事前課題2

非同期ジョブシステム

- 複数の Web アプリケーションから共通で使える非同期ジョブの基盤を提供する
  - 対象となるのは時間がかかる処理。例えば、動画のエンコーディングや大量のサムネイルの作成など。
- 非同期ジョブが失敗した場合は、自動的に5回リトライする
- 5回失敗した非同期ジョブはシステム管理者にエラー通知する
- 非同期ジョブが終了したら、成否に関わらず結果をユーザに通知したい。
  - これは可能か？どのような制約をつければ実現できるか？

Architecture Discussion

Make a presentation about an architecture of the system below. You can use any software, cloud services and protocols. In the presentation, you can bring in documents or write it on whiteboard there.

Asynchronous Job System

- We want to create asynchronous job system for our many web applications
  - such as encode video, create thumbnails
- We want to retry 5 times when asynchronous job is failed
- If asynchronous job failed 5 times, notify to our system admin
- After job finished, even if it's failed, notify the result to users

---

アーキテクチャ

Webアプリケーション

- (複数の)Webアプリケーションは以下のコンポーネントを持つ
  - userからの要求を受け付けるWeb
    - Backendを実行するAPI情報を持っていること
  - (時間がかかる)処理を実施するBackend
    - APIで処理を受け付けられること

ジョブ基盤

- ジョブ基盤は以下のコンポーネントを持つ

    - APIGW ... APIを受け付けるGW
    - MQ ... メッセージキュー。複数のアプリケーションから使われることを想定。
    - FaaS ... Function as a service(ex. AWS Lambda)。イベントをトリガーに処理を行う。ステートレス。
    - job-DB ... ジョブのリトライ回数、ジョブIDを持つ。
    - user-DB ... Webアプリケーション利用者のメールアドレスを格納しているDB。
    - Mail ... メールサーバ。ユーザや管理者にメール通知を行う。

要求仕様(問題より)

- 非同期ジョブが失敗した場合は、自動的に5回リトライする
    - job-DBにjobの状態を保存することで実現する
- 5回失敗した非同期ジョブはシステム管理者にエラー通知する
    - メール通知とする。ジョブ基盤はシステム管理者のメールアドレスを知っている、あるいはuser-DBに格納されているとする
- 非同期ジョブが終了したら、成否に関わらず結果をユーザに通知したい。
    - メール通知とする。ジョブ基盤を利用するアプリケーションは、ユーザのメールアドレスを事前にuser-DBに格納することを制約とする。

処理シーケンス

どのように要求仕様を実現するか、2つのパターンの処理シーケンスを説明する。

1回で成功するパターン

王道ルートでも、3回WebAPIのやりとりを行う。

1. userから処理を受け付けたapp-webからジョブ基盤APIGWへ
2. ジョブを受け付けたFaaSからapp-backendへ
3. ジョブが終了したapp-backendからジョブ基盤APIGWへ

それぞれが持つ情報を以下に示す。

1つめのPOST仕様

- userid
- app-id(app1)
- app1のbackendAPI

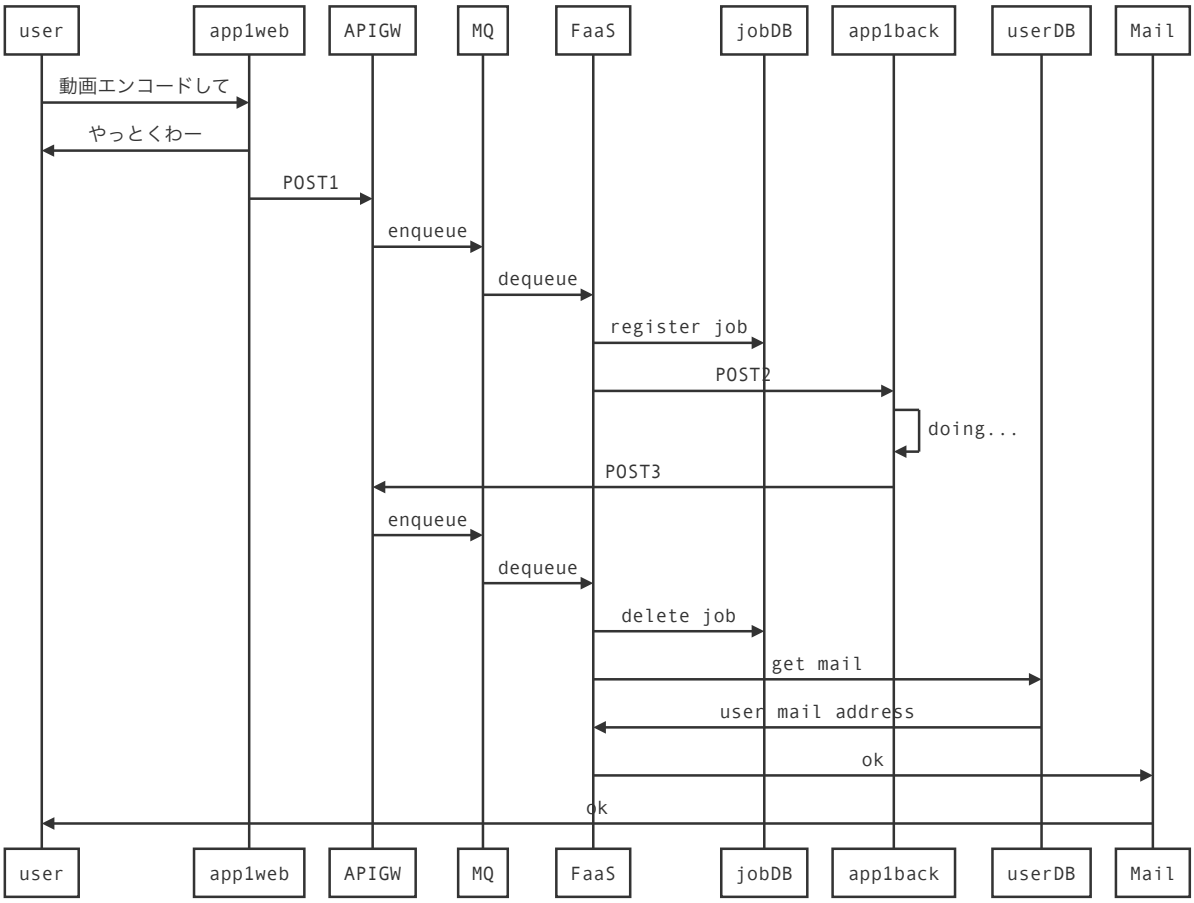
2つめのPOST仕様

- userid
- app-id(app1)
- job-id(DBから取得)

3つめのPOST仕様

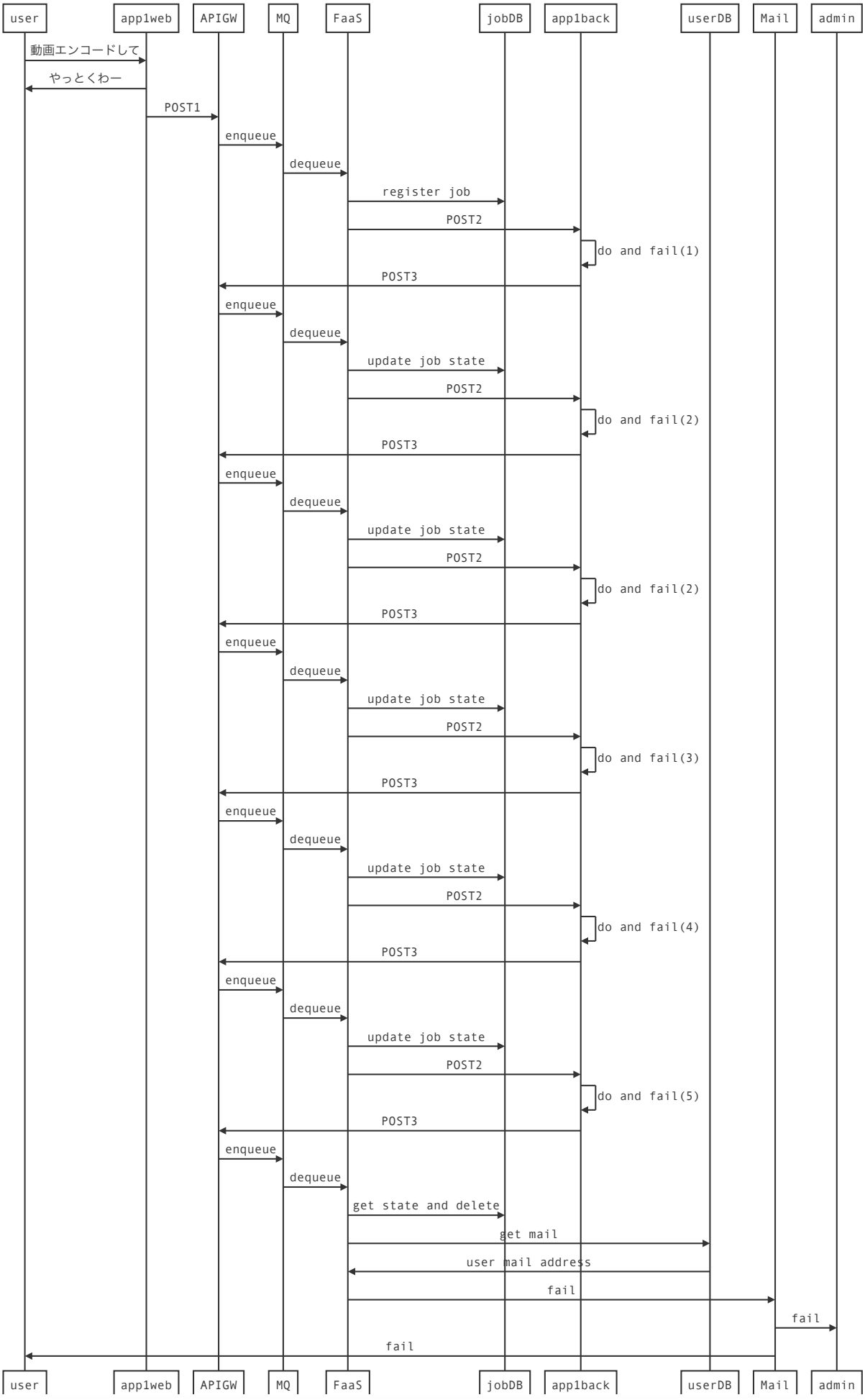
- userid

- app-id
- job-id
- result(ok)



5回連続失敗するパターン

3つめのPOSTで、backendからジョブ基盤にresult(fail)を通知することとなる。


user
app1web
APIGW
MQ
FaaS
jobDB
app1back
userDB
Mail
admin
動画エンコードして
やっとくわー
POST1
enqueue
dequeue
register job
POST2
do and fail(1)
POST3
enqueue
dequeue
update job state
POST2
do and fail(2)
POST3
enqueue
dequeue
update job state
POST2
do and fail(2)
POST3
enqueue
dequeue
update job state
POST2
do and fail(3)
POST3
enqueue
dequeue
update job state
POST2
do and fail(4)
POST3
enqueue
dequeue
update job state
POST2
do and fail(5)
POST3
enqueue
dequeue
get state and delete
get mail
user mail address
fail
fail
fail
user
app1web
APIGW
MQ
FaaS
jobDB
app1back
userDB
Mail
admin

参考

- サーバーレスアーキテクチャのパターン別ユースケース - 非同期Webジョブ
- サーバレスアーキテクチャを実戦投入するにあたって知るべきこと
    - 長時間のバッチ処理は向いてない
    - FaaSはステートレスが前提
- クラウド・ネイティブのお作法（1）「非同期処理」～24時間365日ダウンタイムがゼロのシステムのために必要なこと
    - **非同期処理**（Asynchronous process）
    - **リトライ**（Retry with Exponential Backoff and jitter）
    - **結果整合性**（Eventual Consistency）
    - **冪（べき）等性**（Idempotency）
- SQS。マネージドのメッセージキュー