

Merpay Credit Design Team

2015年メルカリ入社。メルペイでは銀行接続、NFC決済など色々なmicroserviceのTech Leadを経て、今はCredit Designチームのバックエンドエンジニアとして従事。

Hiroyuki Tanaka

Merpay Credit Design Team

2021年1月にメルペイに入社。Credit Designチームのバックエンドエンジニアとして、日々信用を創造してなめらかな社会を創っている。

ご質問をお待ちしております！

本セッションはリアルタイムに実施しています。

YouTubeのチャットに投稿するか、

X（Twitter）のハッシュタグ

#MerpayMercoinTechFest を

ご利用ください。

このセッションの構成

01 時刻とテストの問題について解説

02 fake clock serviceについて

03 トークセッション

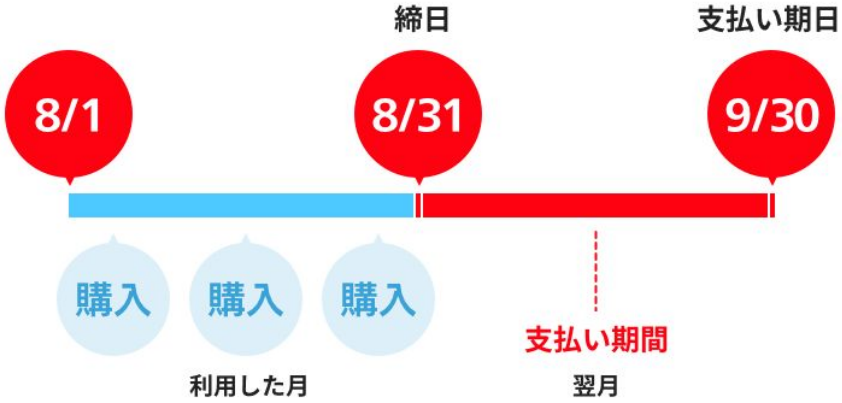
金融領域にありがちなライフサイクル

※メルペイのあと払いの例

どのようにテストするか？

よくあるテスト方法

- システム時刻を8/1に変更
  - 購入をテスト
- システム時刻を9/1に変更
  - 請求バッチをテスト
- システム時刻を10/1に変更
  - 延滞バッチをテスト
  - 延滞後の支払いをテスト
- …



これまでの時刻操作機能

ビジネスロジック上の「現在時刻」を環境単位で変更する機能を、各マイクロサービスで実装。時刻操作APIをテスト環境限定で開放する

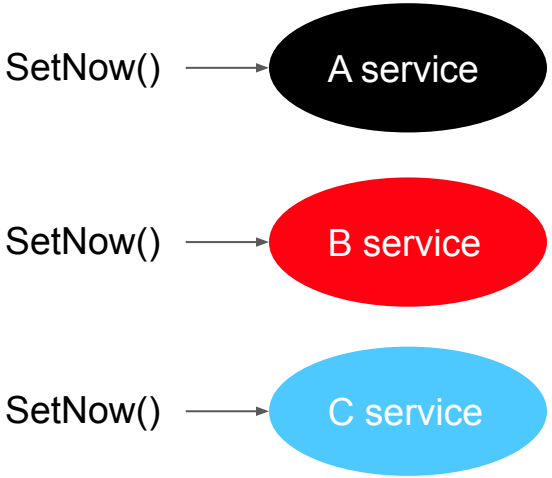
- **debug.SetNow("2022-08-01")**
  - 購入をテスト
- **debug.SetNow("2022-09-01")**
  - 請求バッチをテスト
- **debug.SetNow("2022-10-01")**
  - 延滞バッチをテスト

既存の時刻操作の問題点

01. 時刻操作したいマイクロサービスが複数ある
    a. マイクロサービスは協調動作する場合が多い
02. デバッグが大変
03. 環境を専有し、並列にテストしづらい

複数マイクロサービスがある問題

- ビジネスロジックに関係するマイクロサービスすべてに対してそれぞれ時刻操作が必要
  - 「時刻設定忘れ」のミスが起きがち
- みんな再実装しているのも無駄



デバッグが大変

「時刻設定忘れ」「時刻ズレ」は不可解な挙動を引き起こす

「時間停止状態」ならではのバグ

- 同じ時刻に2件snapshotレコードを保存しようとしてエラーになるなど

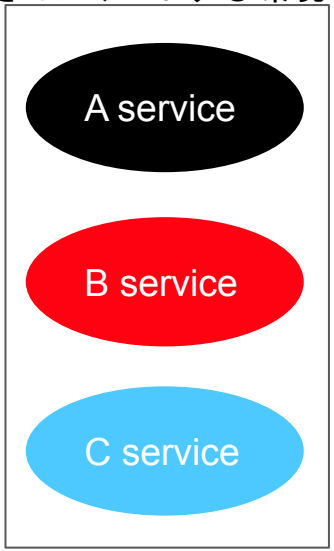
本番では発生し得ない事象のデバッグに追われる

環境を専有する

環境自体の時刻を固定化してしまうため、設定を分割するために
複数のマイクロサービス環境を用意しているが、無駄が多い

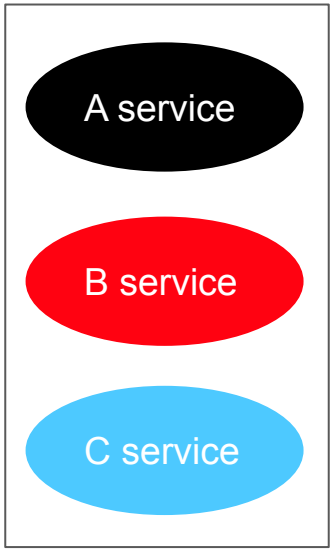
Aさんがテストする環境

- **システム時刻を8/1に変更**
  - 購入をテスト
- **システム時刻を9/1に変更**
  - 請求バッチをテスト
- **システム時刻を10/1に変更**
  - 延滞バッチをテスト
  - 延滞後の支払いをテスト
- …



Bさんがテストする環境

- **システム時刻を8/1に変更**
  - 購入をテスト
- **システム時刻を9/1に変更**
  - 請求バッチをテスト
- **システム時刻を10/1に変更**
  - 延滞バッチをテスト
  - 延滞後の支払いをテスト
- …



求めているもの

01 一度の操作で各マイクロサービスの時刻を変更したい

02 環境ごとではなく、リクエストごとに時刻操作をしたい

03 手軽に時刻を操作したい

検討案

01 時刻を固定した環境を複製する

02 メタデータとして伝播させる

03 時刻操作用のマイクロサービスを作る

時刻を固定した環境を複製する

必要なタイミングでマイクロサービス群を複製してしまう

Pros

- 状態が本番環境に近い

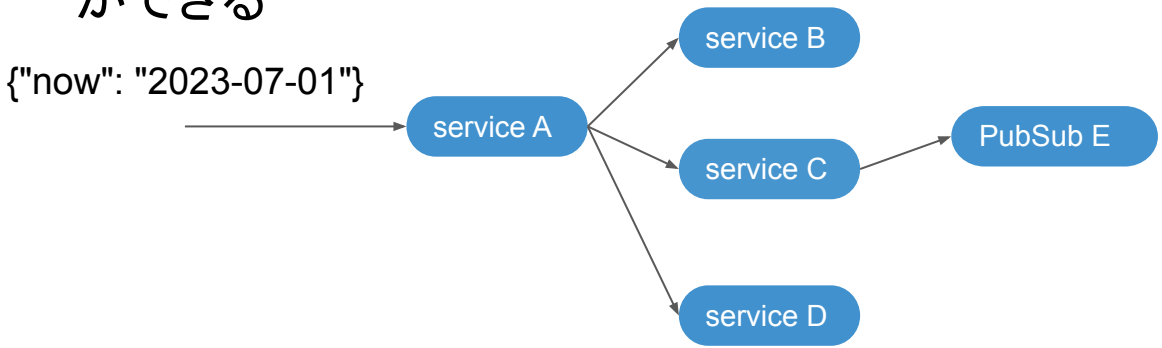
Cons

- 環境立ち上げに数分かかるので、手軽にテストができない

メタデータとして伝播させる

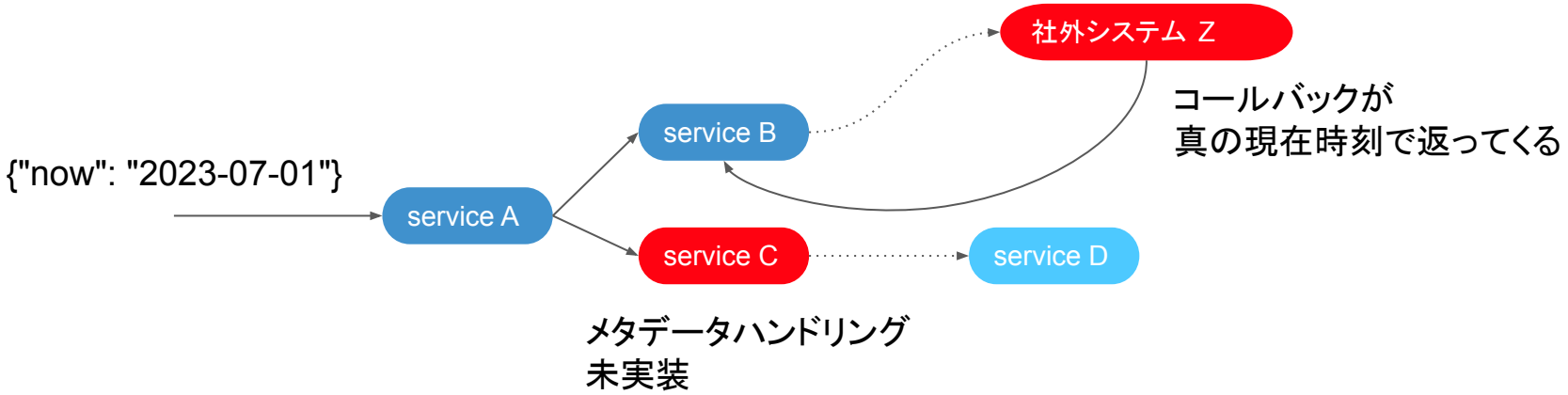
gRPCやHTTP, PubSubのメタデータで時刻設定を含めておく
その時刻でビジネスロジックを動作させる
下流のサービスへのリクエストでも設定を伝播させていく

✅時刻設定がリクエスト単位になり、1つの環境をシェアして並列テストができる



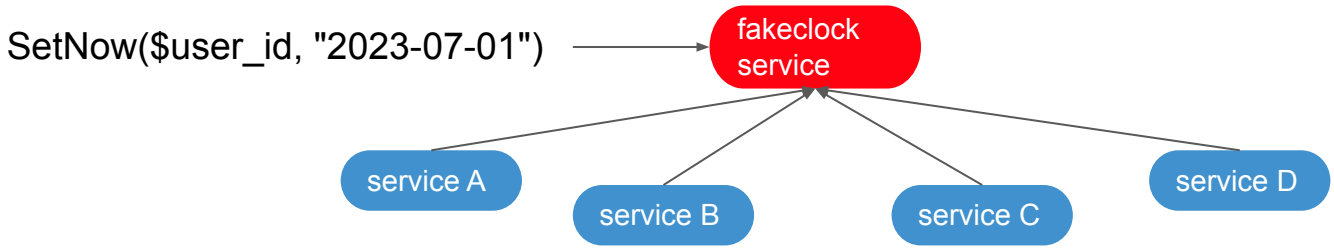
メタデータ案で問題視された点

- 伝播がどうしても止まってしまう箇所があった
  - 例: 社外システム
- ルート上のどこかにメタデータの伝播の未実装があると、動作がおかしくなる



時刻操作をユーザー単位にする管理用サービスを作る

各サービスは処理中のuser_idに対応する時刻設定をfakeclock serviceから取得する



fakeclock service案

Pros

✅ テストがユーザー単位のため、別ユーザーを使えば並列性は確保できる

✅ 外部システムが系に入っていても、時刻を強制的に復元できる

✅ サービス毎の実装を共通化する効果

✅ 未実装サービスがあっても、導入済みサービスは恩恵がある

Cons

😇 本番環境と構成が違う

😇 新たなサービス運用 (テスト環境限定のSPOF)

fakeclock導入の現状

導入、難しい！
最初から導入していれば…

実はこれから各マイクロサービスへ導入していくフェーズです。
運用が開始されてからの知見はまた後日共有したい。

SDKはできてる + 設定用MS(user-tkool)は稼働している + パイロットのMSにコード上の変更はできたが検証工数が…

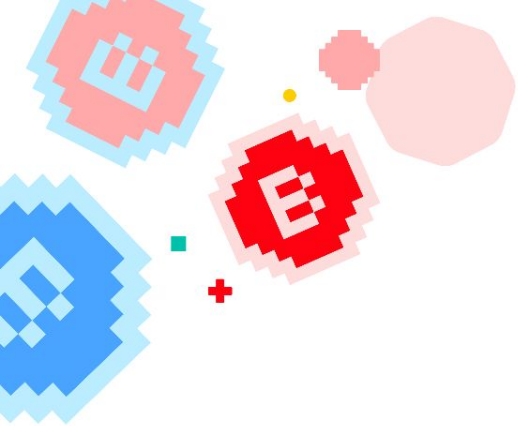
質問の確認

- YouTube, SNS のコメントを一回確認します！
- #MerpayMercoinTechFest

パネルディスカッション

パネルディスカッション (テーマ)

- 今どこまで進んでいるのか
- 導入時のQAどうするのか問題
- グループ全体から見た位置づけ
- 時刻操作あるあるトーク
  - OS時刻ずらしすぎてtokenが失効して通信できなくなる
- マイクロサービスとe2eテスト
- ここがすごいよ新fakeclockサービス



Thank you!