

1200万  
ダウンロード突破！



# BrainWarsを支える AWSサービスたち

2015.06.03

株式会社トランスリミット CTO

松下 雅和 (@matsukaz)



松下 雅和

@matsukaz

株式会社トランスリミット

ダブルCTOの一人<sup>(\*)</sup>

カメラ, 折りたたみ自転車,  
卓球, ボウリング

(\*) <http://hirokitakaba.github.io/blog/2015/04/03/double-cto/>

# 主な担当

- 【インフラ】 AWS, Google Cloud Platform, Chef
- 【サーバ】 Ruby on Rails, Node.js, 各種スクリプト
- 【フロント】 cocos2d-x
- 【ChatOps】 slack + (hubot x 6)
- 【開発 / 運用】 開発環境用Docker, Adminサーバ,  
データ分析, クラッシュレポート分析
- 【その他】 ≡ ≡ ≡ c (^っ°Д°)っ なんでも球拾い

BrainWars概要

アーキテクチャ基本方針

BrainWarsとAWSサービスたち

まとめ

BrainWars概要

アーキテクチャ基本方針

BrainWarsとAWSサービスたち

まとめ



BrainWars  
ご存知ですか？



# ゲームの流れ

# ホーム



同一リーグの  
ユーザと対戦

バトル開始！



マッチング



# 3ラウンド・バトル



(現時点で)  
29種類のゲームの  
中からランダムで決定

1ラウンド  
20秒バトル

x3

スタンプによる  
ノンバーバル  
コミュニケーション

# バトル結果



3ラウンドの  
合計点数で勝敗決定

各種SNSへ  
結果を共有



## ランキング



面白そうでしょ？  
(・A・)



# BrainWars

# 12,000,000

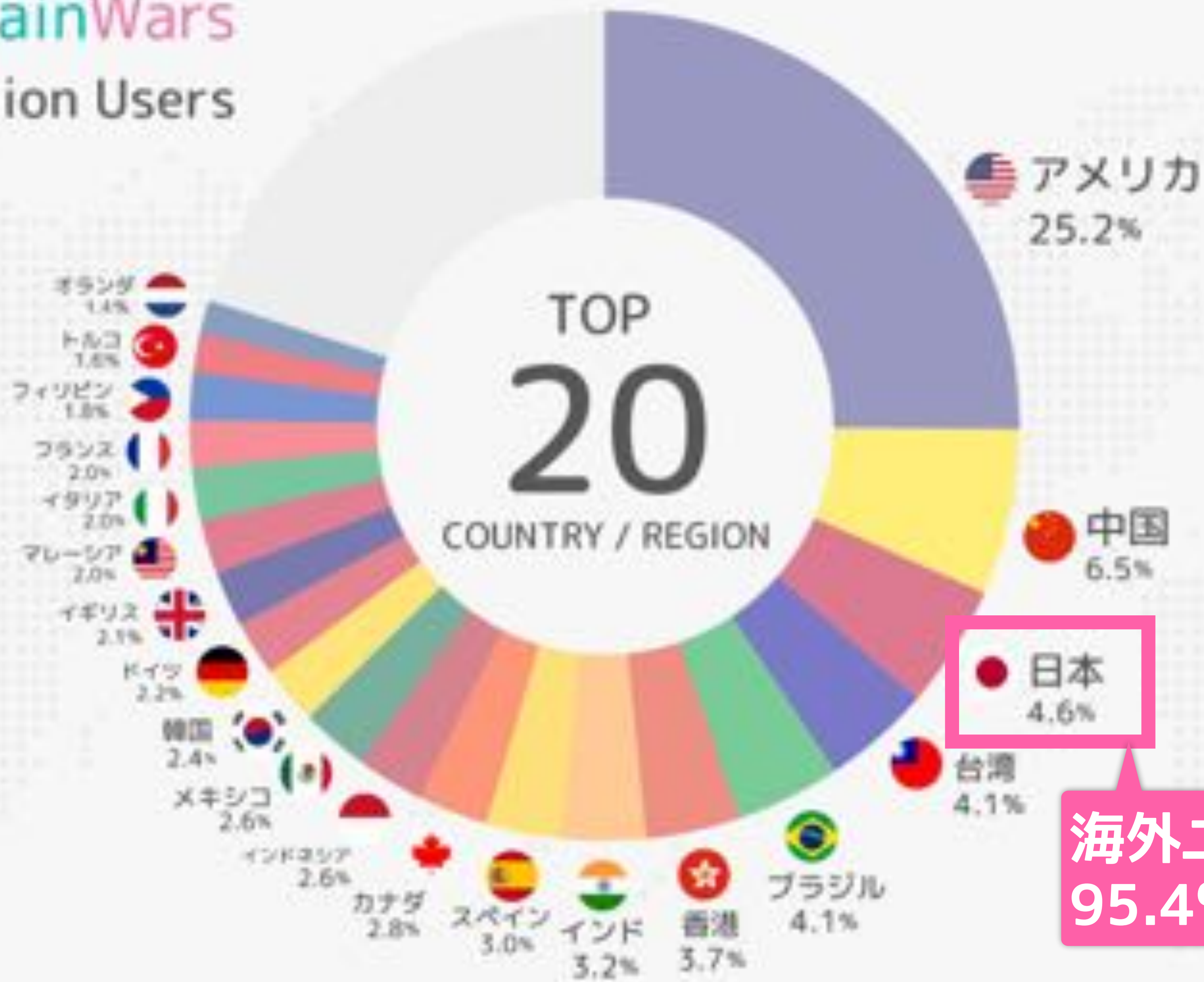
## Downloads



11ヶ月で  
1200万突破！



12 Million Users

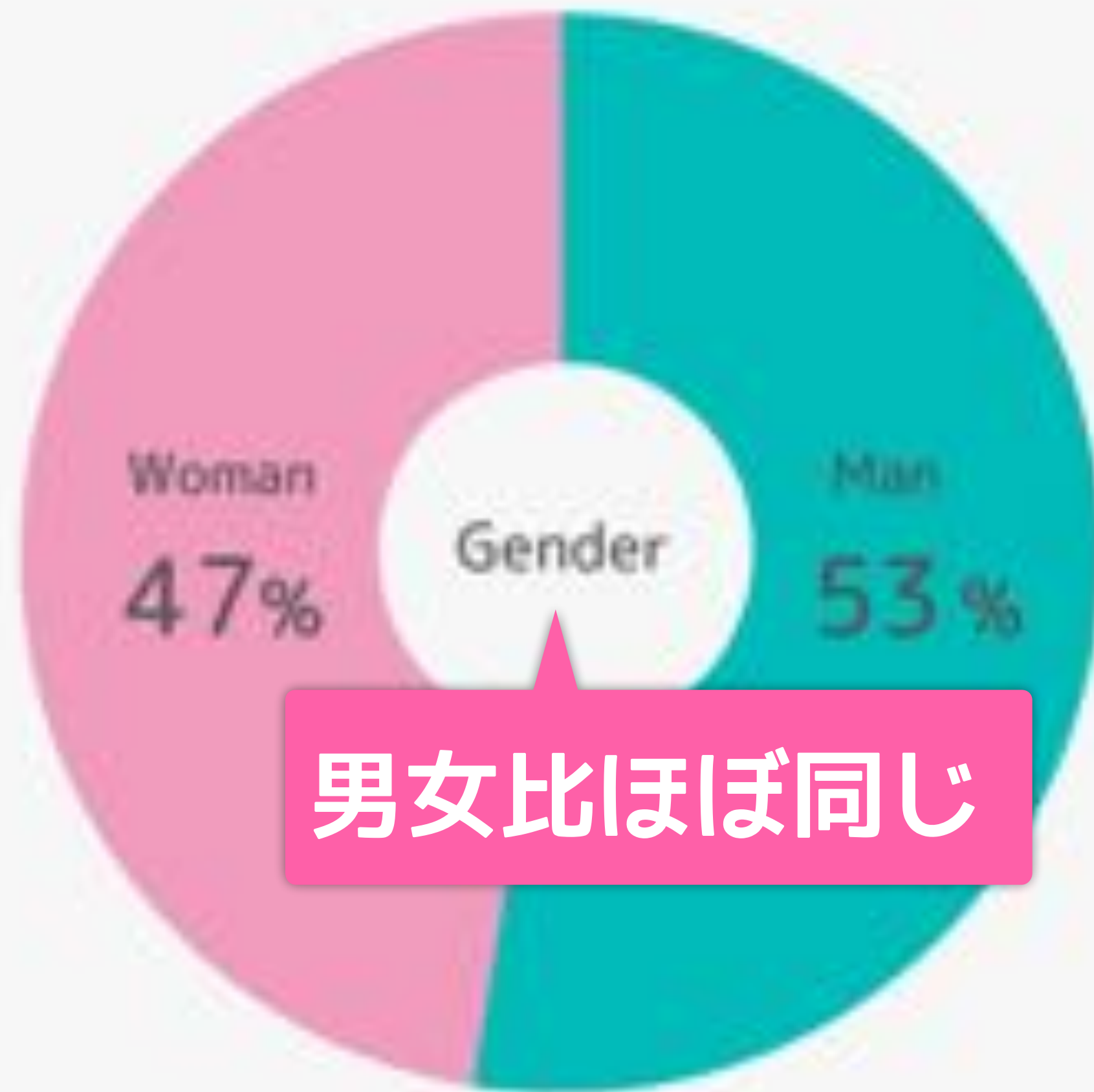


海外ユーザ比率  
95.4%

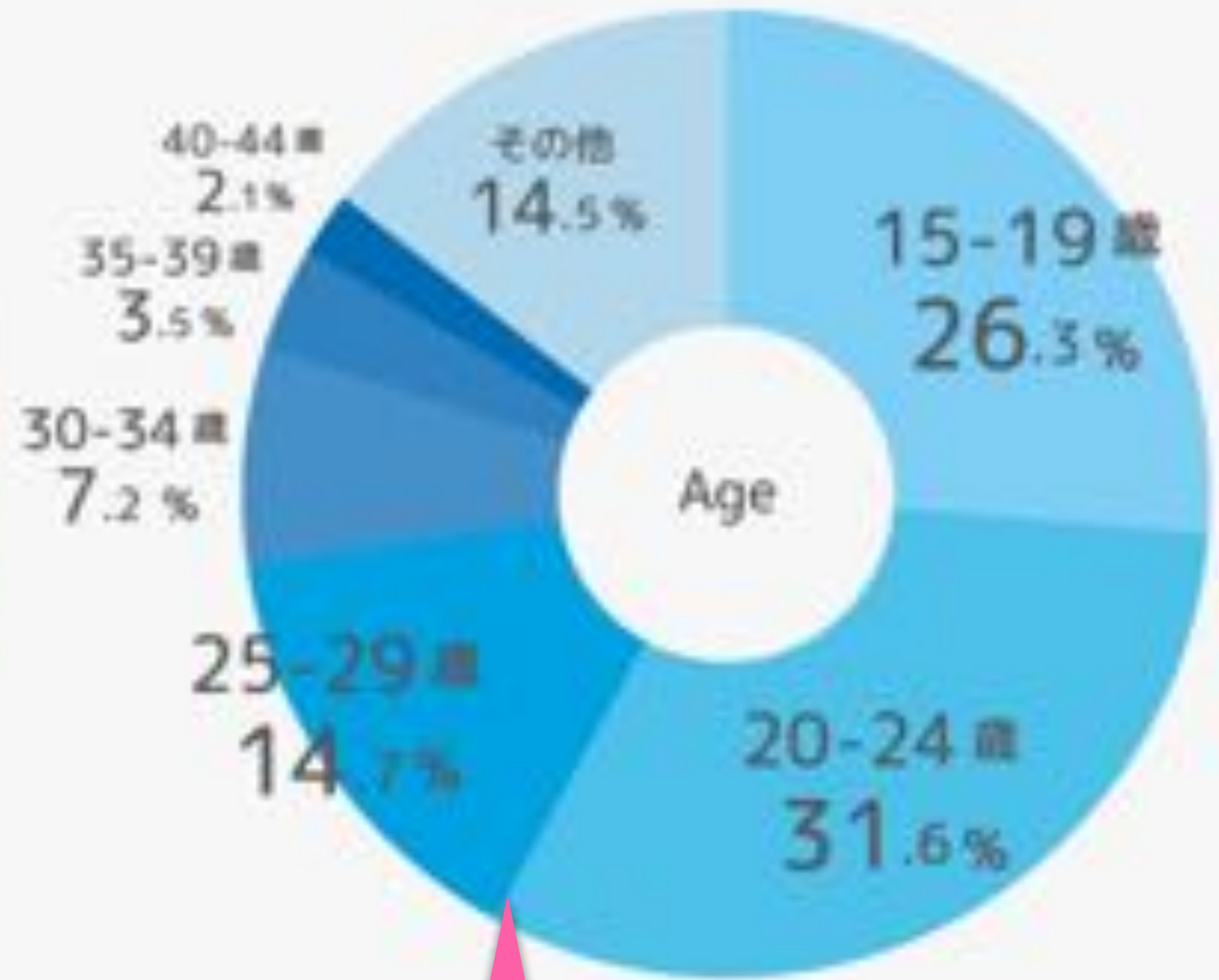


BrainWars

10 Million Users

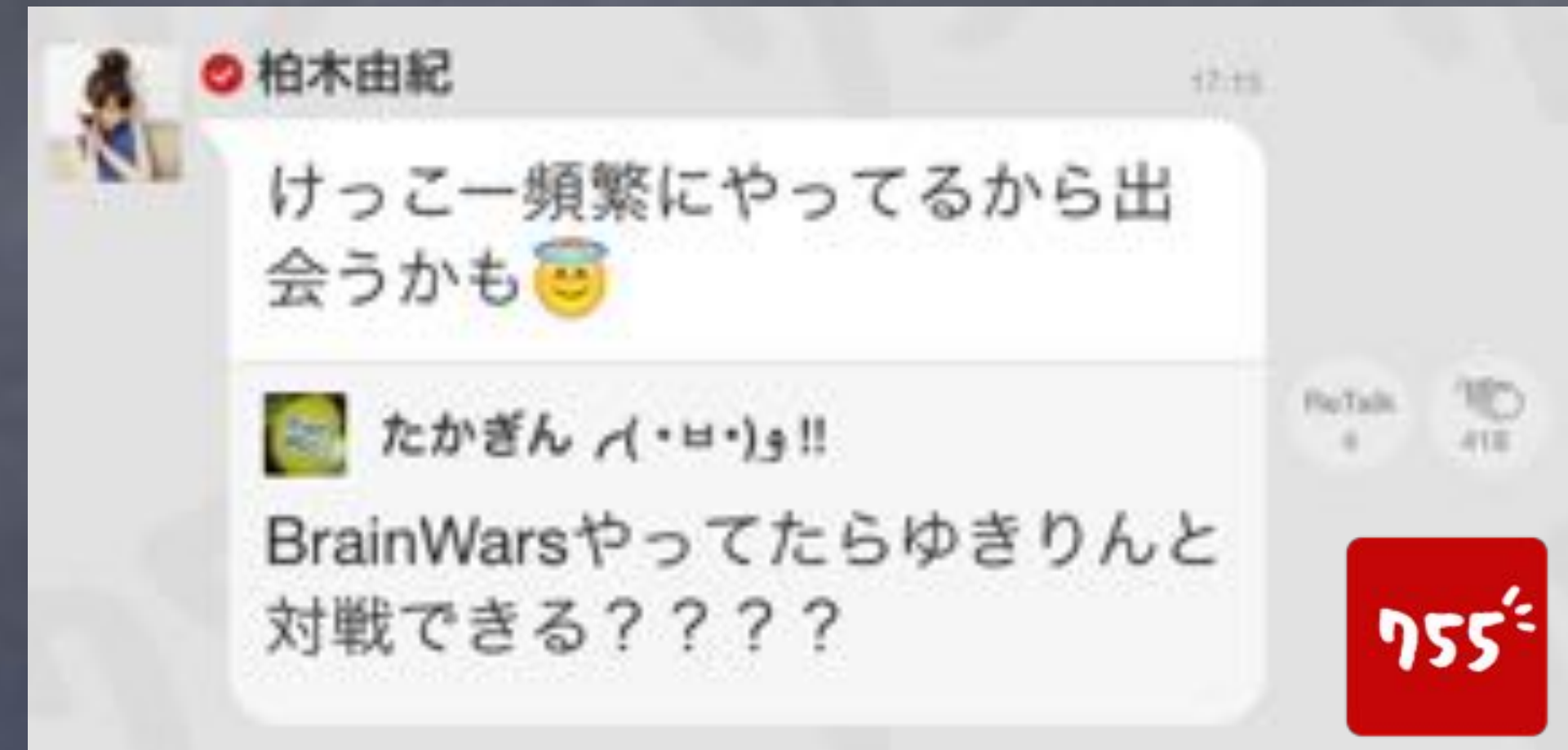


男女比ほぼ同じ



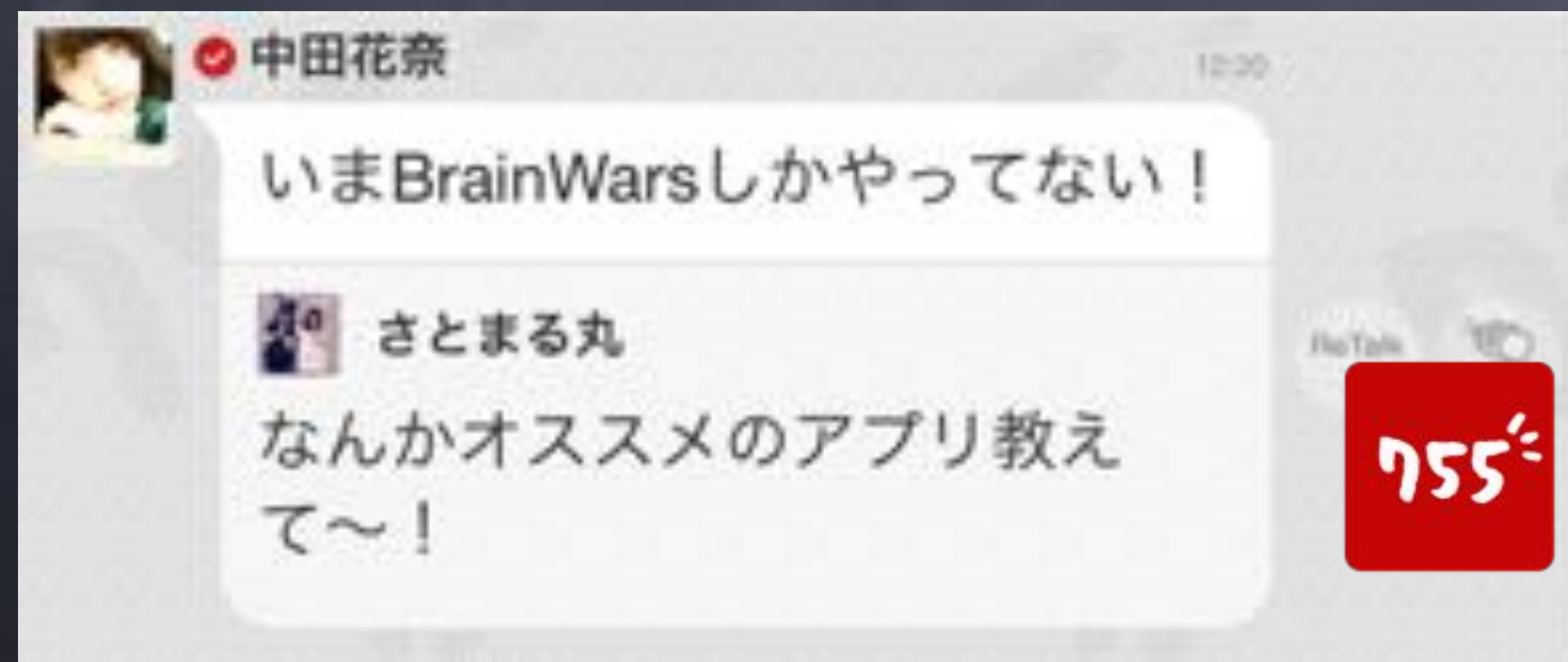
20代が約半数

# AKB48もハマる！



## 乃木坂46や

## 芸人まで



# 創業者たち

デザイナー  
花城氏

高場社長

工藤CTO

( ; °Д° )。oO ( ラーメン屋か... ? )



BrainWars概要

アーキテクチャ基本方針

BrainWarsとAWSサービスたち

まとめ

# 積極的に見直す

- スピード最優先！
- 事業的なインパクトがあるなら、既存構成にしばられずに積極的にアーキテクチャを見直す
- 面白そうなプロダクトやサービスはどんどん採用
- もちろんクリティカルなところは要検証で
- AWSはOregonやVirginia中心に（まだ東京リージョンが多い）
- | $\Delta^\circ$ ) < AWSにこだわらない

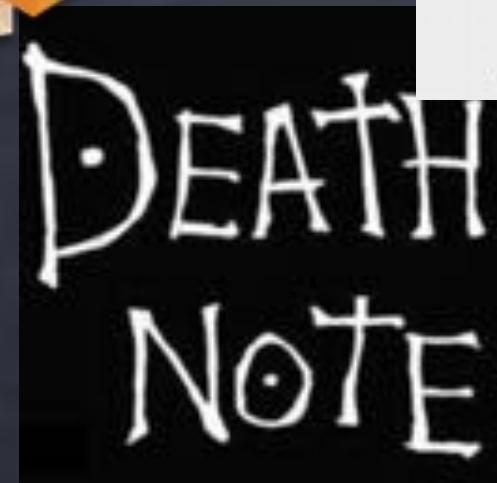
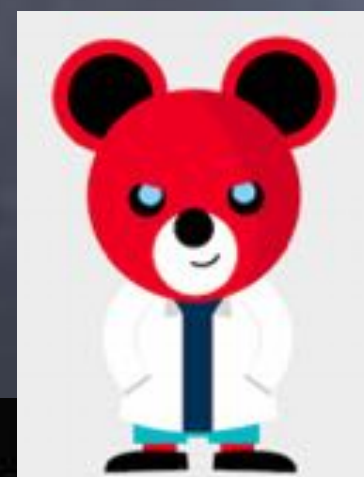
|ミサッ

# コストはシビアに

- 費用対効果が合わないプロダクトやサービスは使わない
  - プッシュ通知サービスとか
- 必要な機能だけどコストがかかるなら、自作も検討する
  - クラッシュレポーティングサービスとか
- 無駄なものは作らない
  - 管理画面などメンテが大変なもの
- AWSの中の人もびっくりするぐらい低コスト運用中！

# 運用を楽に楽しく

- 負荷や障害に強いアーキテクチャを意識する
  - 楽したいし、安眠したい
- hubotによるChatOpsベースの運用
  - みんなで楽しくDevOps！



BrainWars概要

アーキテクチャ基本方針

BrainWarsとAWSサービスたち

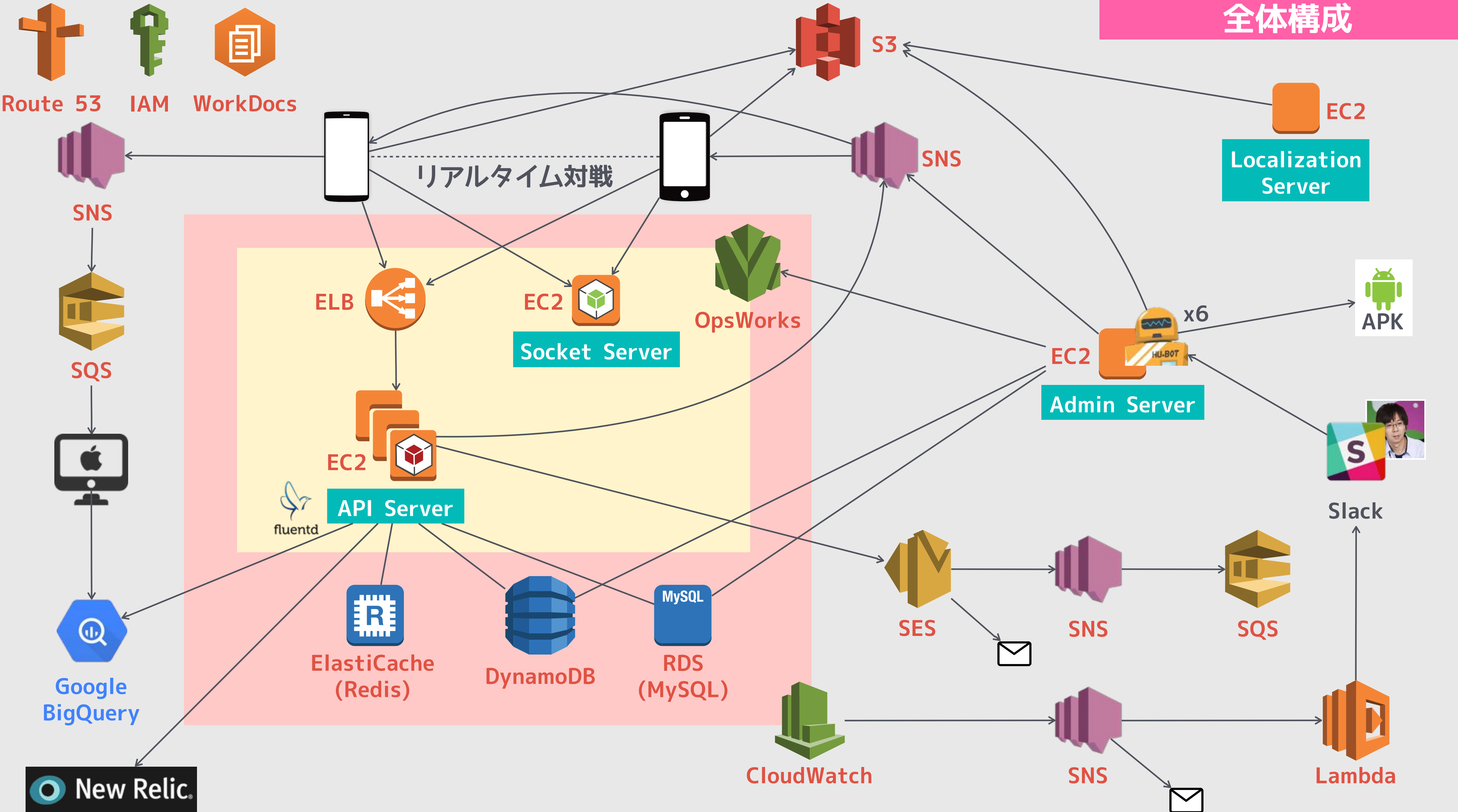
まとめ



BrainWars

全体アーキテクチャ

# 全体構成



# 利用中のサービスたち



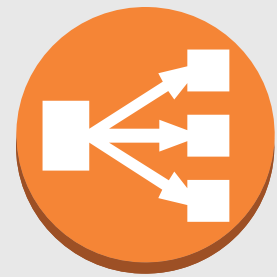
EC2



OpsWorks



SNS



ELB



CloudWatch



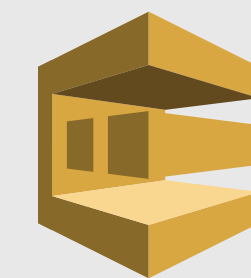
SES



Lambda



IAM



SQS



Route 53



RDS (MySQL)



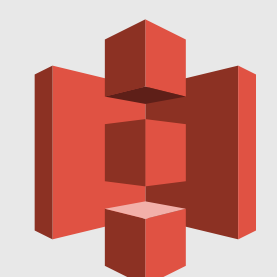
WorkDocs



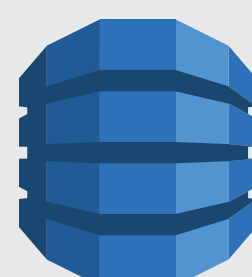
ElastiCache (Redis)



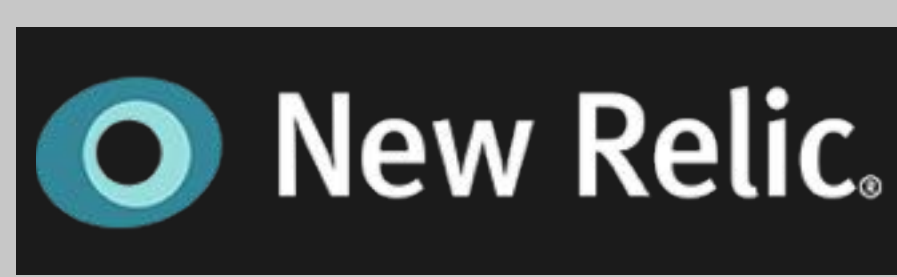
Google BigQuery



S3



DynamoDB





だしいぶ複雑？

分かりやすく  
機能別にご紹介

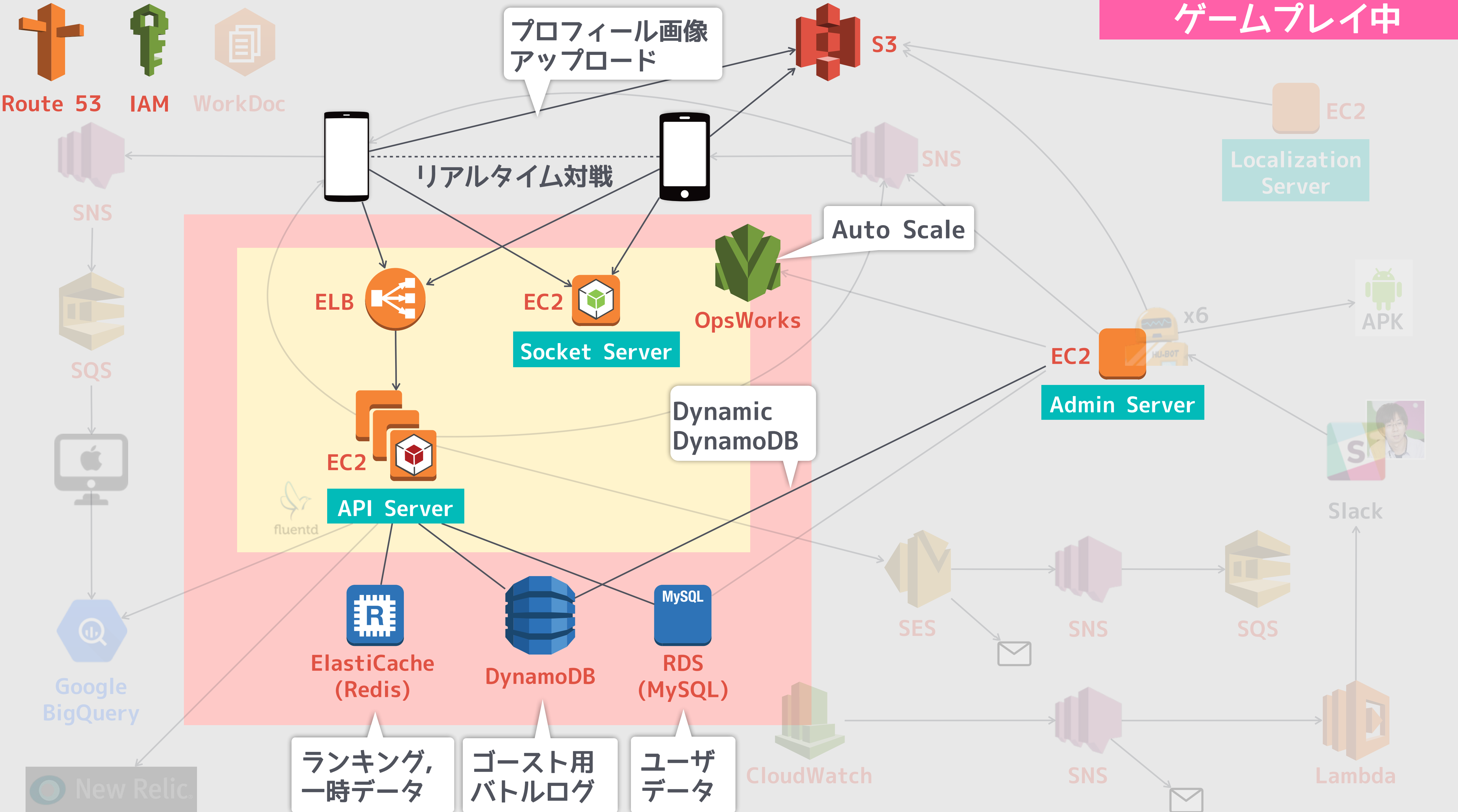


# BrainWars

ゲームプレイ中

のアーキテクチャ

ゲームプレイ中



ゲームプレイ中

プロフィール画像  
アップロード

リアルタイム対戦

インスタンス管理に  
OpsWorksを採用

Route 53 IAM WorkDoc

Auto Scale

Localization



Socket Server

Dynamic  
DynamoDB

Admin Server

API Server

ElastiCache  
(Redis)

DynamoDB

RDS  
(MySQL)

ランキング、  
一時データ

ゴースト用  
バトルログ

ユーザ  
データ

CloudWatch

SNS

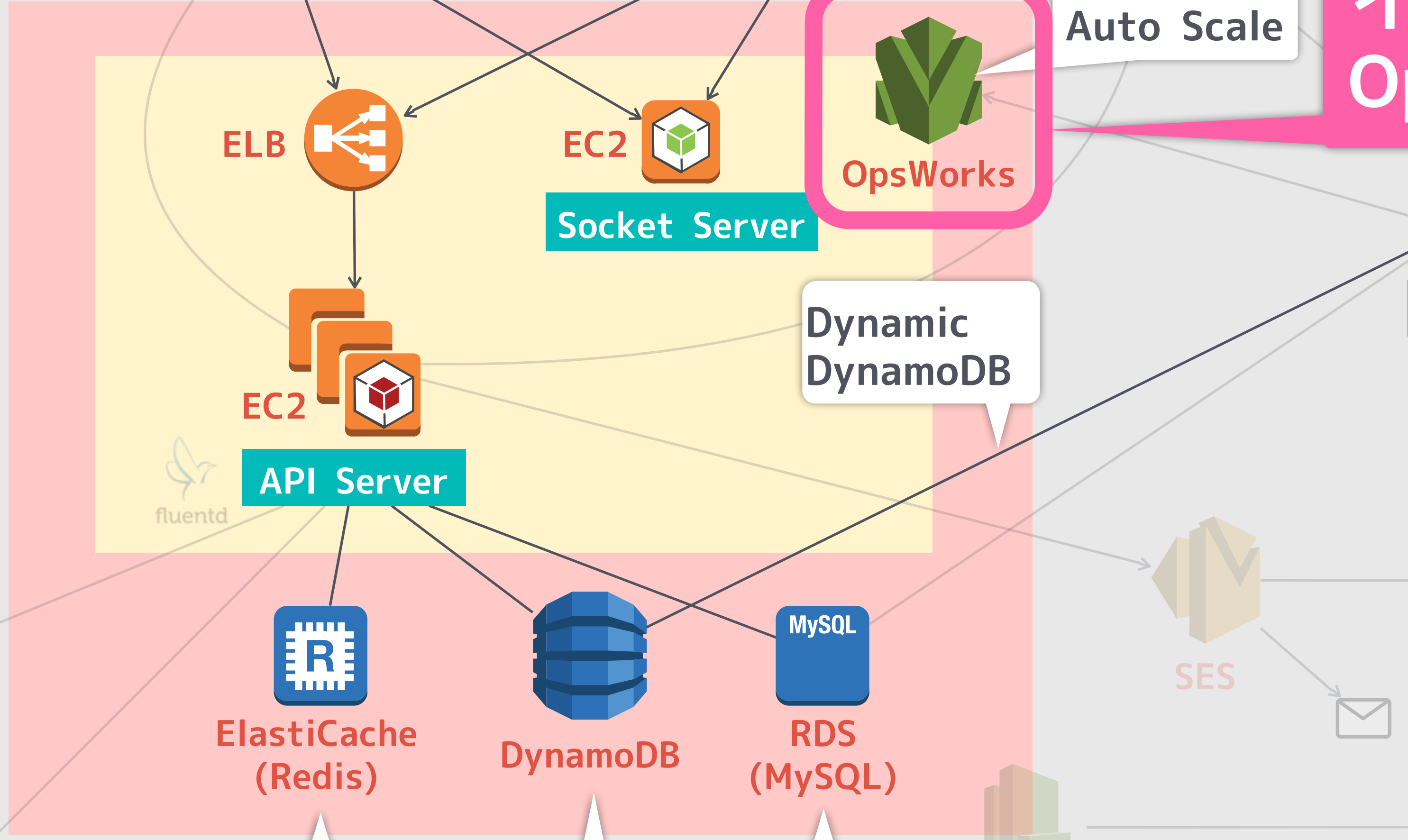
Lambda

New Relic

Google  
BigQuery

Slack

APK



# OpsWorksの活用

- OpsWorksとは
  - Chefを利用したアプリケーション管理サービス
  - 環境構築とアプリケーションのデプロイを自動化
  - 幅広いアーキテクチャをデフォルトでサポート
  - Chefのレシピ追加などでカスタマイズも可能
  - ELBと自動連携
  - time と load でオートスケール設定可能



# OpsWorksの活用

- BrainWarsで利用中のインスタンス
  - API Server (x2)
    - Amazon Linux, Ruby on Rails
    - time (ランキング確定直前) と load でオートスケール
    - ELB連携, EIP付与
    - ResourcesとしてRDSを設定
  - Socket Server
    - Amazon Linux, Socket.IO

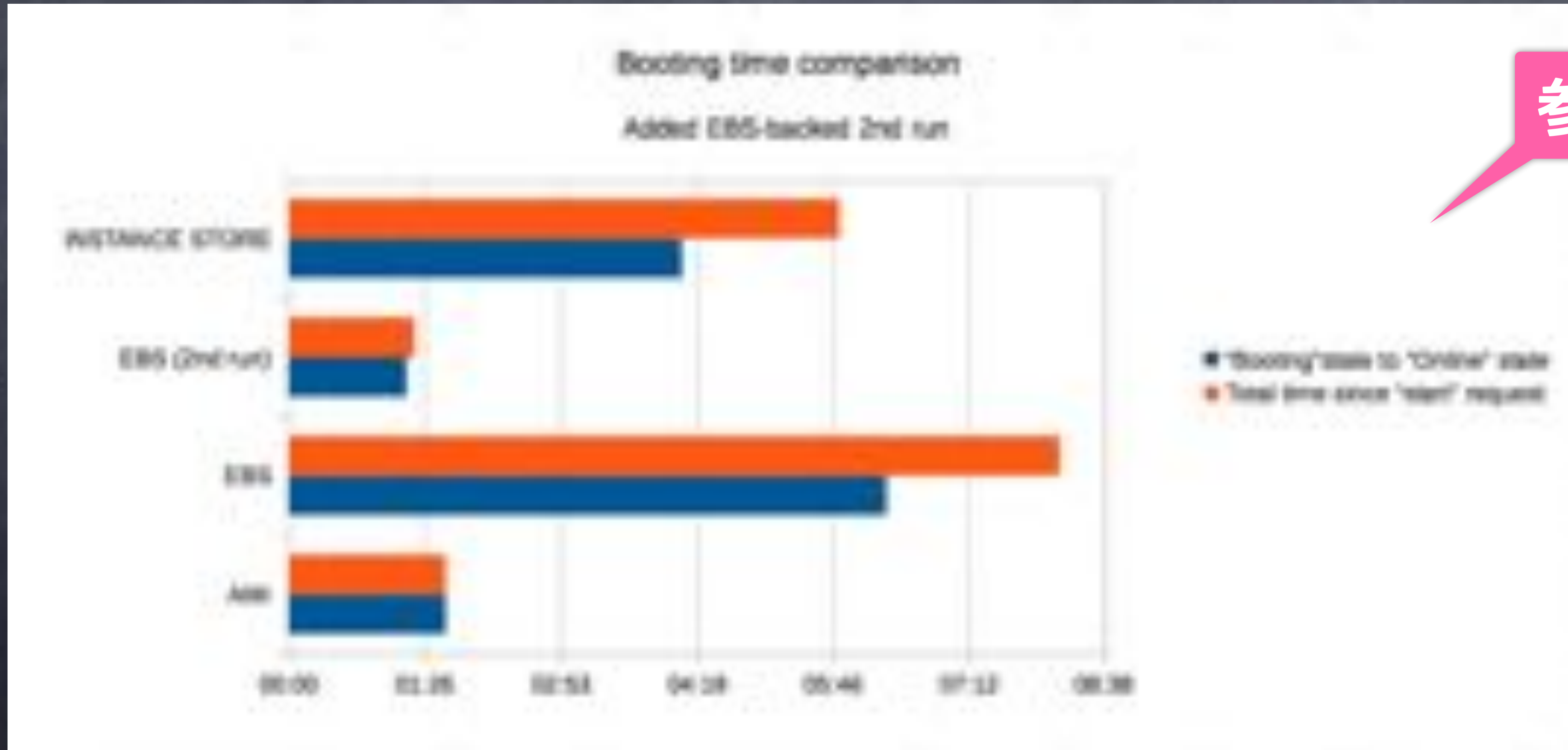
# OpsWorksの活用

- 工夫している点
  - Chefのレシピを追加
    - file descriptorなどのチューニング
    - monitプロセスの再起動（チューニング内容の反映）
    - CloudWatch連携（独自メトリクスの追加, アラーム登録）
    - td-agentなどのインストール



# OpsWorksの活用

- カスタムAMI, EBS-Bootで起動速度向上 (7分強 → 2分弱)



参考データ

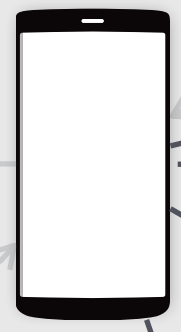
ゲームプレイ中

Route 53 IAM WorkDoc

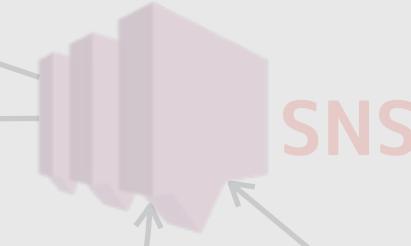
プロフィール画像  
アップロード



S3



リアルタイム対戦



SNS



EC2

Localization  
Server

Auto Scale

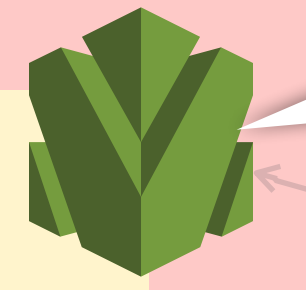


ELB



EC2

Socket Server



OpsWorks



EC2

Admin Server



APK

Dynamic  
DynamoDB

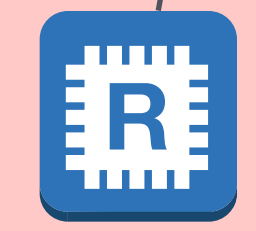


EC2

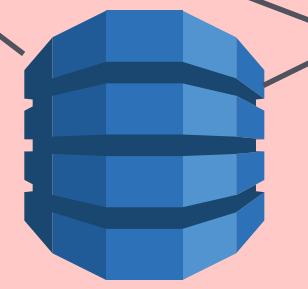
API Server



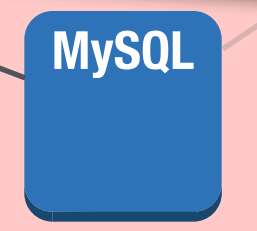
fluentd



ElastiCache  
(Redis)

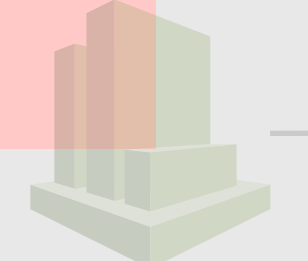


DynamoDB

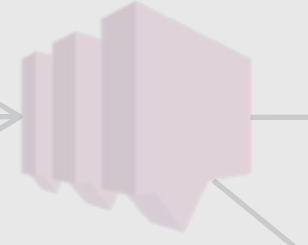


RDS  
(MySQL)

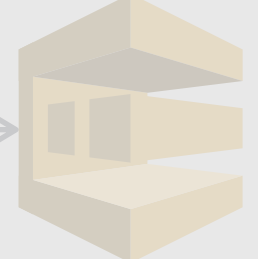
バトルログのみ  
DynamoDBを採用



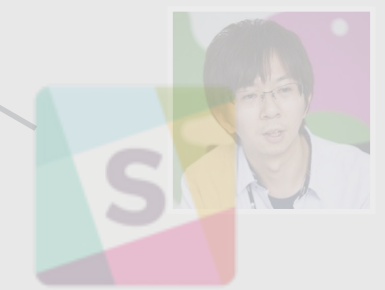
CloudWatch



SNS



SQS



Slack



Lambda

ランキング、  
一時データ

ゴースト用  
バトルログ

ユーザ  
データ

New Relic

SNS

SQS

Google  
BigQuery

# DynamoDBの活用

- DynamoDBとは
  - 高い信頼性, スケーラビリティ, 低レイテンシ,  
安定した性能を備えたフルマネージド型NoSQLデータベース
  - IOPSをいつでも変更できる
  - スキーマレスなデータ構造

# DynamoDBの活用

- バトルログテーブル
  - ユーザがバトルしたときの操作ログ
  - PK は Hash(ユーザID) & Range(トレーニングID)
  - データ量は93GB、アイテム数は1.1億
  - 通常はWriteのみ、Readは後述するケースのみ
- Dynamic DynamoDBでオートスケール

# DynamoDBの活用

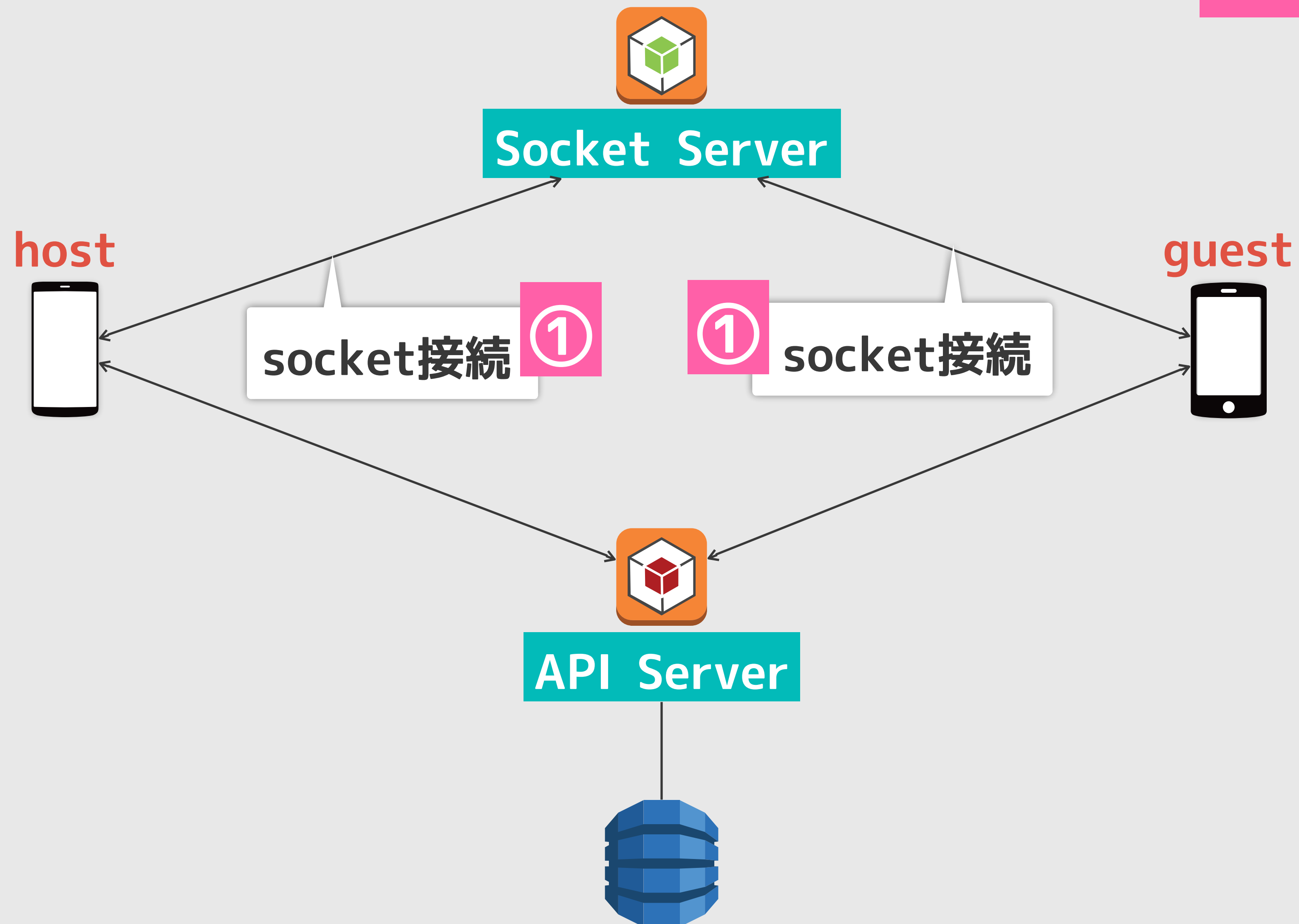
通常対戦の流れ

対戦するぜ！！！！

(´・ω・´)シャキーン

# DynamoDBの活用

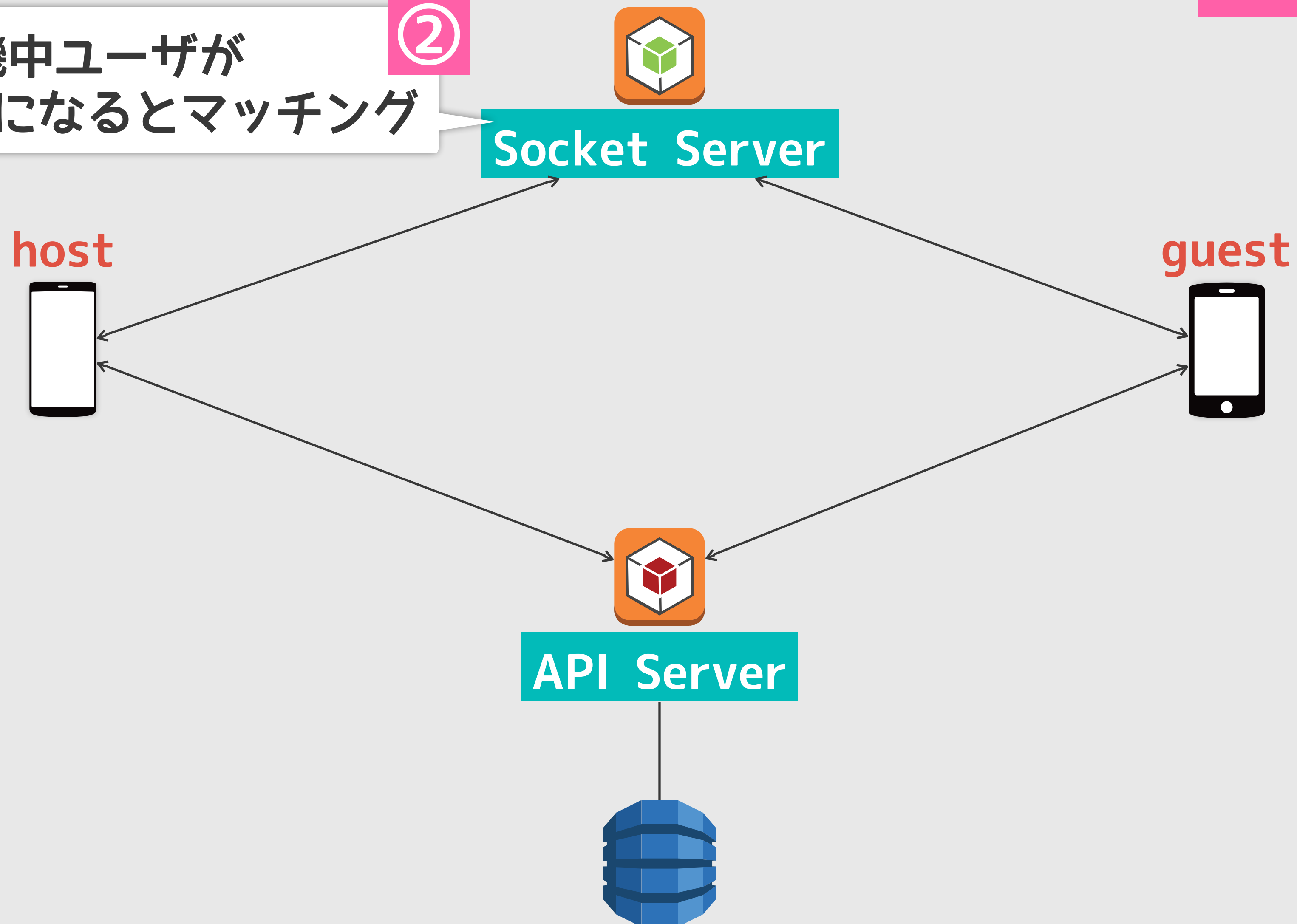
通常対戦の流れ



# DynamoDBの活用

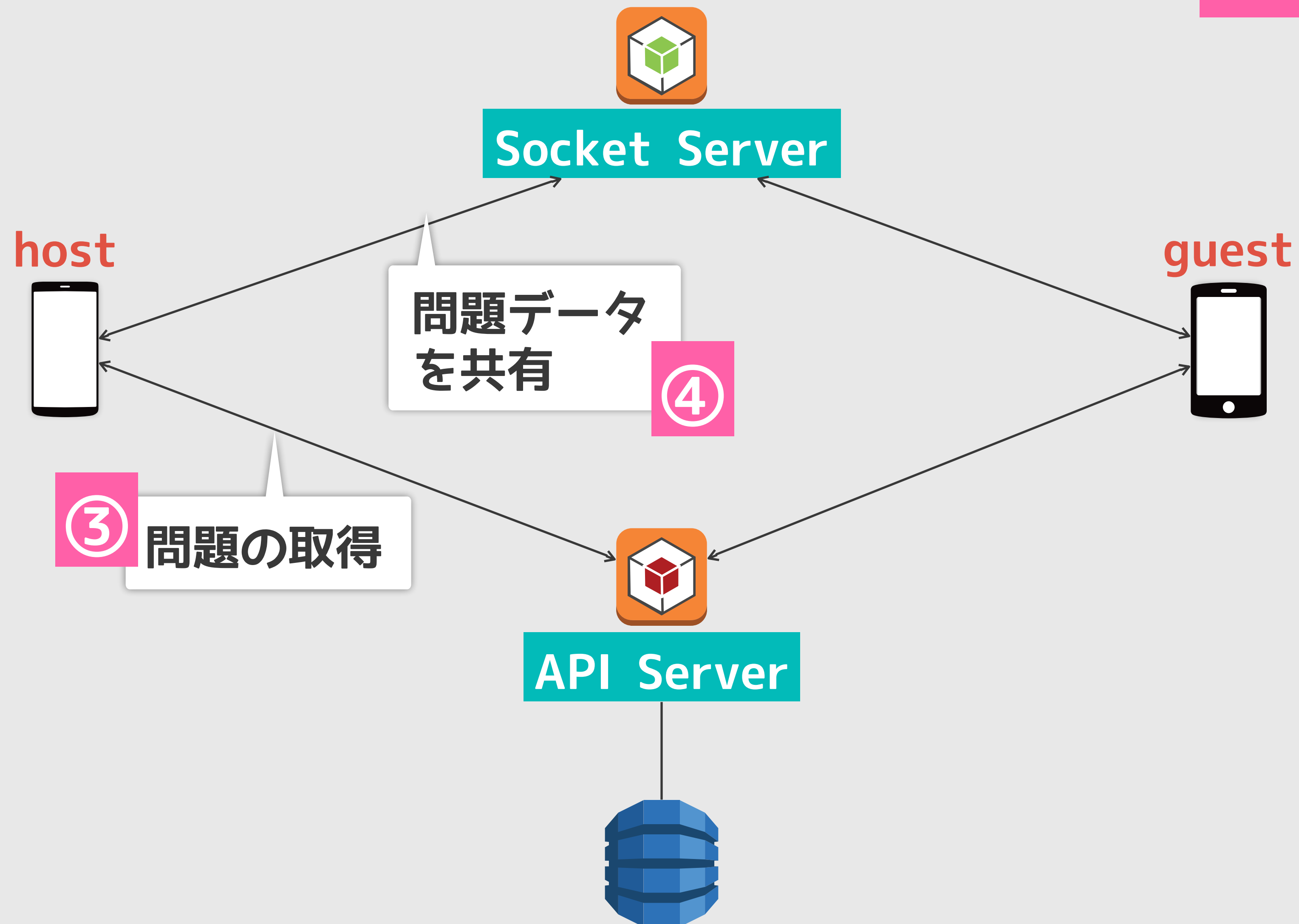
通常対戦の流れ

②  
待機中ユーザが  
2人になるとマッチング



# DynamoDBの活用

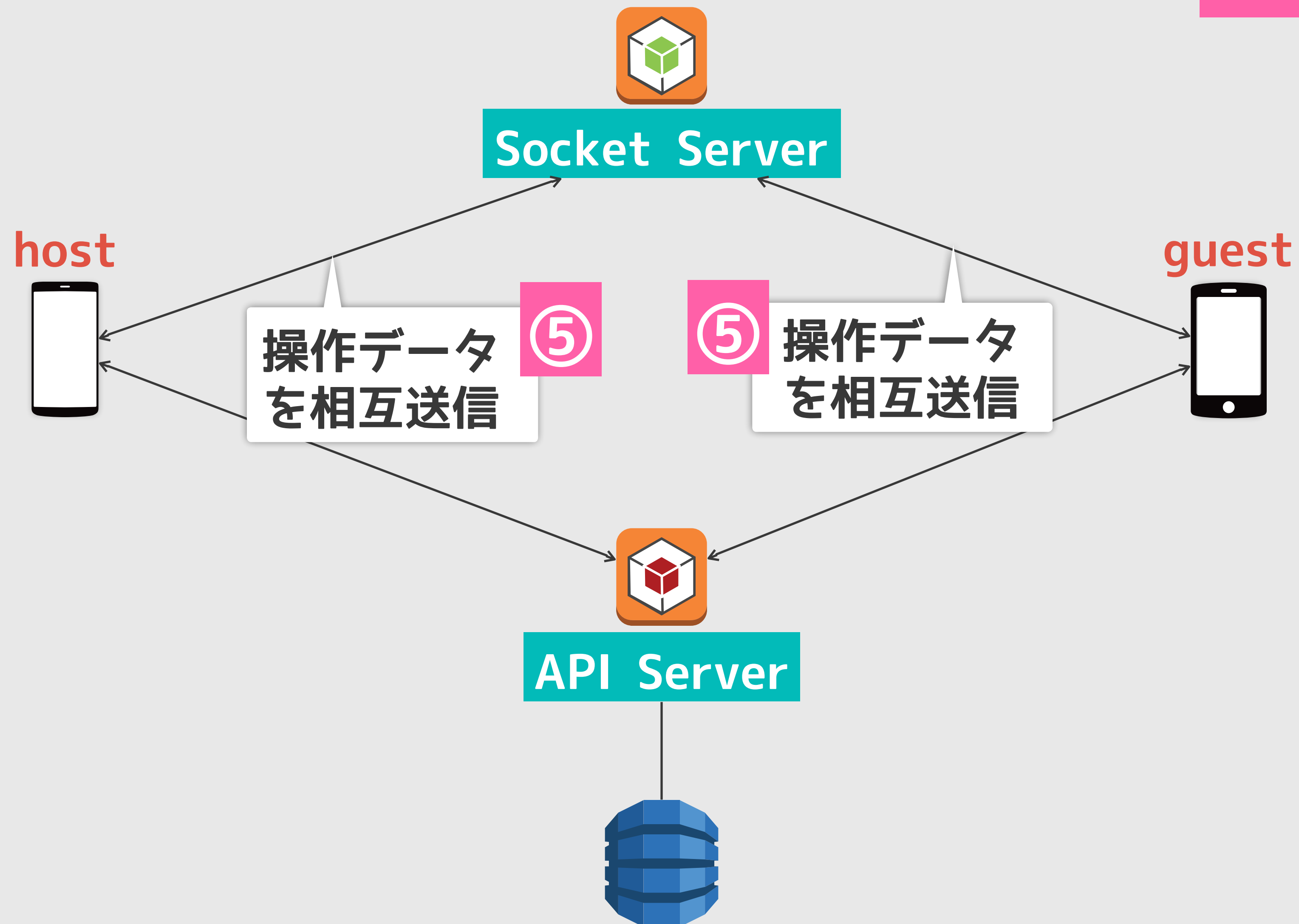
通常対戦の流れ





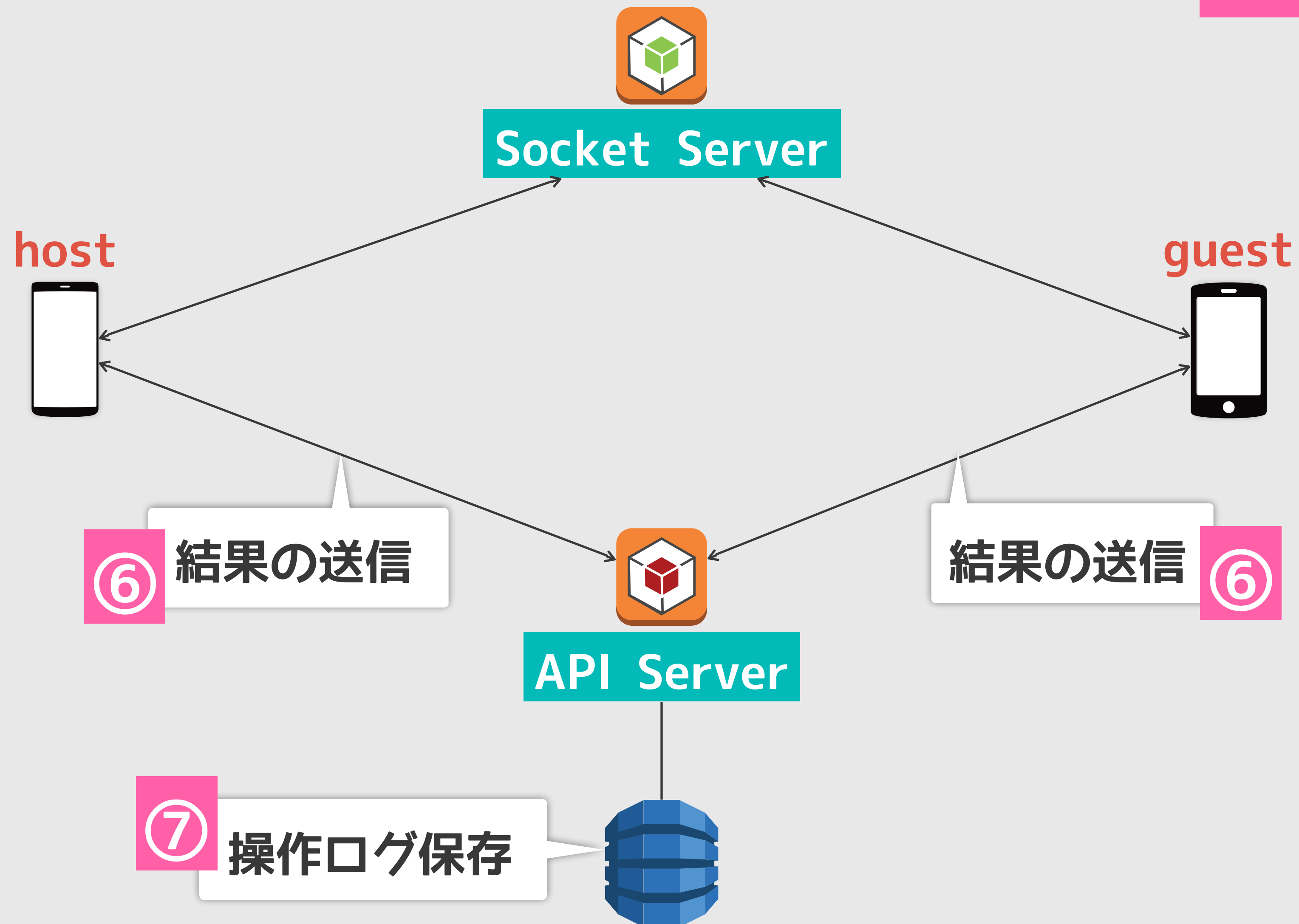
# DynamoDBの活用

通常対戦の流れ



# DynamoDBの活用

通常対戦の流れ



# DynamoDBの活用

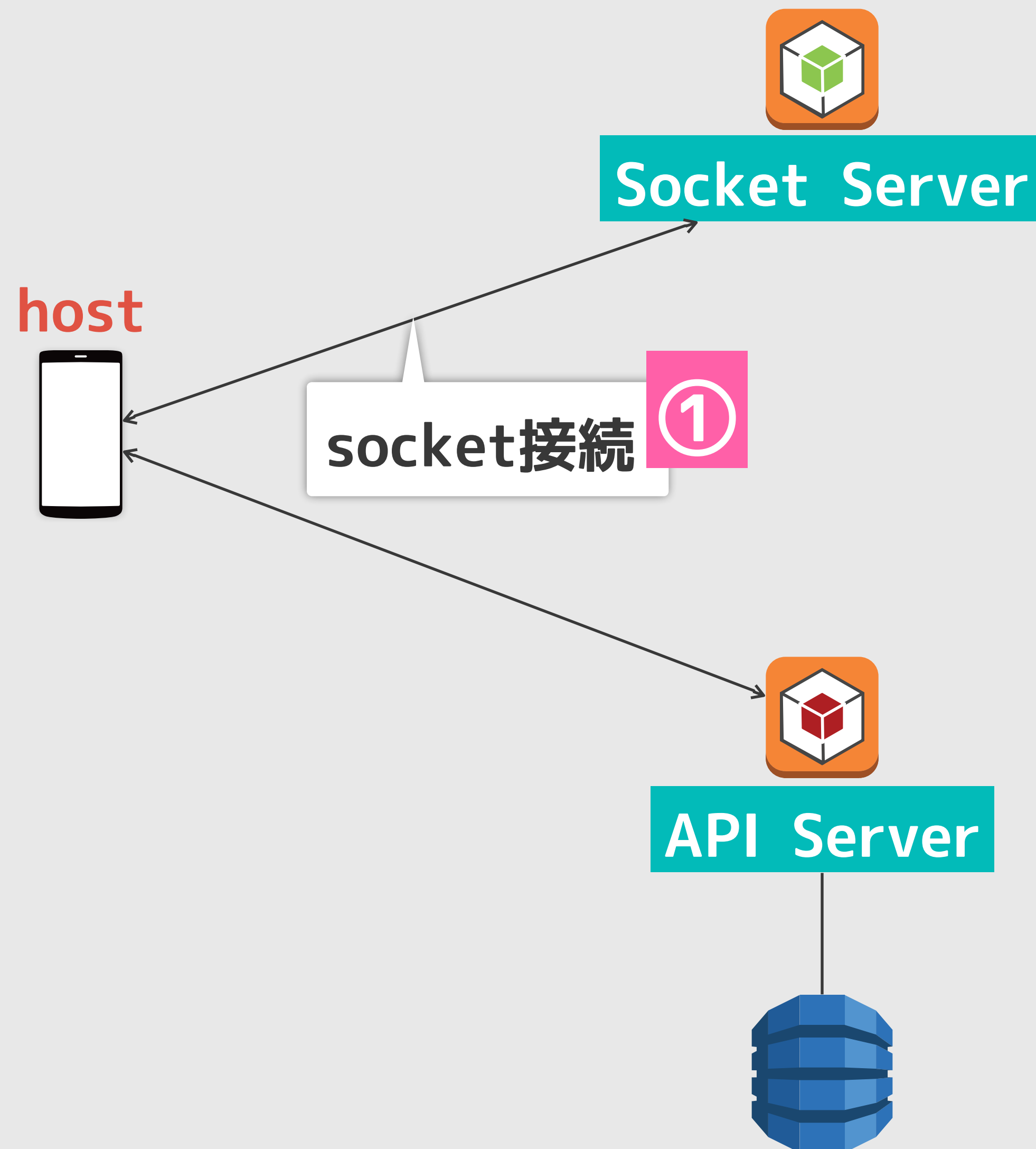
マッチング失敗の流れ

相手がしななかったら  
どうしよう？

(((:°Д°)))

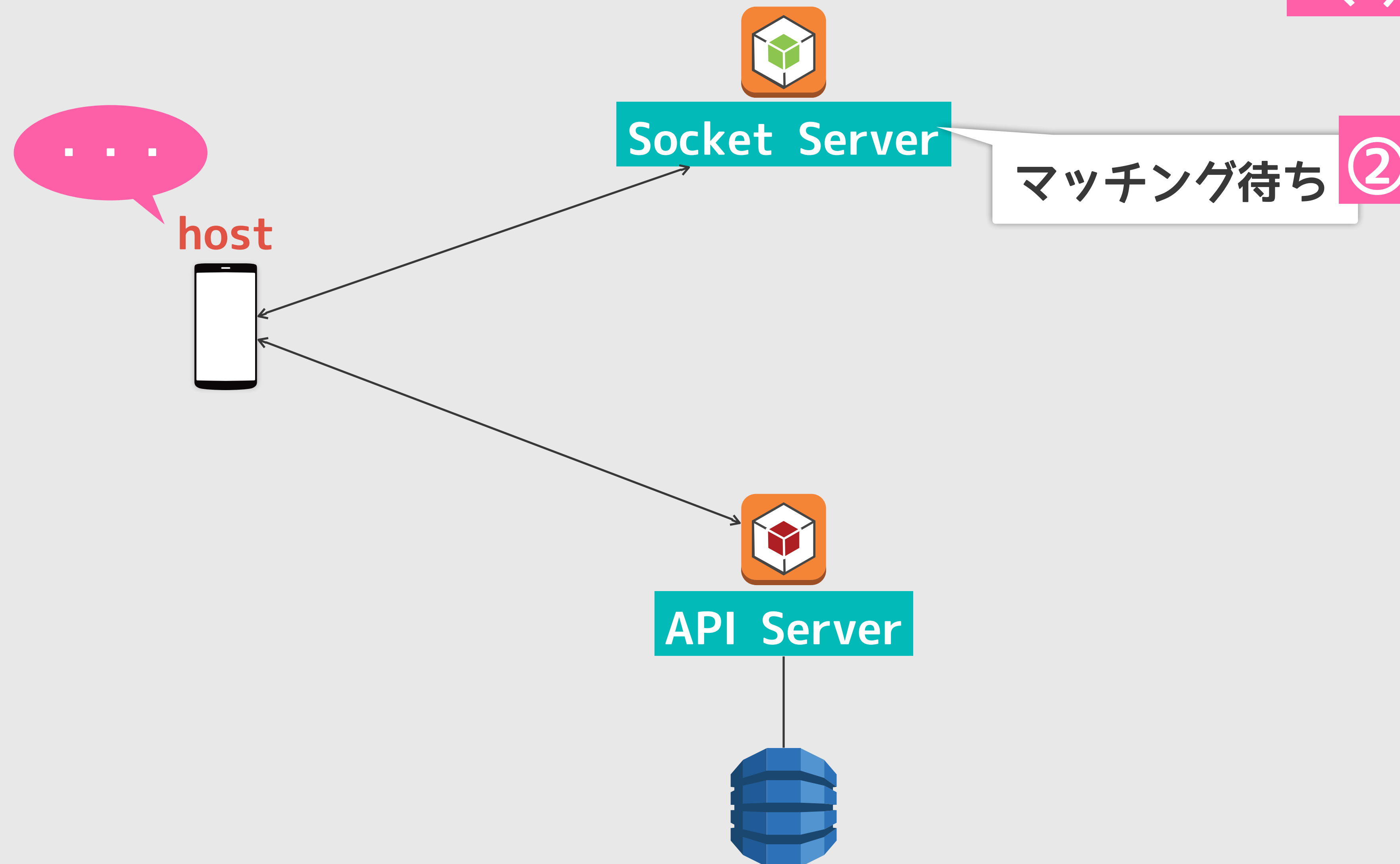
# DynamoDBの活用

マッチング失敗の流れ



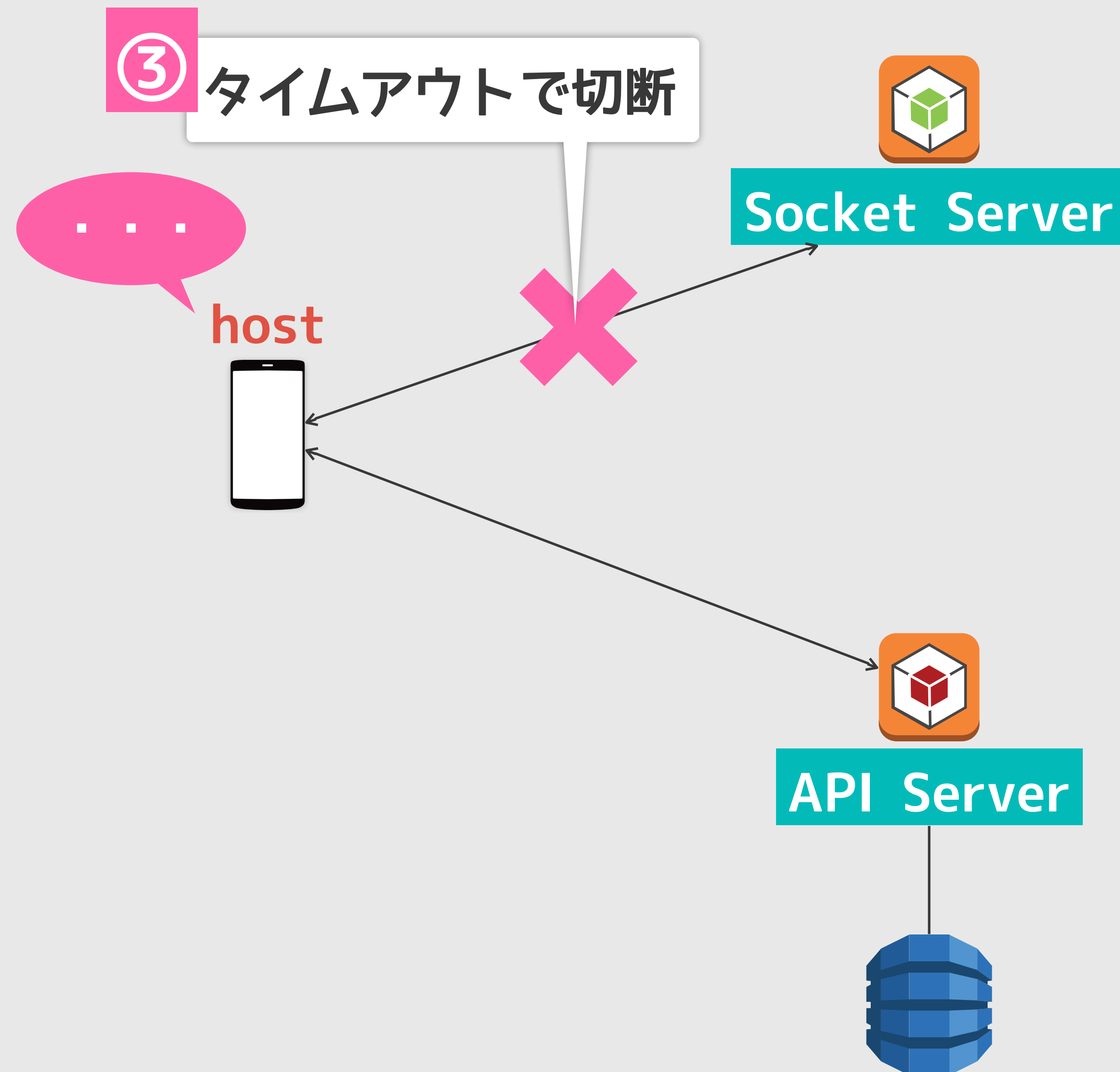
# DynamoDBの活用

マッチング失敗の流れ



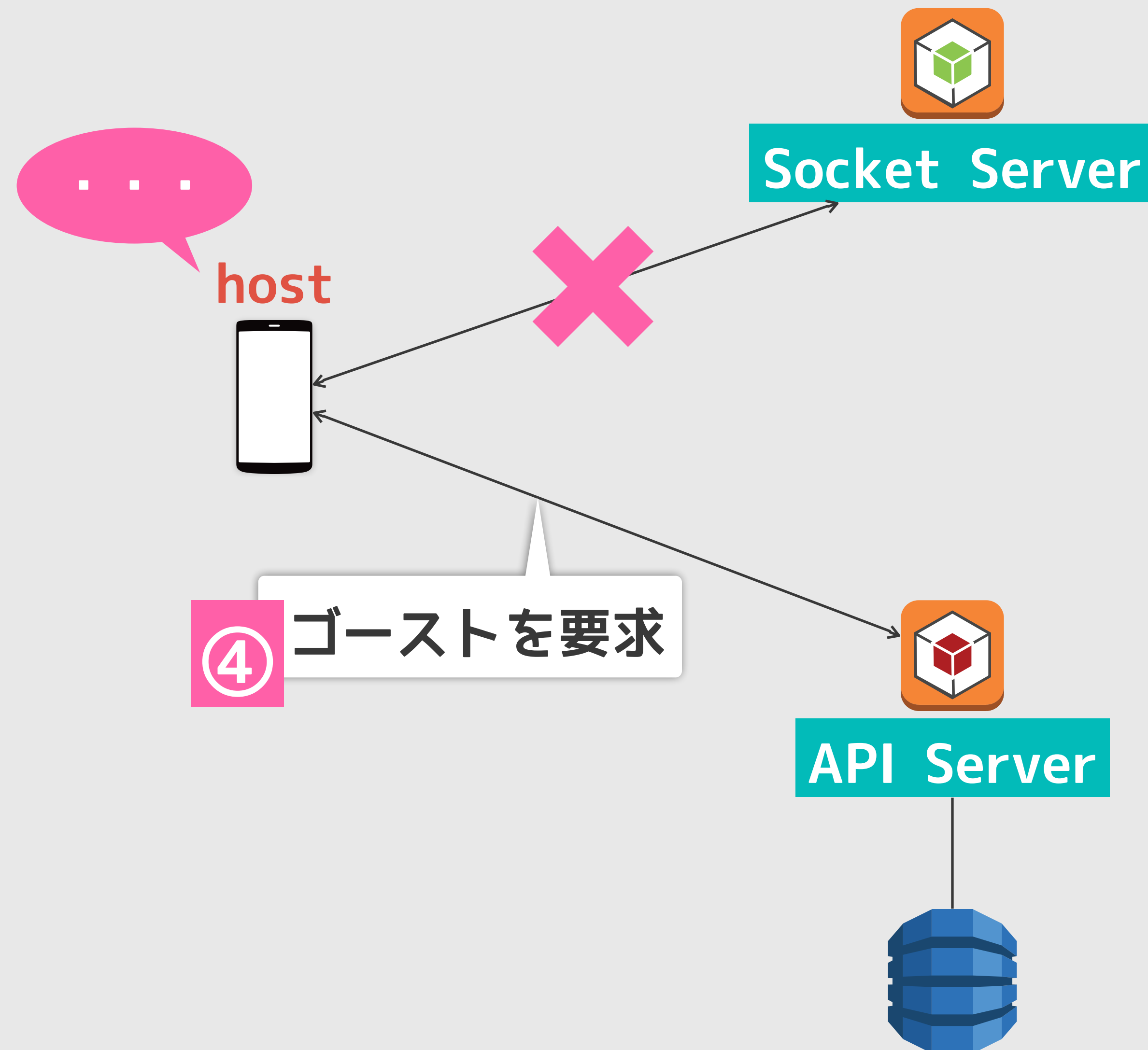
# DynamoDBの活用

マッチング失敗の流れ



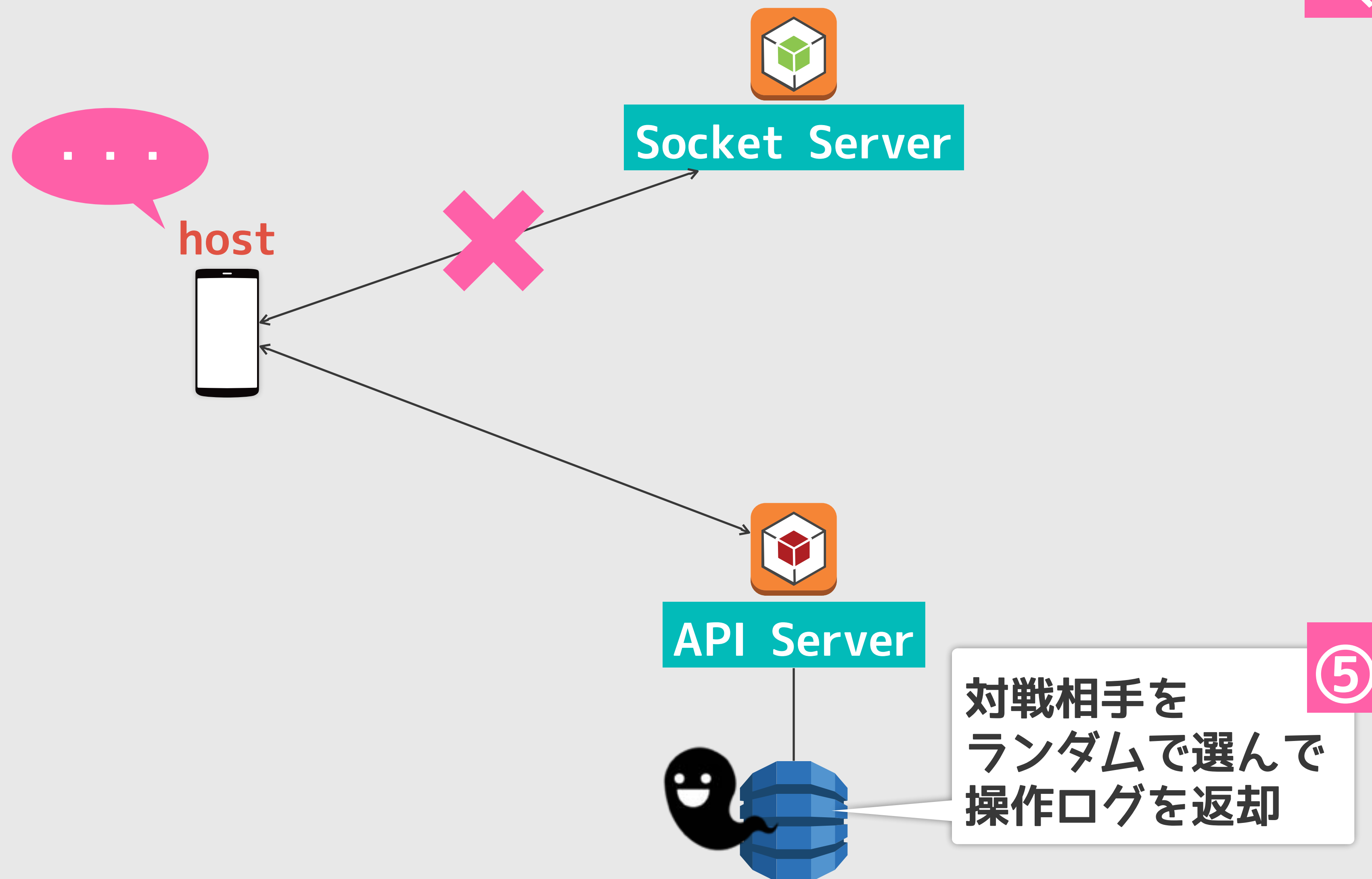
# DynamoDBの活用

マッチング失敗の流れ



# DynamoDBの活用

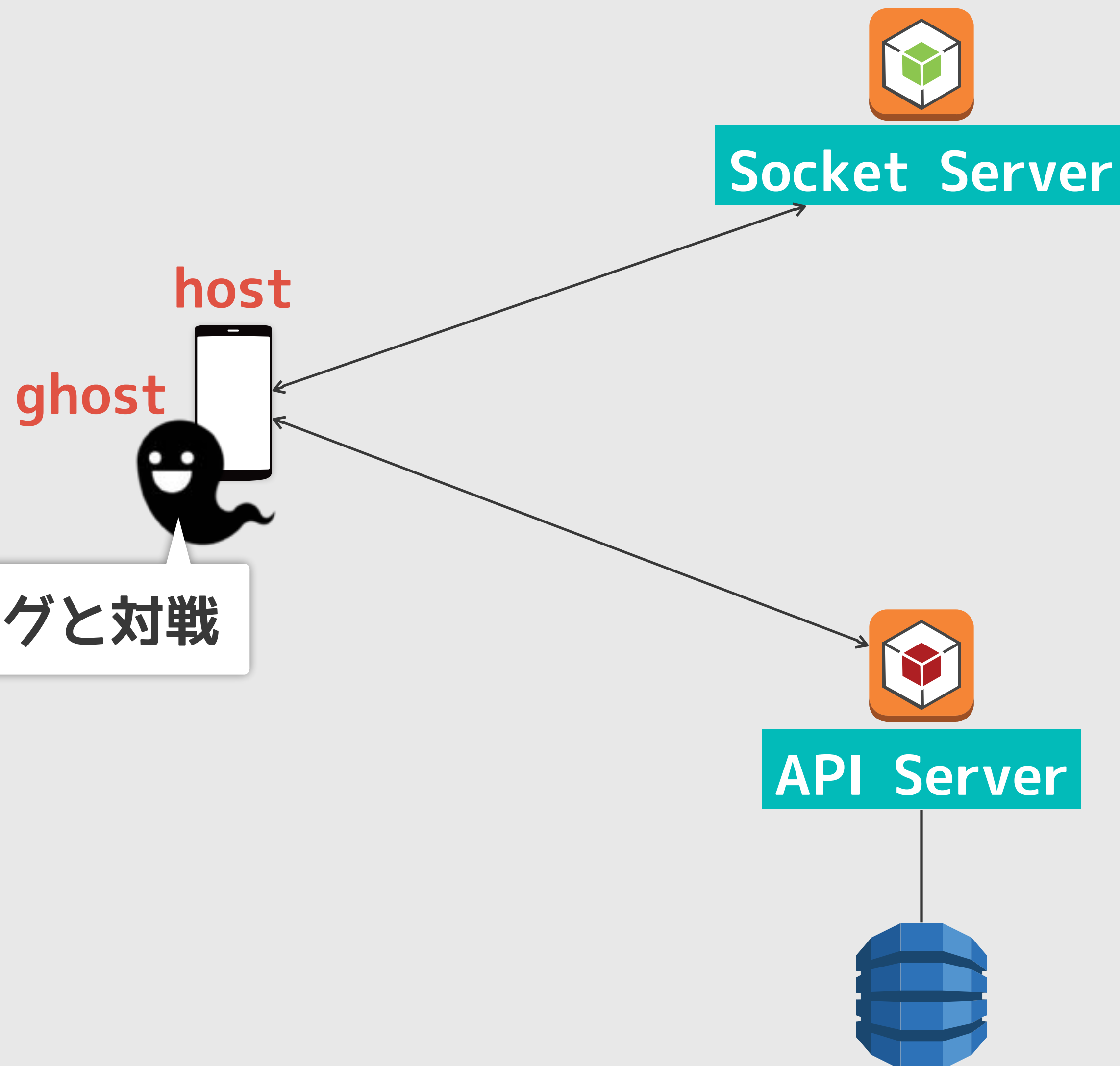
マッチング失敗の流れ





# DynamoDBの活用

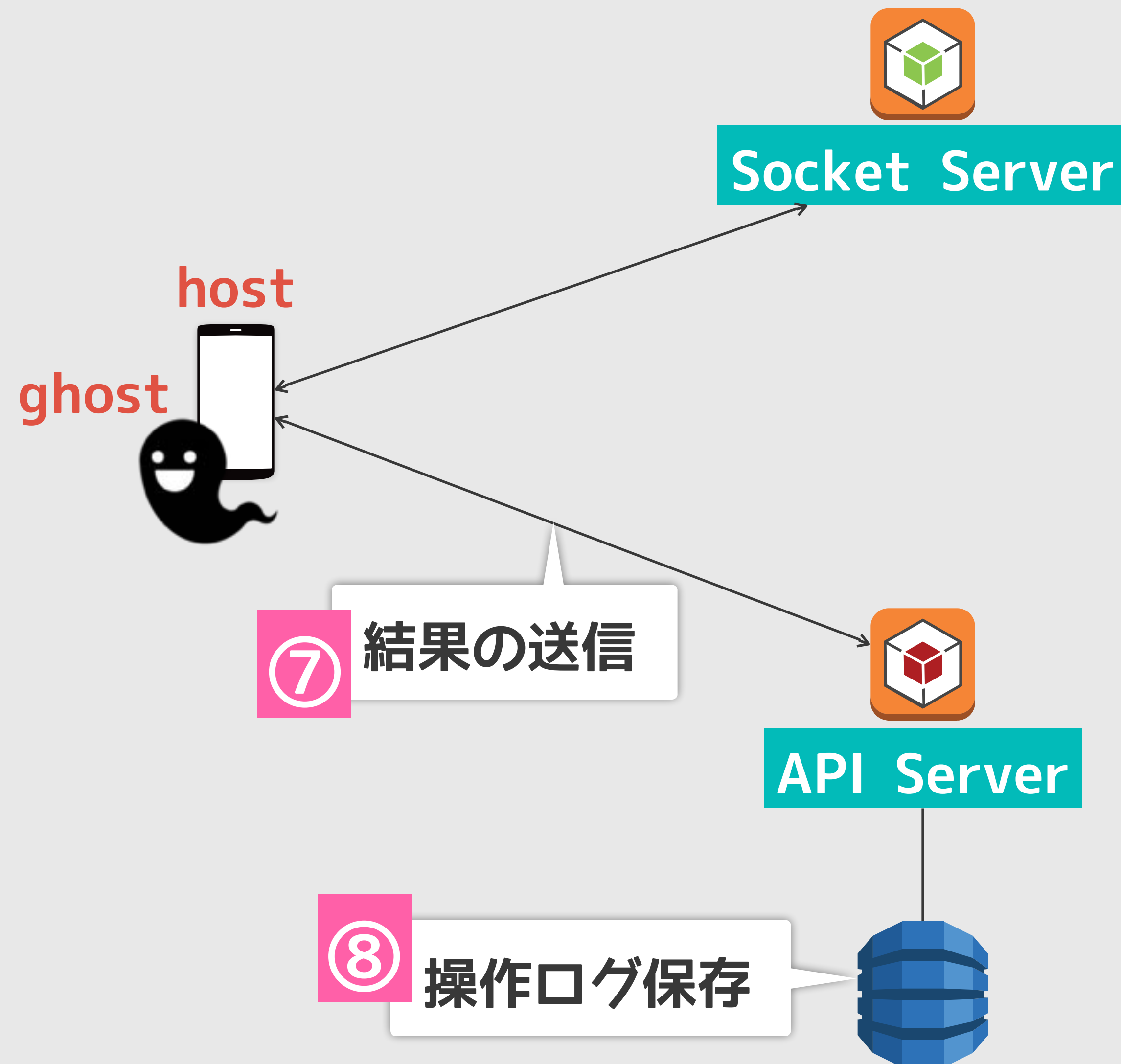
マッチング失敗の流れ



⑥ 操作ログと対戦

# DynamoDBの活用

マッチング失敗の流れ



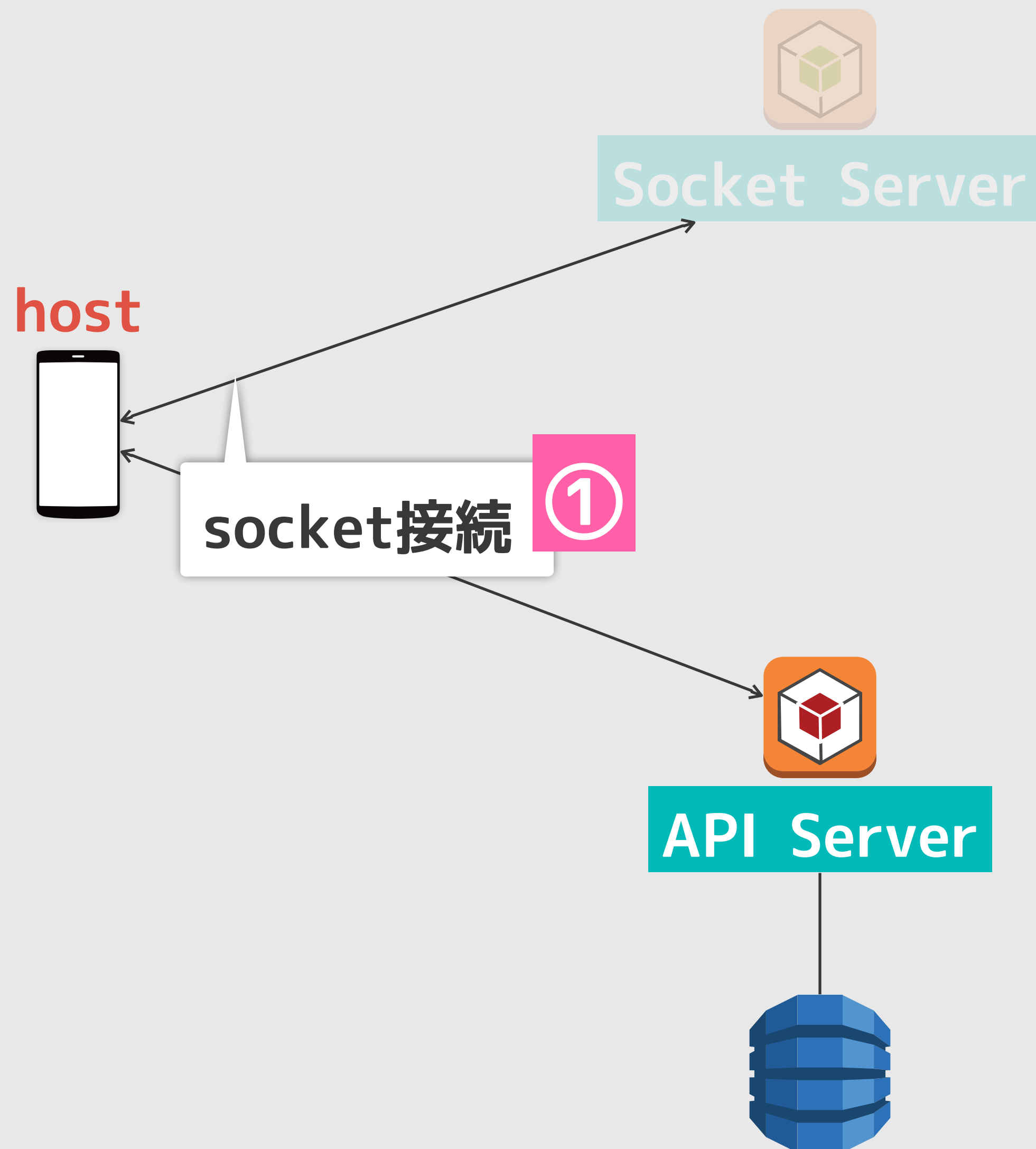
# DynamoDBの活用

socket停止の流れ

socketサーバが落ちたら  
どうしよう？  
((( ; °Д° )))

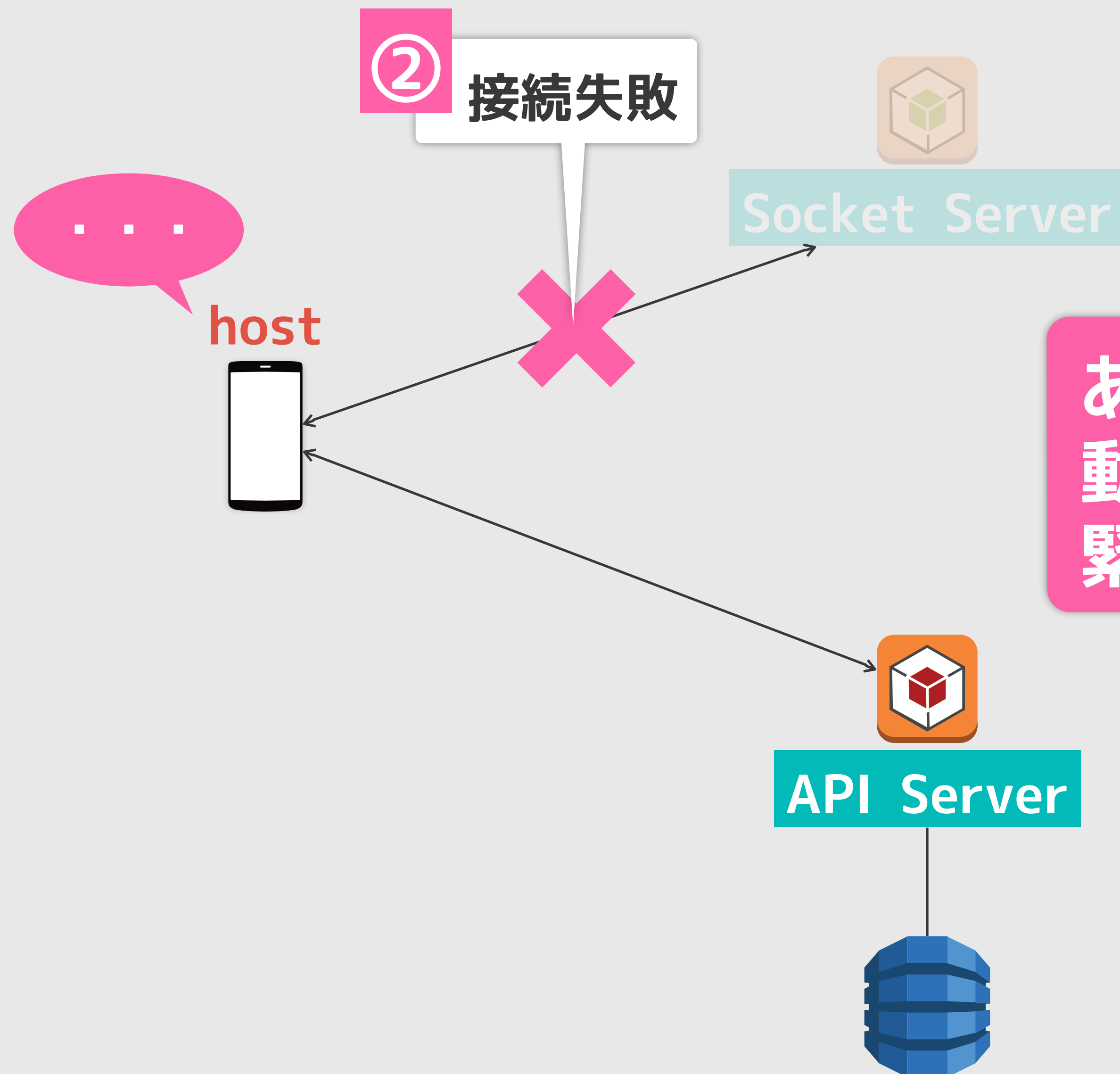
# DynamoDBの活用

socket停止の流れ



# DynamoDBの活用

socket停止の流れ



あとはマッチング失敗時と  
動きは一緒！  
緊急時の動作にも対応してた！

# DynamoDBの活用

socket停止の流れ

ちなみにスタンプは？  
相手は無反応になる？

# DynamoDBの活用

socket停止の流れ

ロジックで  
人間っぽく反応  
(Φ ω Φ)フフフ...

# DynamoDBの活用

socket停止の流れ



弊社デザイナー

「いやあ、♥送ったらいい♥きたわ」  
「オレ、モテるかもしんねー(\*ノω・\*)テハ」



弊社代表

「昨日はsocketサーバ落ちてたわ…」  
「〇〇さん、それゴーストですよ」



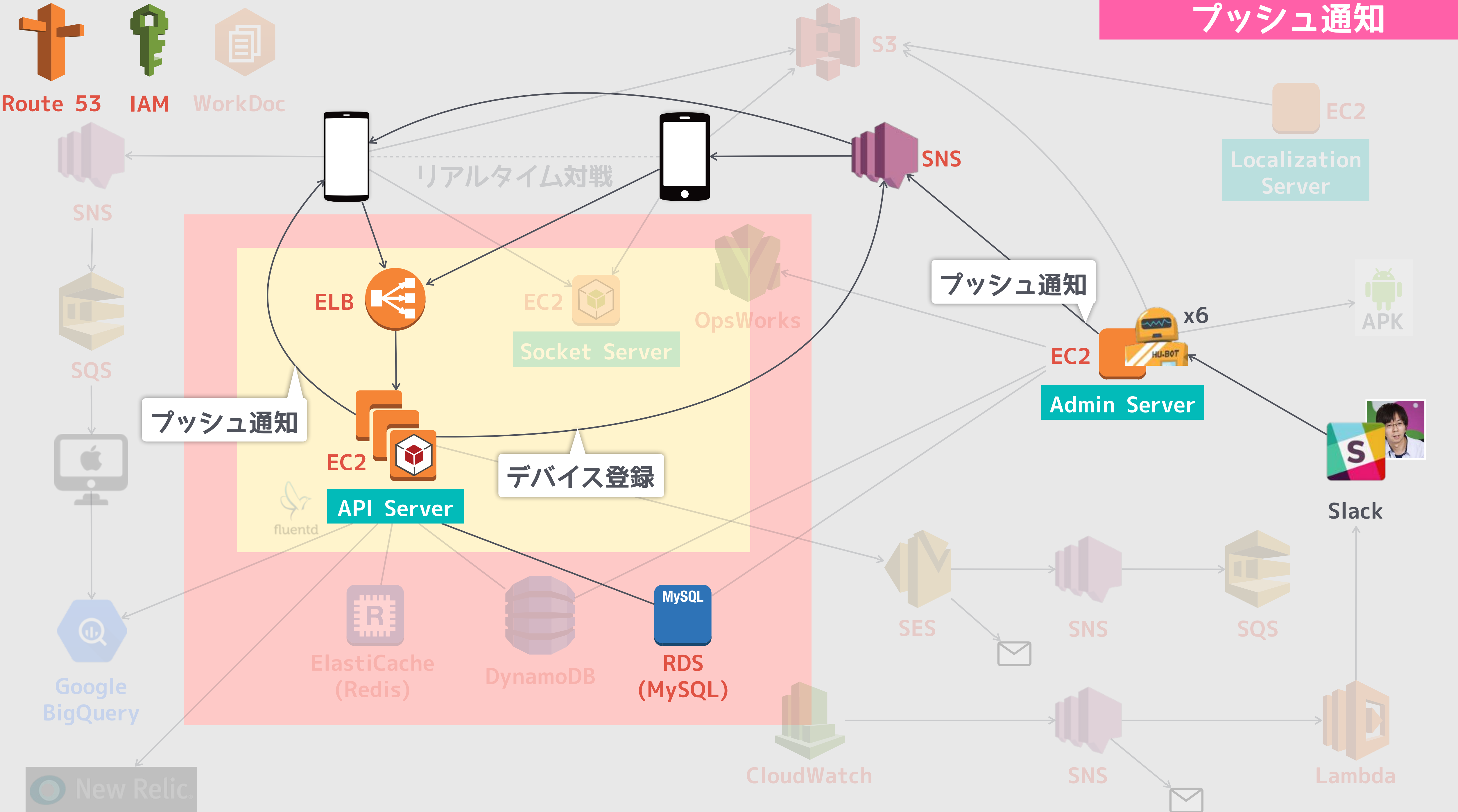


# BrainWars

プッシュ通知

のアーキテクチャ

# プッシュ通知



プッシュ通知

SNSから一括で  
プッシュ通知

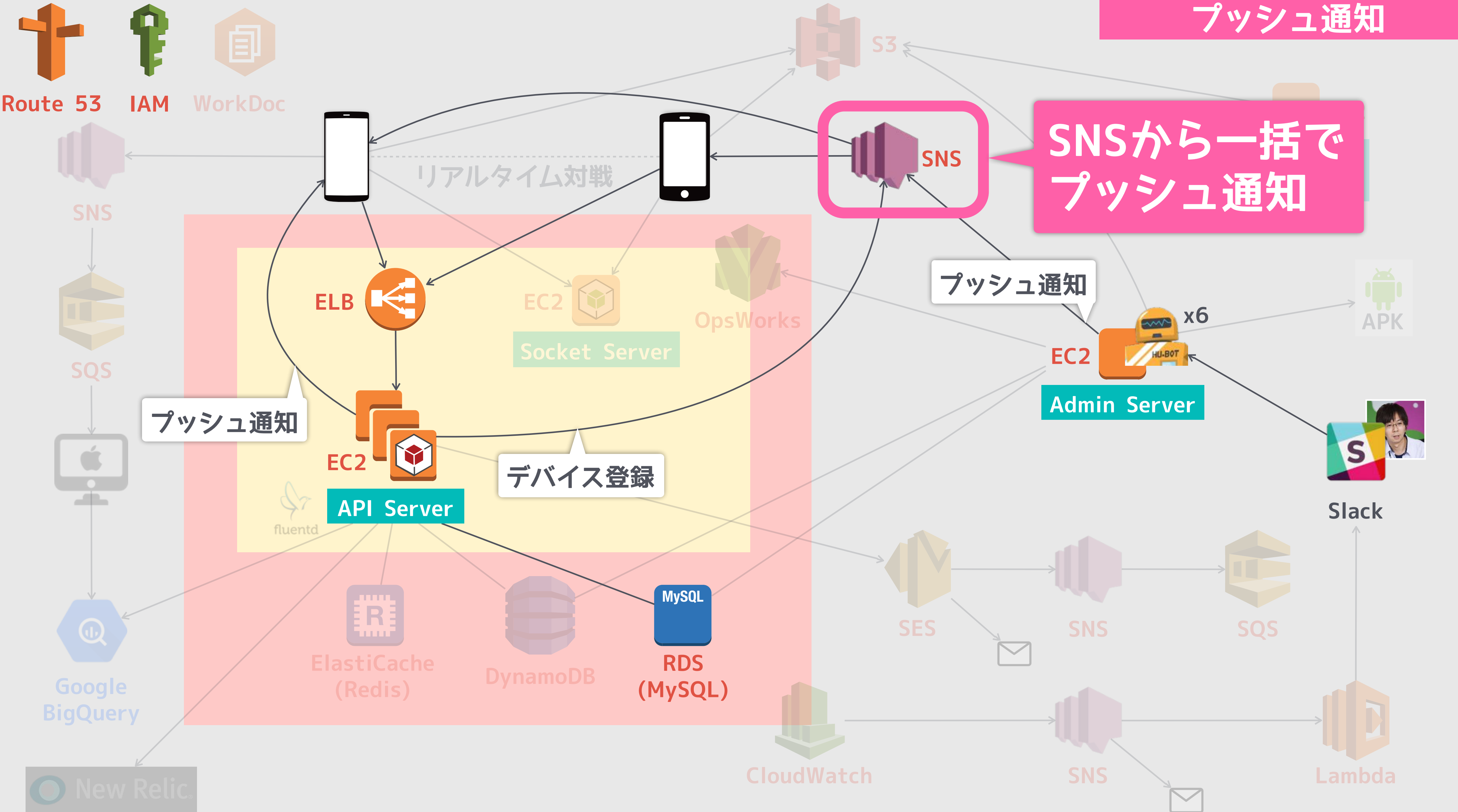
プッシュ通知

Admin Server

デバイス登録

プッシュ通知

リアルタイム対戦



Route 53 IAM WorkDoc

SNS

SQS

Google BigQuery

New Relic

fluentd

ElastiCache (Redis)

DynamoDB

MySQL RDS (MySQL)

OpsWorks

CloudWatch

SES

SNS

SQS

Lambda

Slack

APK

EC2

EC2

EC2

EC2

ELB

SNS

S3

# SNSの活用

- SNSとは
  - 高速かつ柔軟なフルマネージド型の通知サービス
  - さまざまなエンドポイントに通知することができる
    - HTTP(S)
    - Email
    - SQS
    - モバイルプッシュ通知
    - Lambda

# SNSの活用

- 全ユーザをja/en別で、100のトピックに均等分割
- ChatOpsで通知 (cron指定, 日時指定, 手動実行)

```
12:09      tibo BOT boss push-notify '今週のランキング確定まで残り数時間！','Just a little  
          time remaining until the next Rankings!' to prod in groups of 5 every 5 --run  
12:09      boss BOT Going to send push notifications to topic 1 to 5.  
12:10      All notifications sent. Waiting 5 minutes for next topic groups.
```

- SNSはびっくりするほど格安！
  - 全ユーザに毎月数回通知しても \$10 ちょい

# SNSの活用

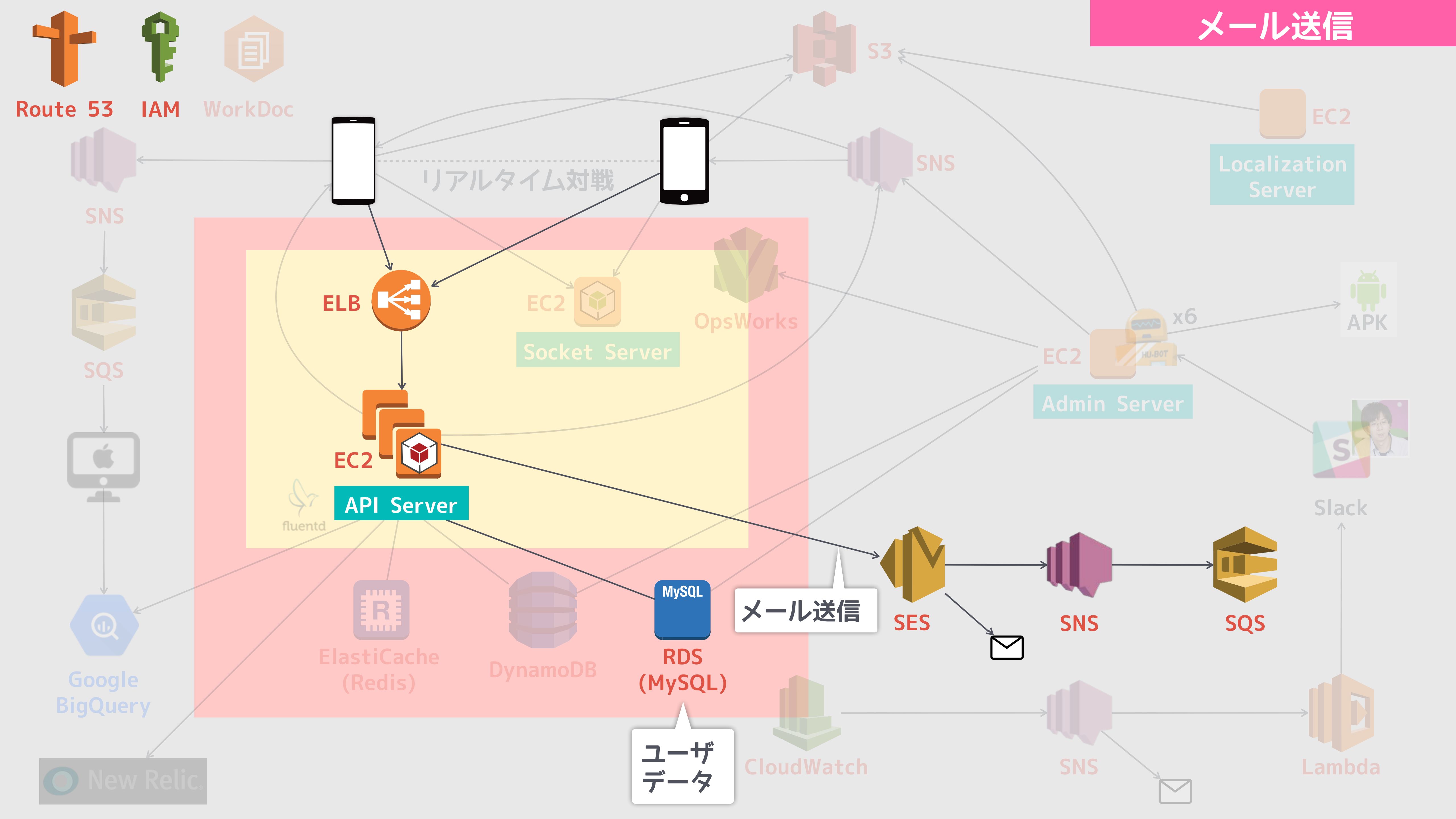
- 利用時の注意点
  - トピック数やメッセージサイズだけではなく、APIの呼び出し頻度にも制限がある（ドキュメントは未記載）
- 制限に引っかかるとAPIがRate Exceedエラーに
  - バッチなどでまとめてデバイス登録する場合に注意（1000万件の登録に1週間かかった…）



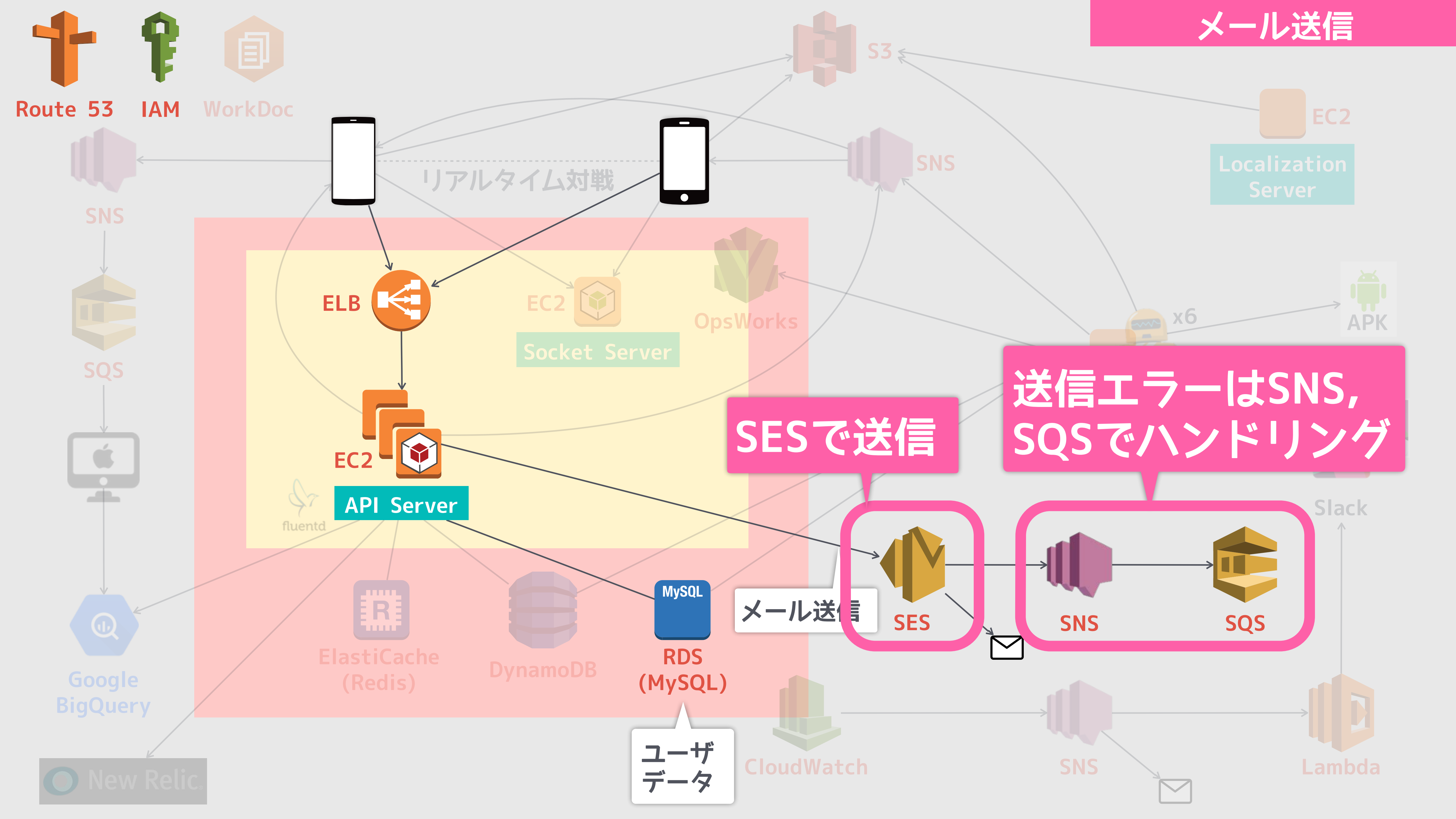
# BrainWars

メール送信

のアーキテクチャ







# メール送信

リアルタイム対戦

送信エラーはSNS,  
SESで送信

送信エラーはSNS,  
SQSでハンドリング

SESで送信

メール送信

ユーザ  
データ

Route 53 IAM WorkDoc

EC2  
Localization  
Server

EC2  
Socket Server

EC2  
API Server

ElastiCache  
(Redis)

DynamoDB

MySQL  
RDS  
(MySQL)

Google  
BigQuery

New Relic

CloudWatch

SNS

Lambda

Slack

SNS  
SQS

S3

SNS

x6

APK

SES

SNS

SQS



# SES, SNS, SQSの活用

- デバイス移行用のメールアドレス登録で利用
- SESの注意点
  - バウンス率, 苦情率, コンテンツの質によってはAWSから警告を受けたり一時停止処分を受ける
  - エラーになったメールはいつでも確認できるようにSNS経由でSQSにキューイング  
(SQS最大保持期間の14日分のみ)

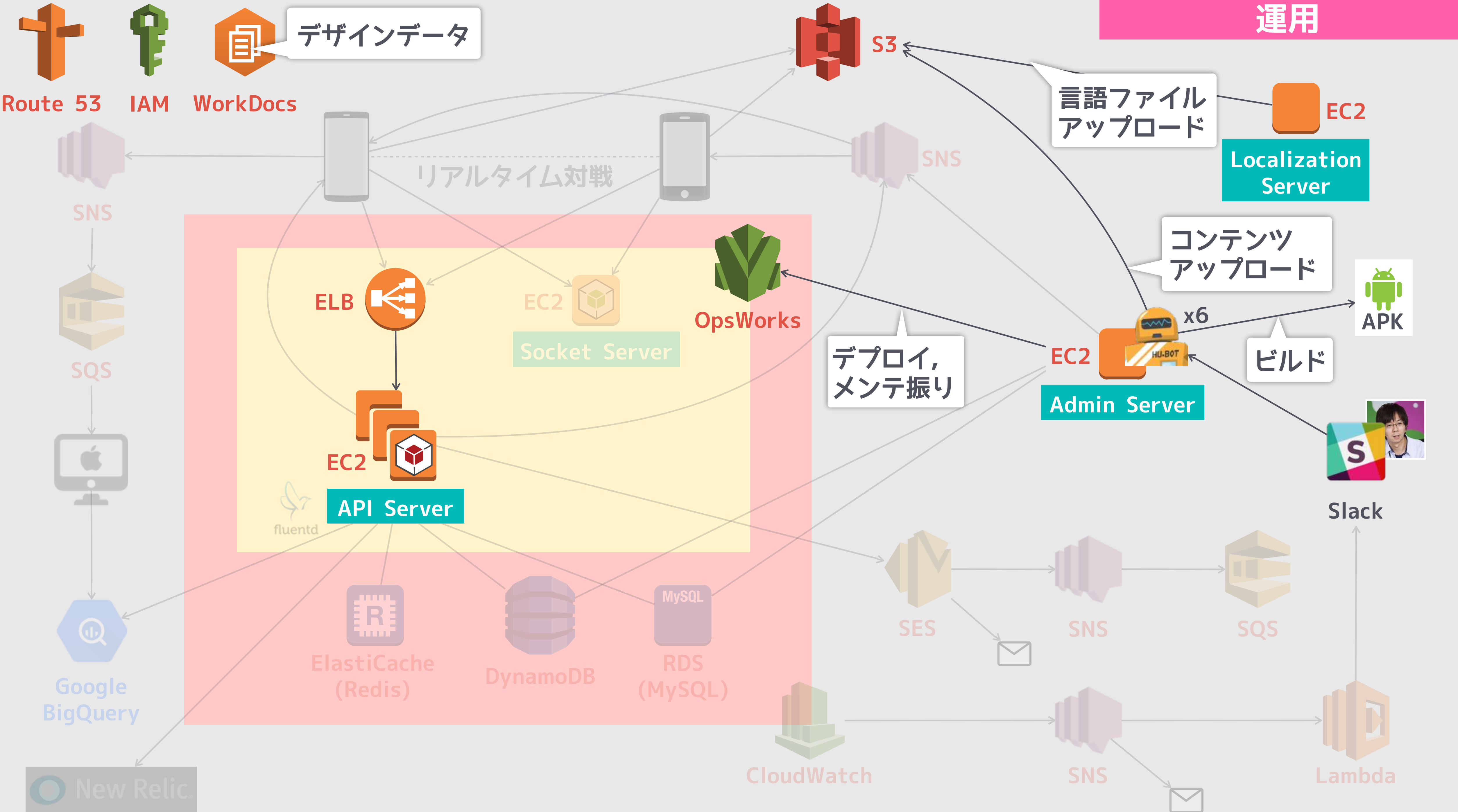


# BrainWars

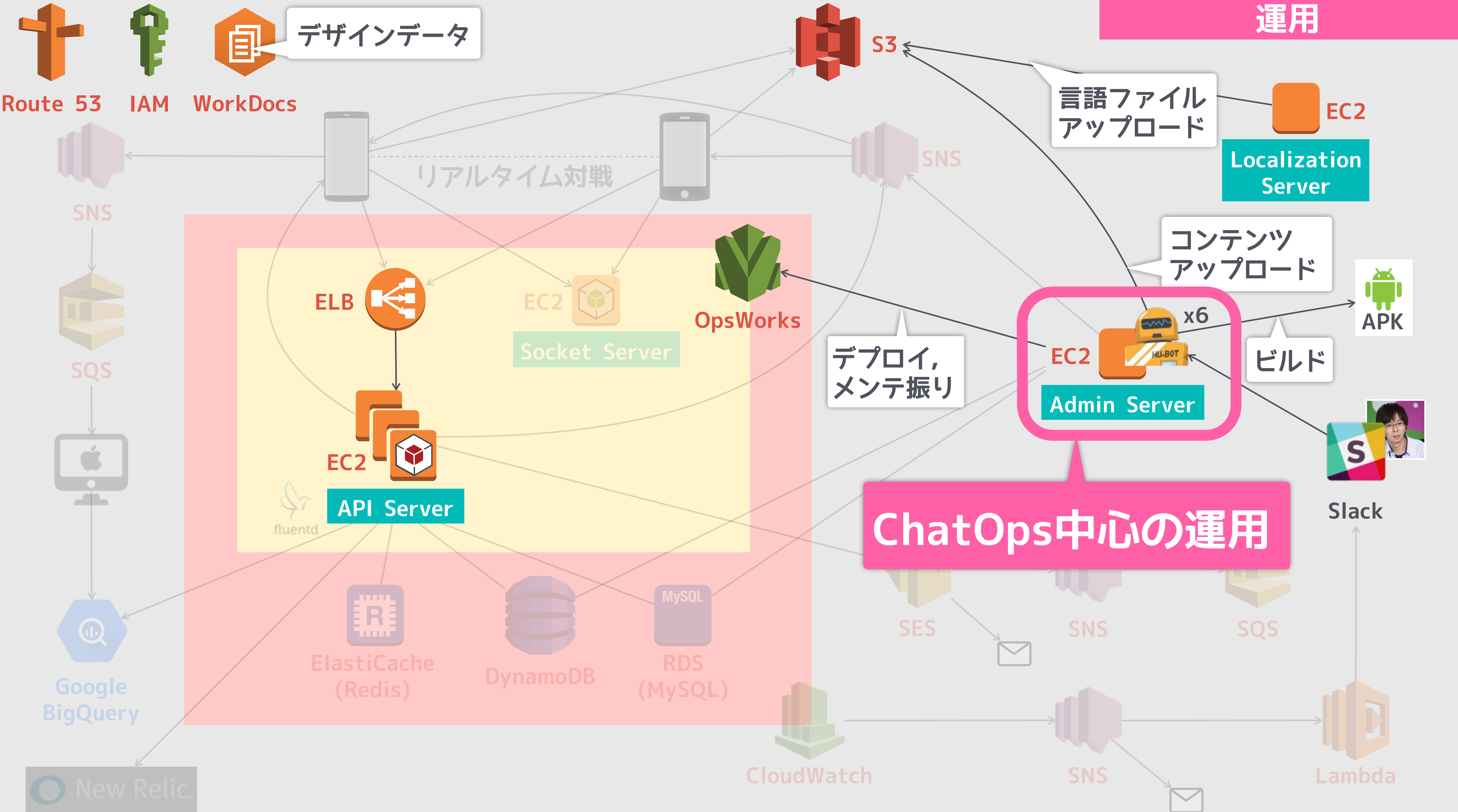
## 運用

## のアーキテクチャ

運用



# 運用



デザインデータ

S3

言語ファイル  
アップロード

Localization  
Server

リアルタイム対戦

コンテンツ  
アップロード

APK

デプロイ,  
メンテ振り

ビルド

Admin Server

ChatOps中心の運用



Slack

API Server

ElastiCache  
(Redis)

DynamoDB

RDS  
(MySQL)

CloudWatch

SNS

Lambda

New Relic

Google  
BigQuery

Route 53 IAM WorkDocs

SNS

SQS

ELB

EC2

OpsWorks

Socket Server

EC2

EC2

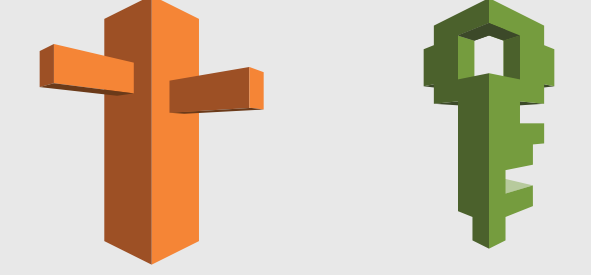
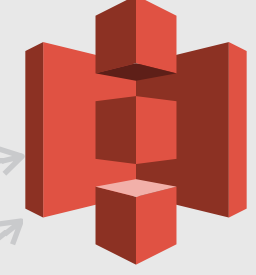
x6

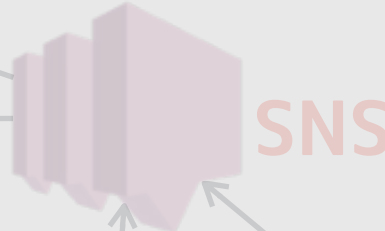
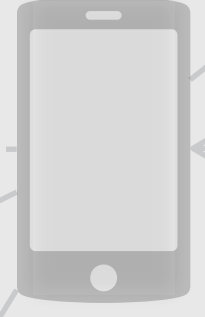
SES

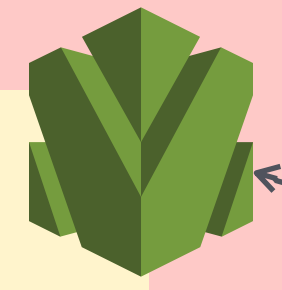
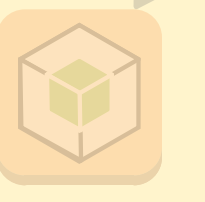
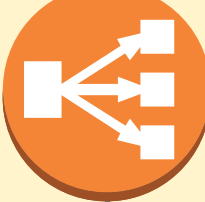
SNS

SQS

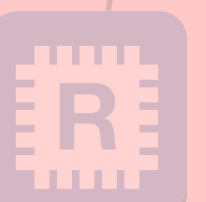
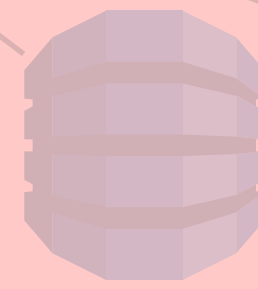


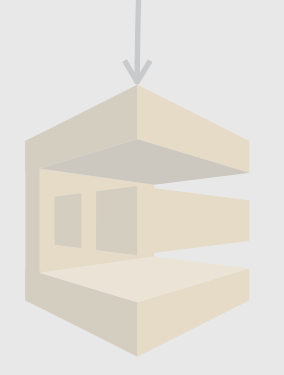


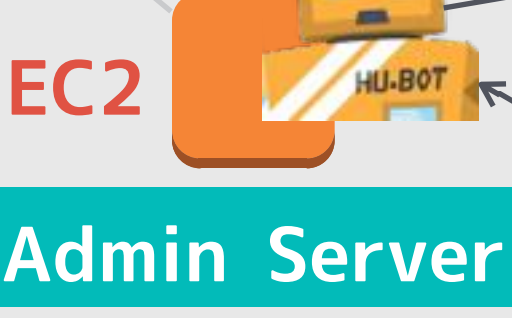
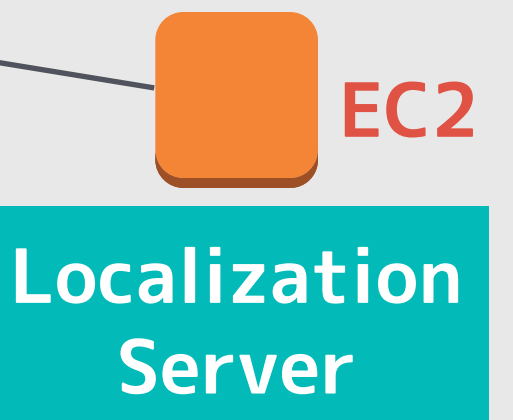



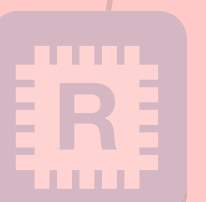
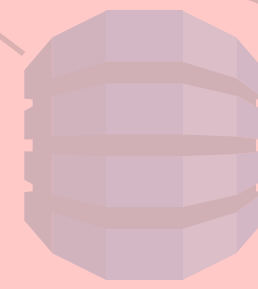








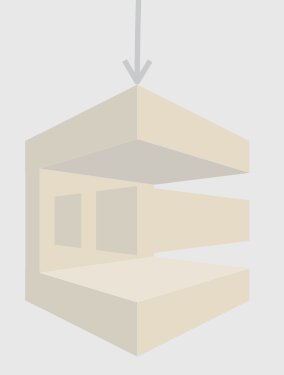



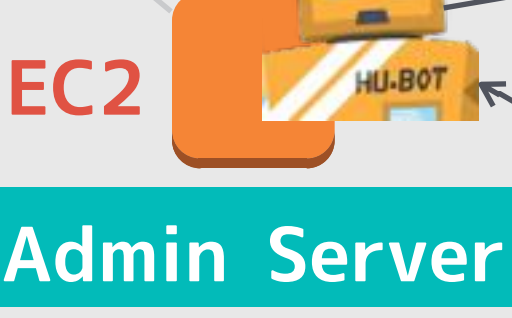
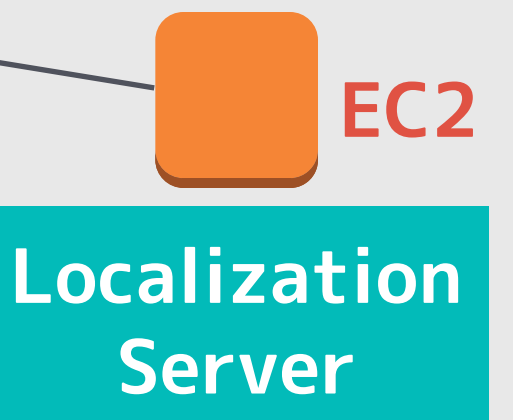




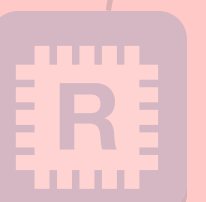
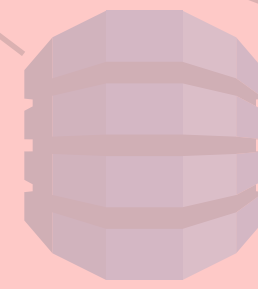








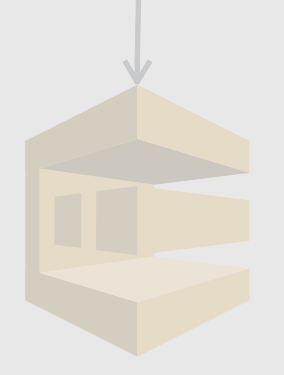



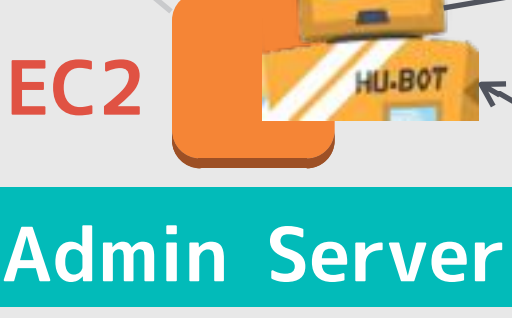






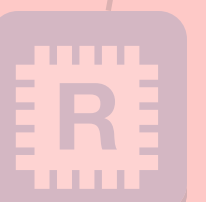
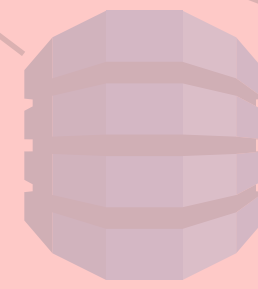








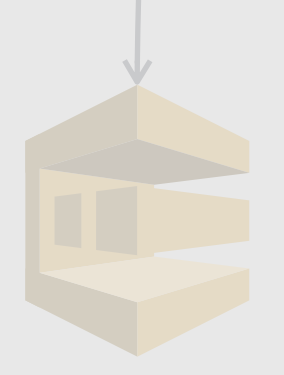


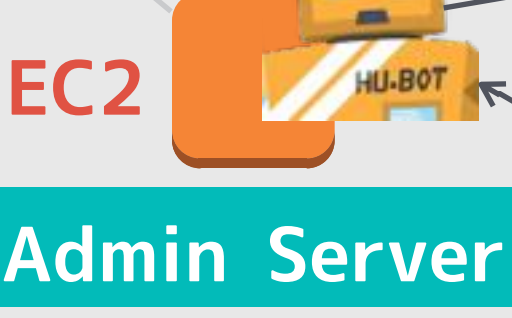
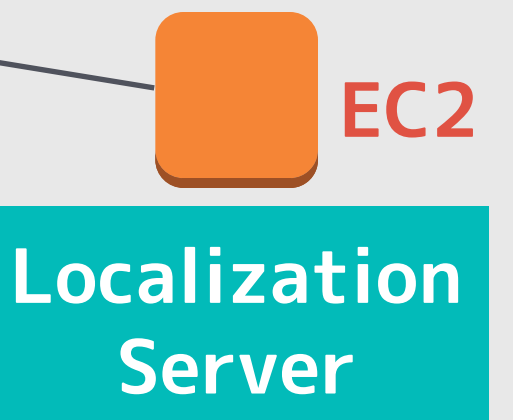



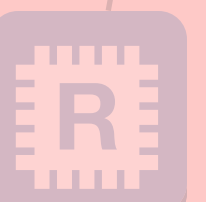
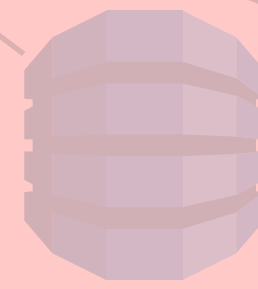








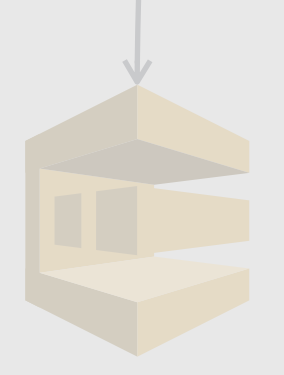



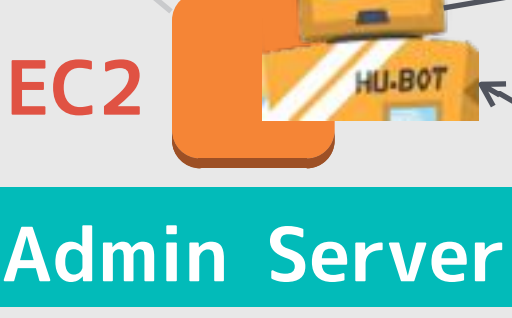
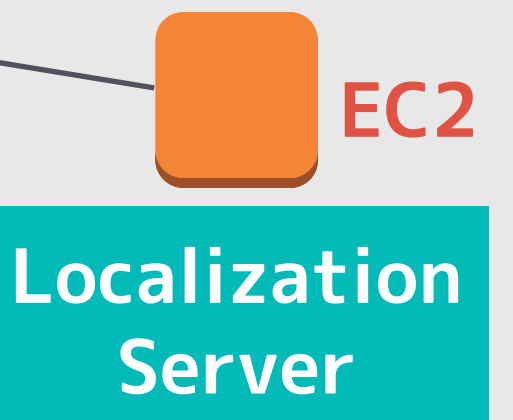




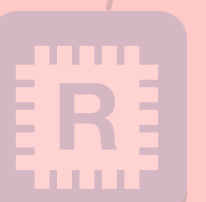
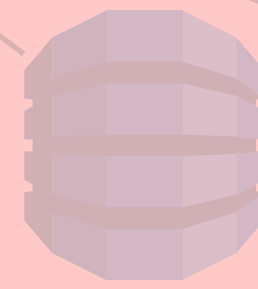








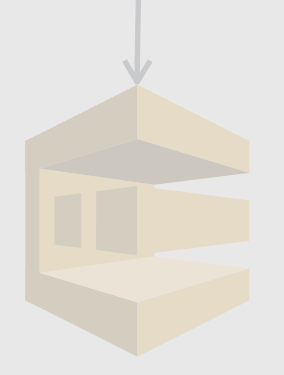



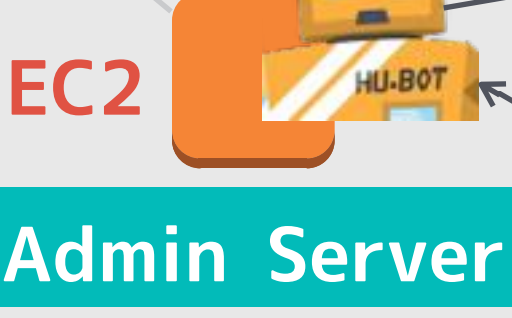
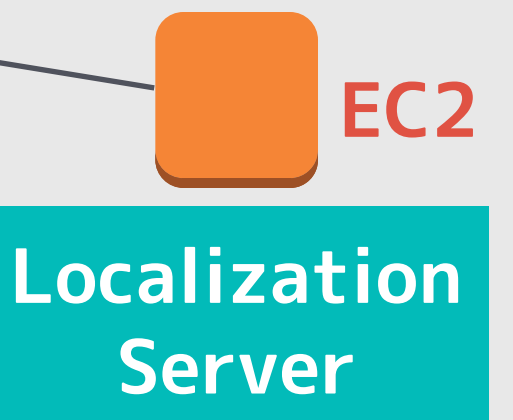




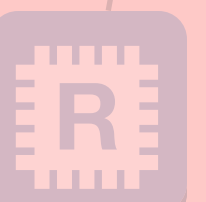
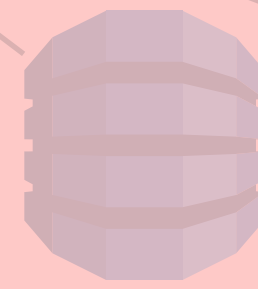








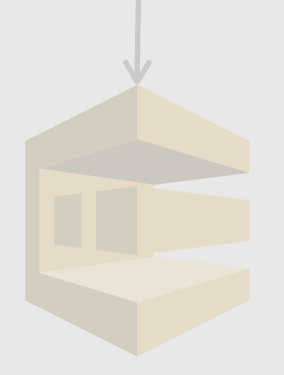



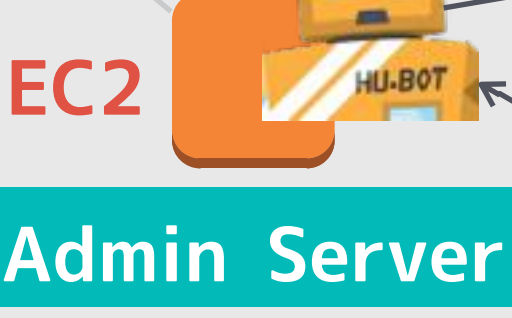
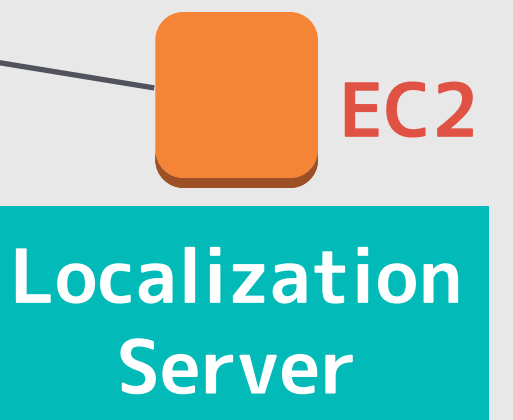




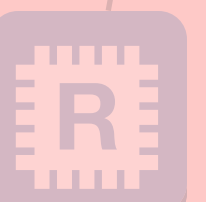
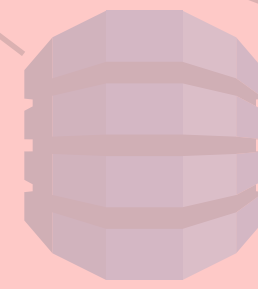








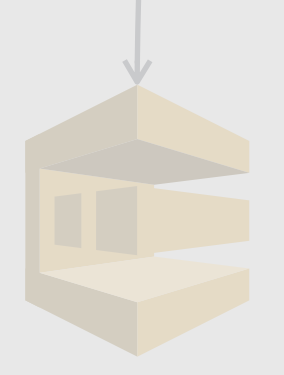



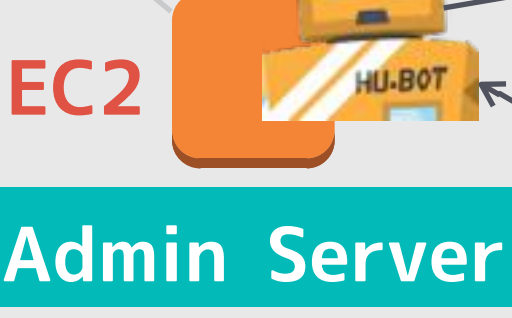
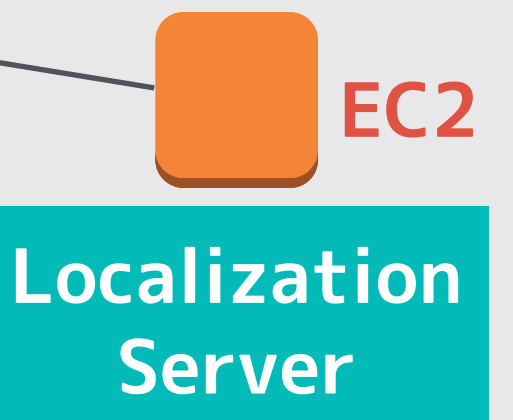




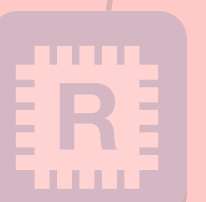
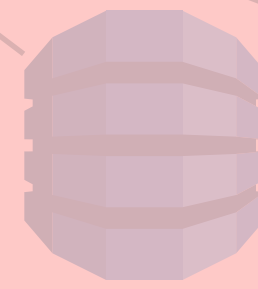








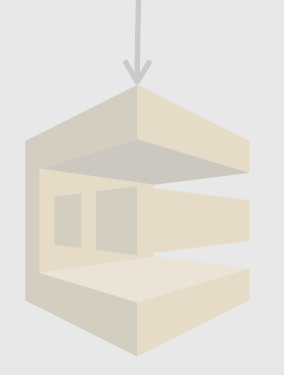



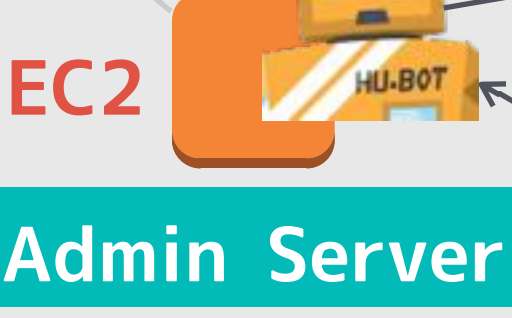
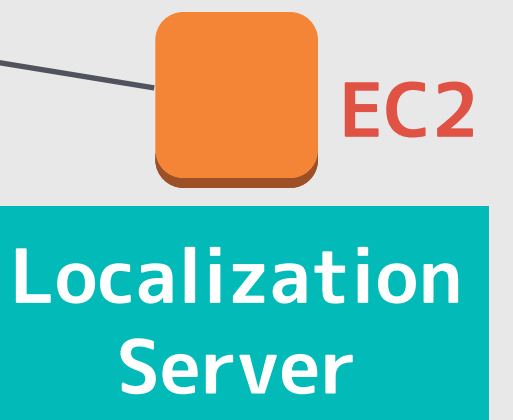




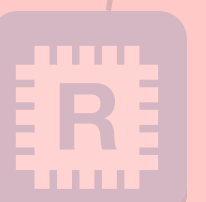
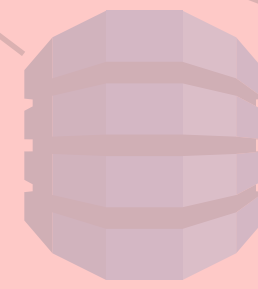








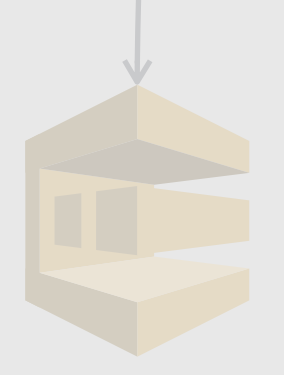



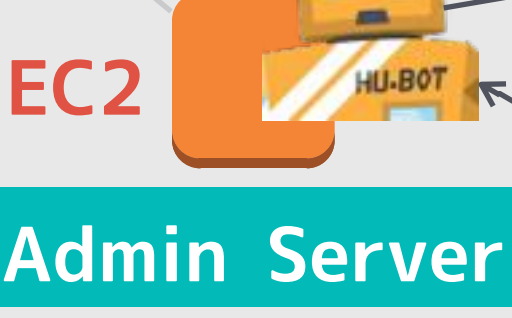
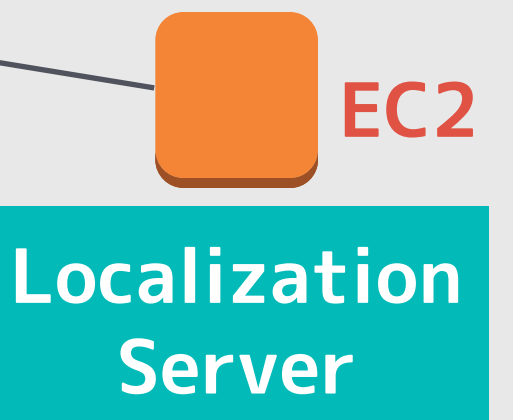




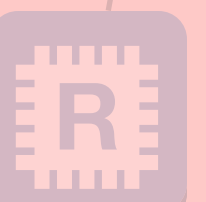
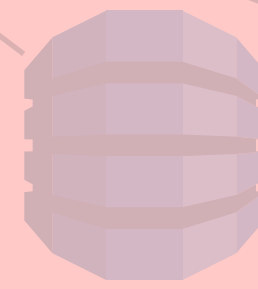








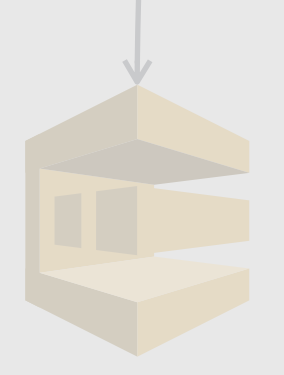



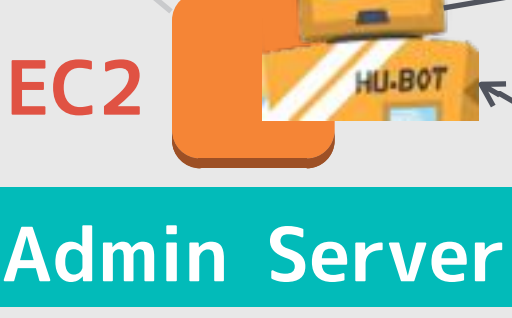
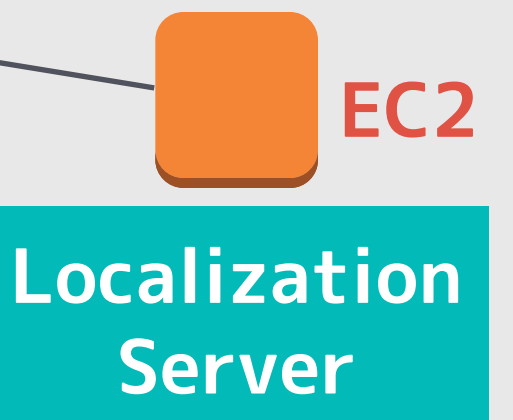




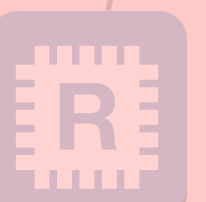
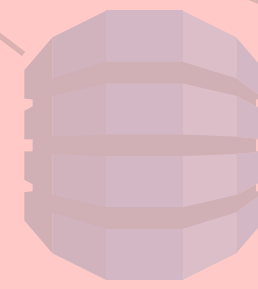








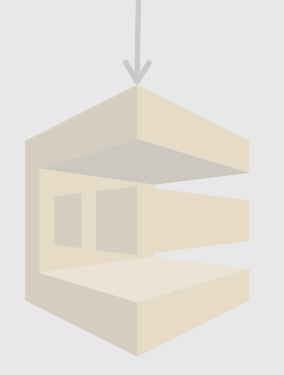



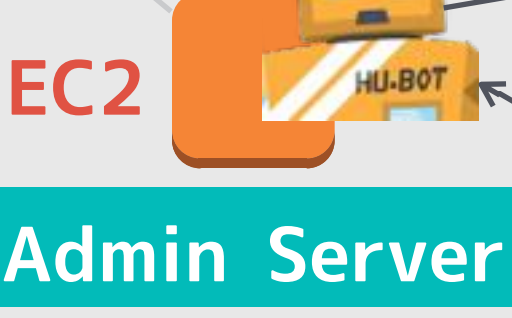
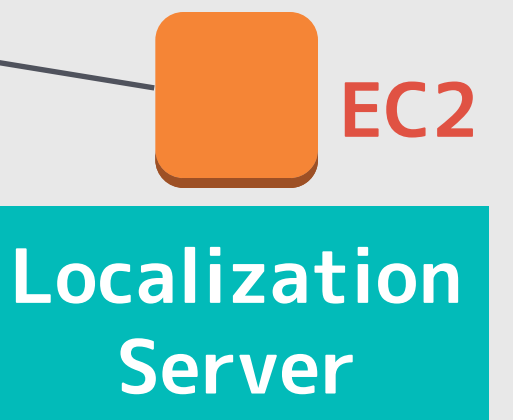




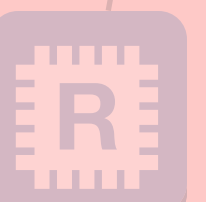
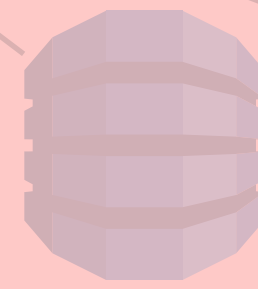








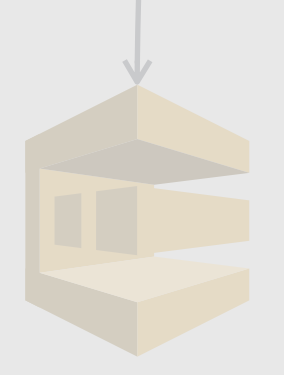



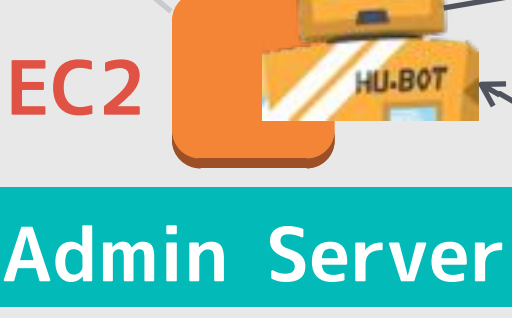
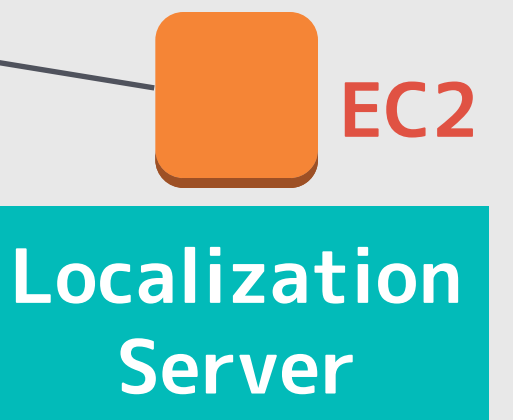




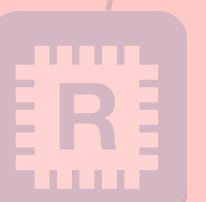
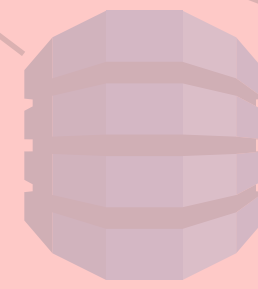








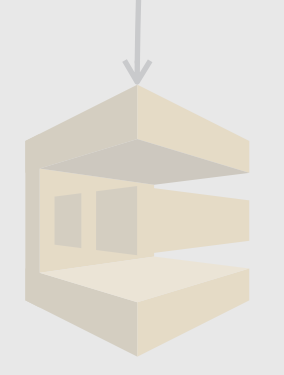



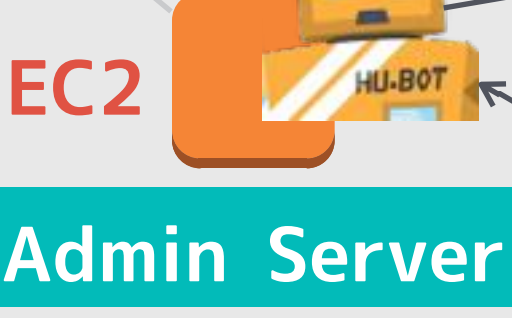
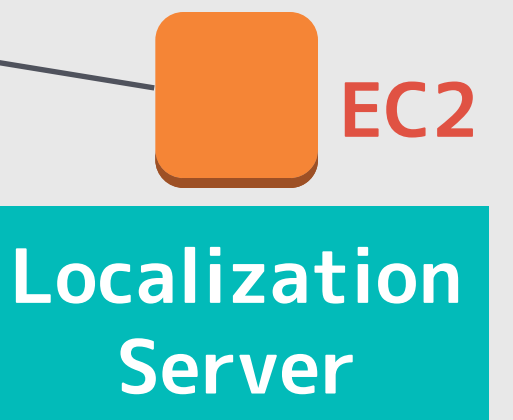




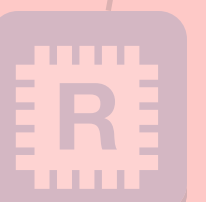
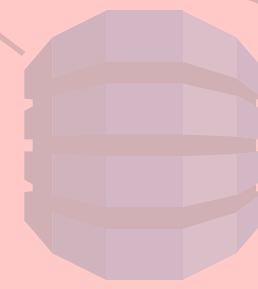








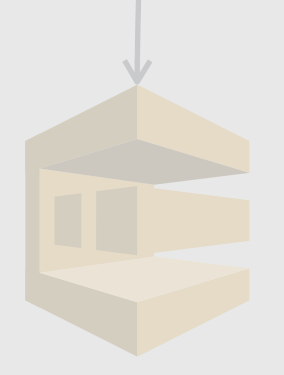



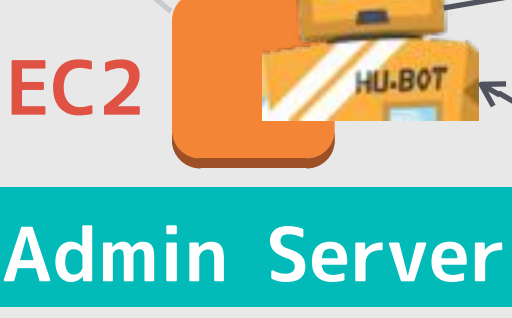
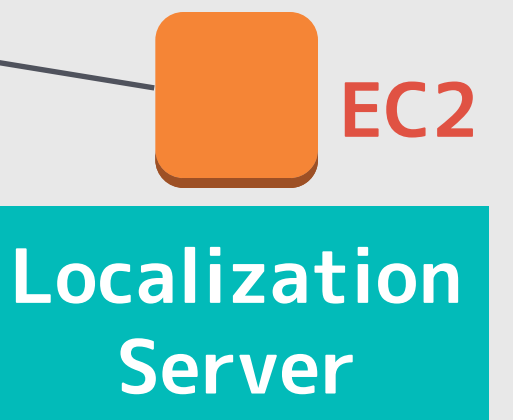




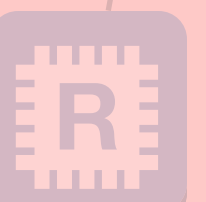
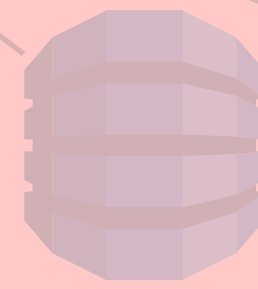








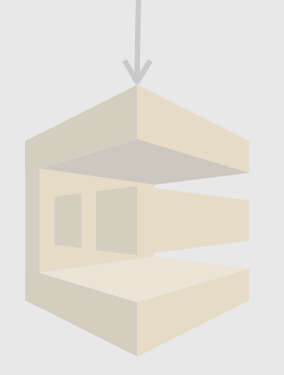



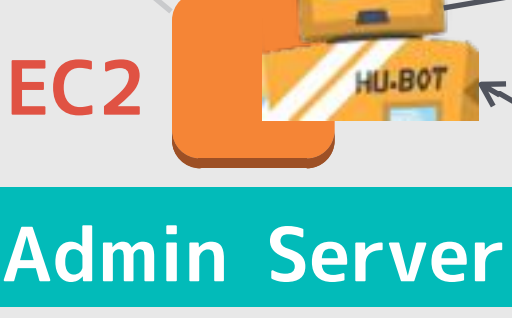
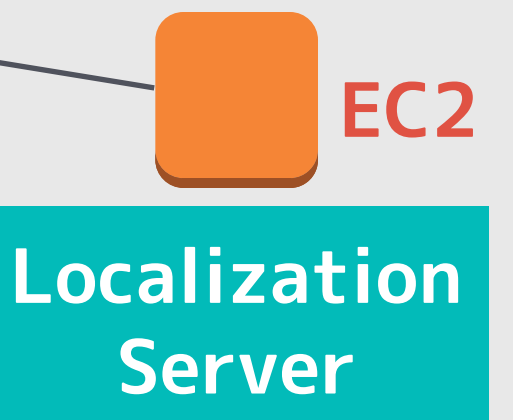




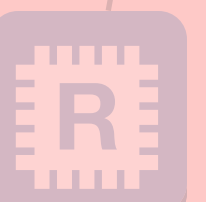
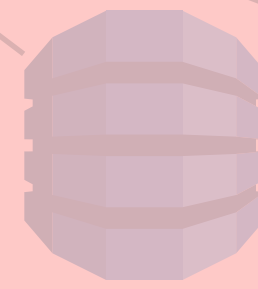








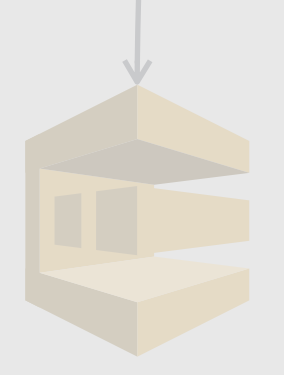



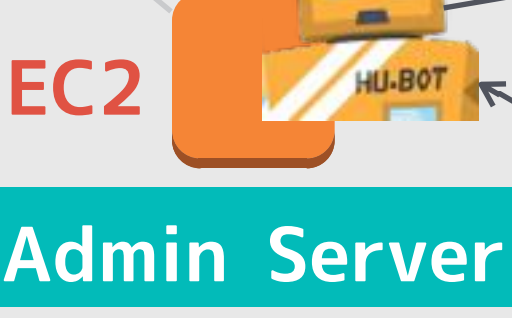
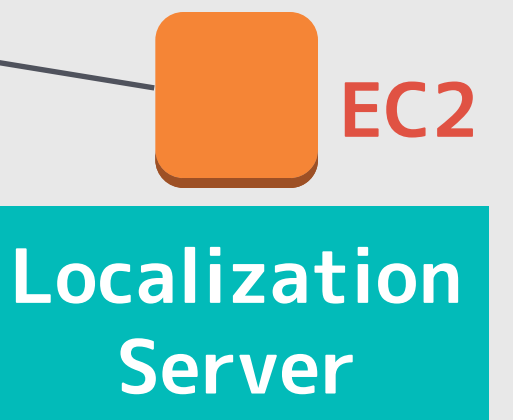




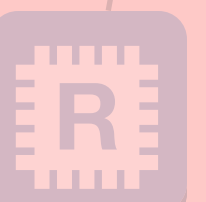
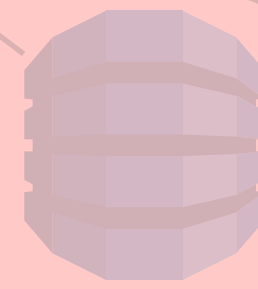








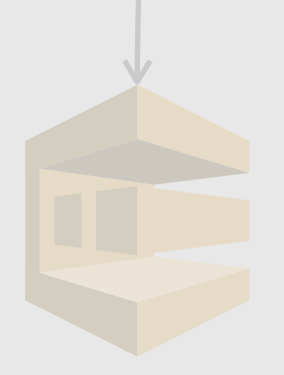



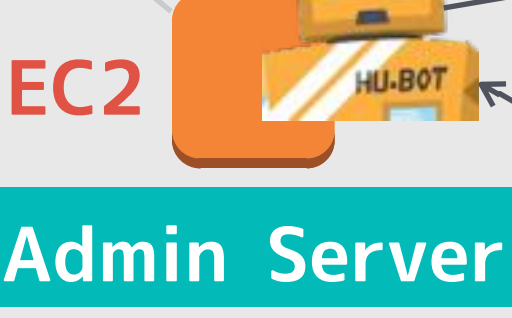
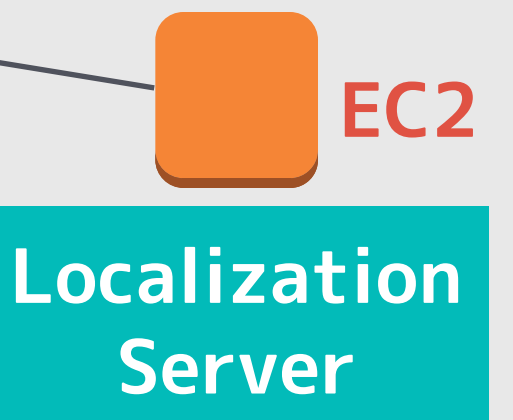




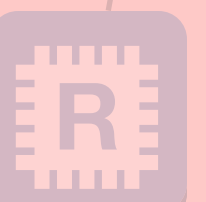
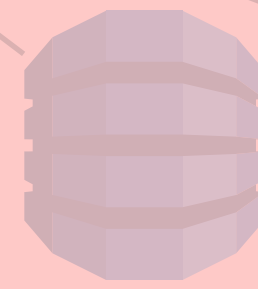








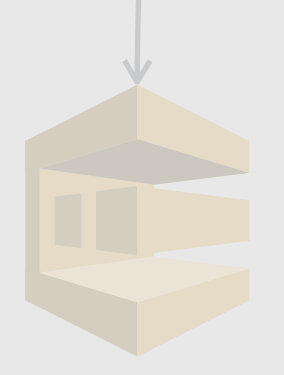



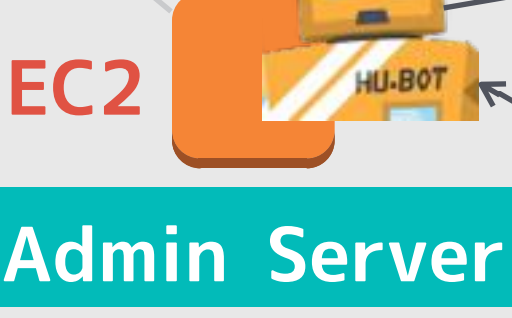
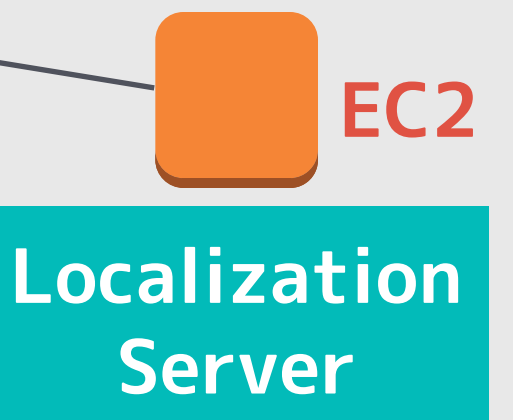




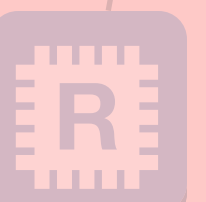
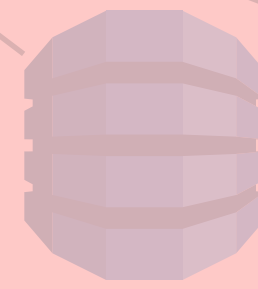








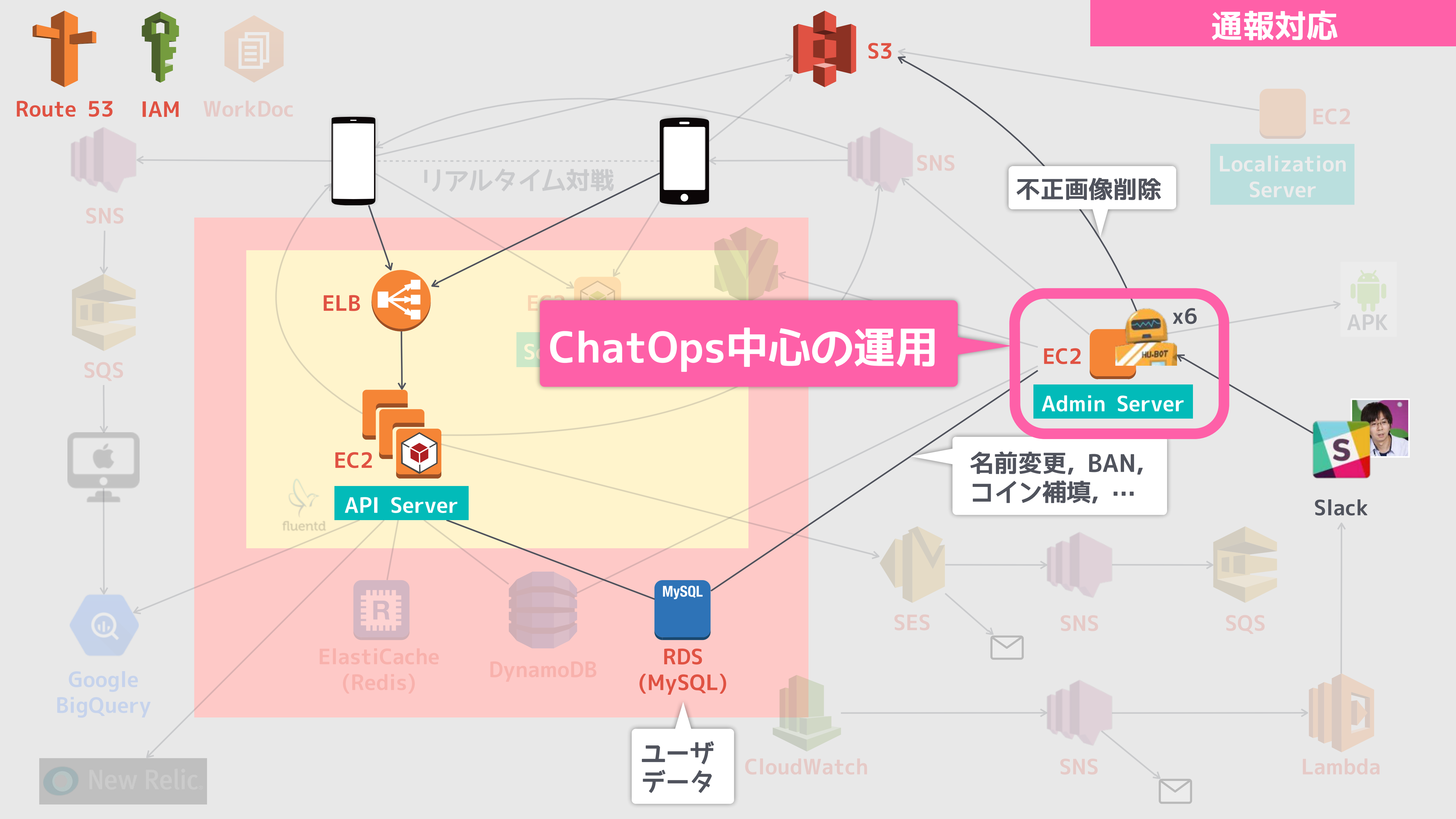


# BrainWars

## 通報対応

### のアーキテクチャ





通報対応

Route 53 IAM WorkDoc

リアルタイム対戦

ChatOps 中心の運用

不正画像削除

名前変更, BAN, コイン補填, ...

ユーザデータ

New Relic

Localization Server

Admin Server

API Server

ElastiCache (Redis)

DynamoDB

RDS (MySQL)

CloudWatch

SES

SNS

SQS

Lambda

Slack

APK

EC2 x6

ELB

EC2

S3

SNS

EC2

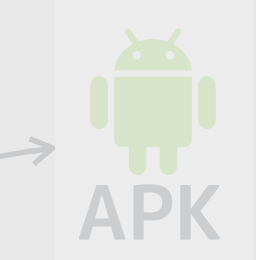
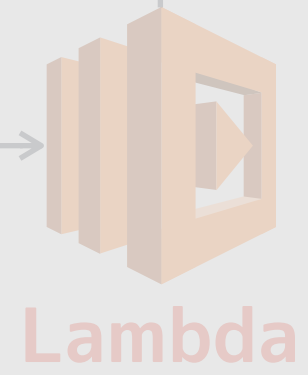
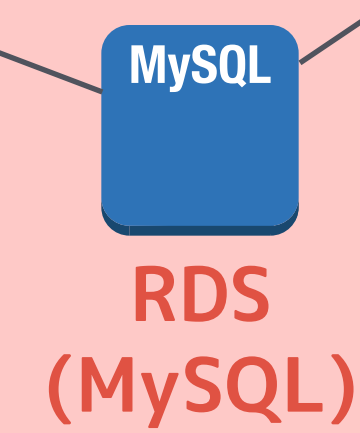
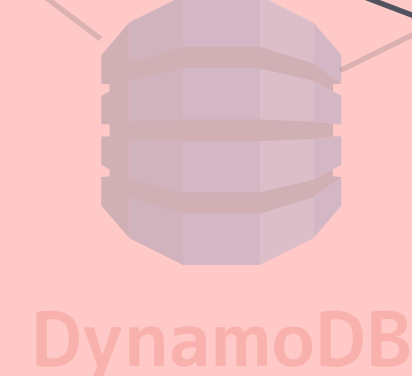
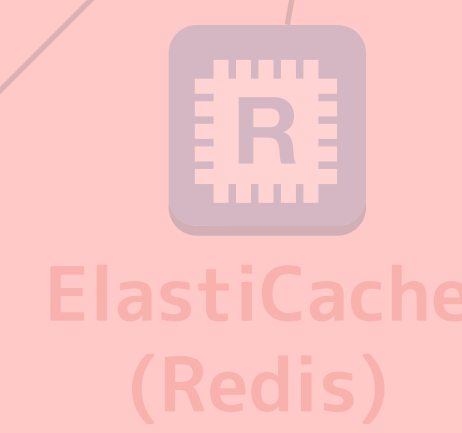
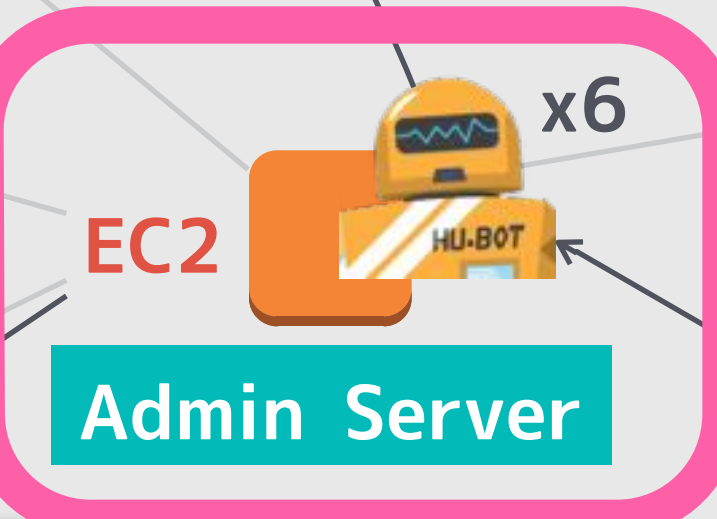
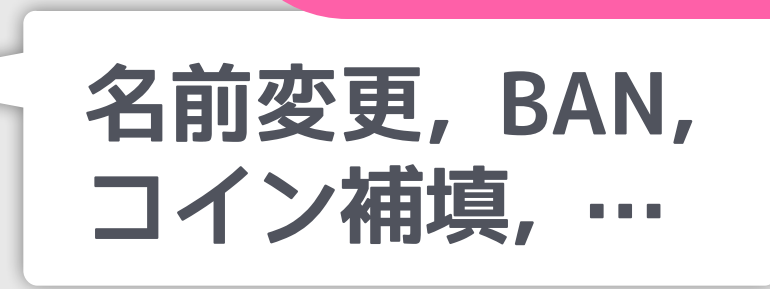
SNS

SQS

Google BigQuery



fluentd



Route 53 IAM WorkDoc

リアルタイム対戦

ChatOps 中心の運用

不正画像削除

名前変更, BAN, コイン補填, ...

ユーザデータ

New Relic

Localization Server

Admin Server

API Server

ElastiCache (Redis)

DynamoDB

RDS (MySQL)

CloudWatch

SES

SNS

SQS

Lambda

Slack

APK

EC2 x6

ELB

EC2

S3

SNS

EC2

SNS

SQS

Google BigQuery



fluentd





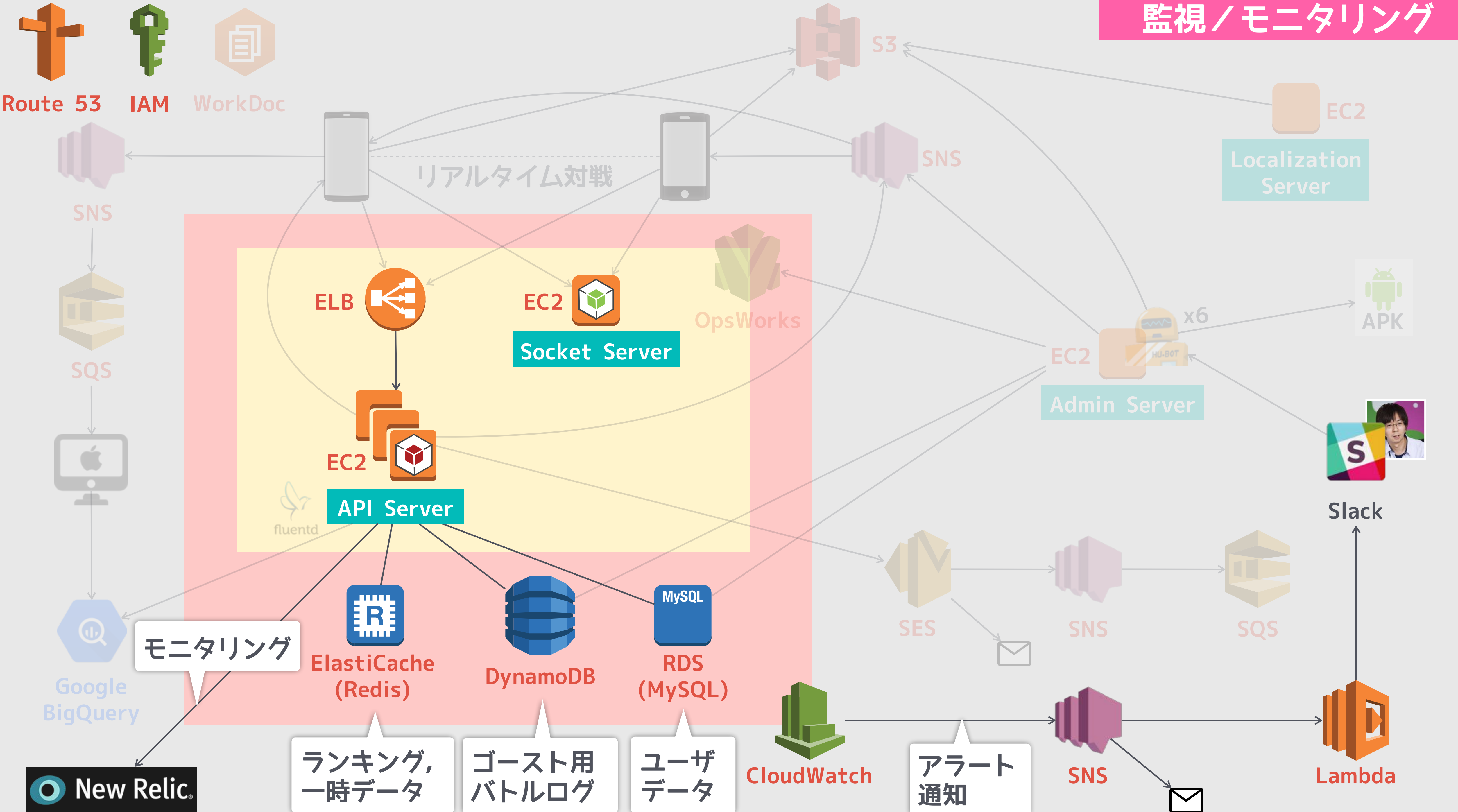


# BrainWars

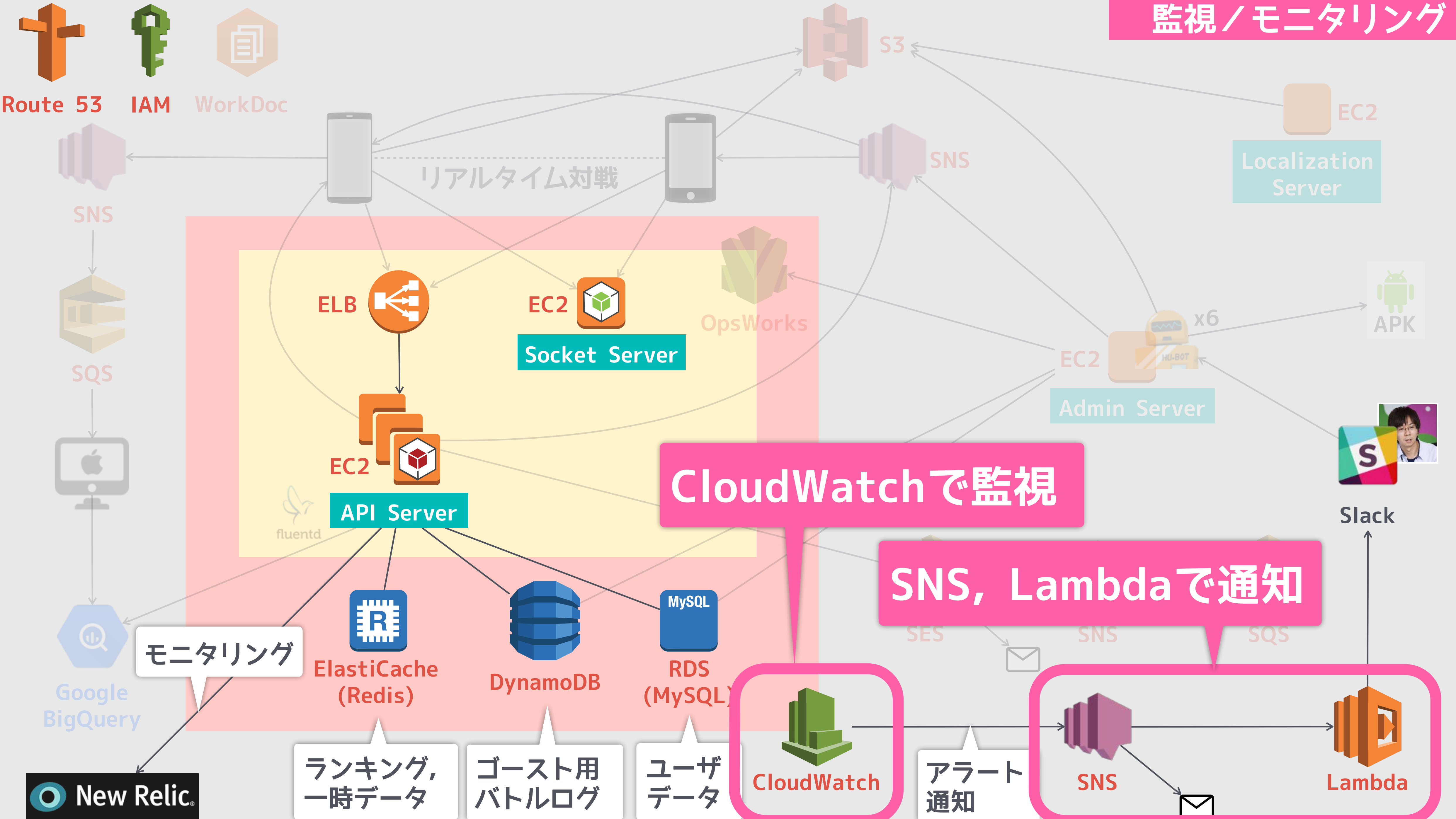
監視 / モニタリング

のアーキテクチャ

# 監視 / モニタリング



# 監視 / モニタリング



CloudWatchで監視

SNS, Lambdaで通知

モニタリング

ランキング、一時データ

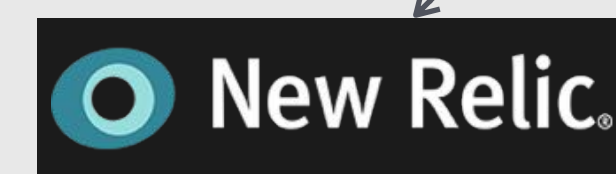
ゴースト用バトルログ

ユーザデータ

CloudWatch

アラート通知

SNS Lambda



# CloudWatch, SNS, Lambdaの活用

- インスタンス構築時にCloudWatchのアラームも登録
- アラームに引っかかったらSNSへ
- SNSからメール送信
- SNSからLambdaにも通知
- LambdaからSlack APIを実行してChatOps！



15:22 taro BOT

**CPU utilization overload**  
Threshold Crossed: 1 datapoint (83.084) was greater than the threshold (80.0).

AlarmName	Status
prod-brainwars-server-rails-app1-i-682a779b-cpu-usage1	ALARM

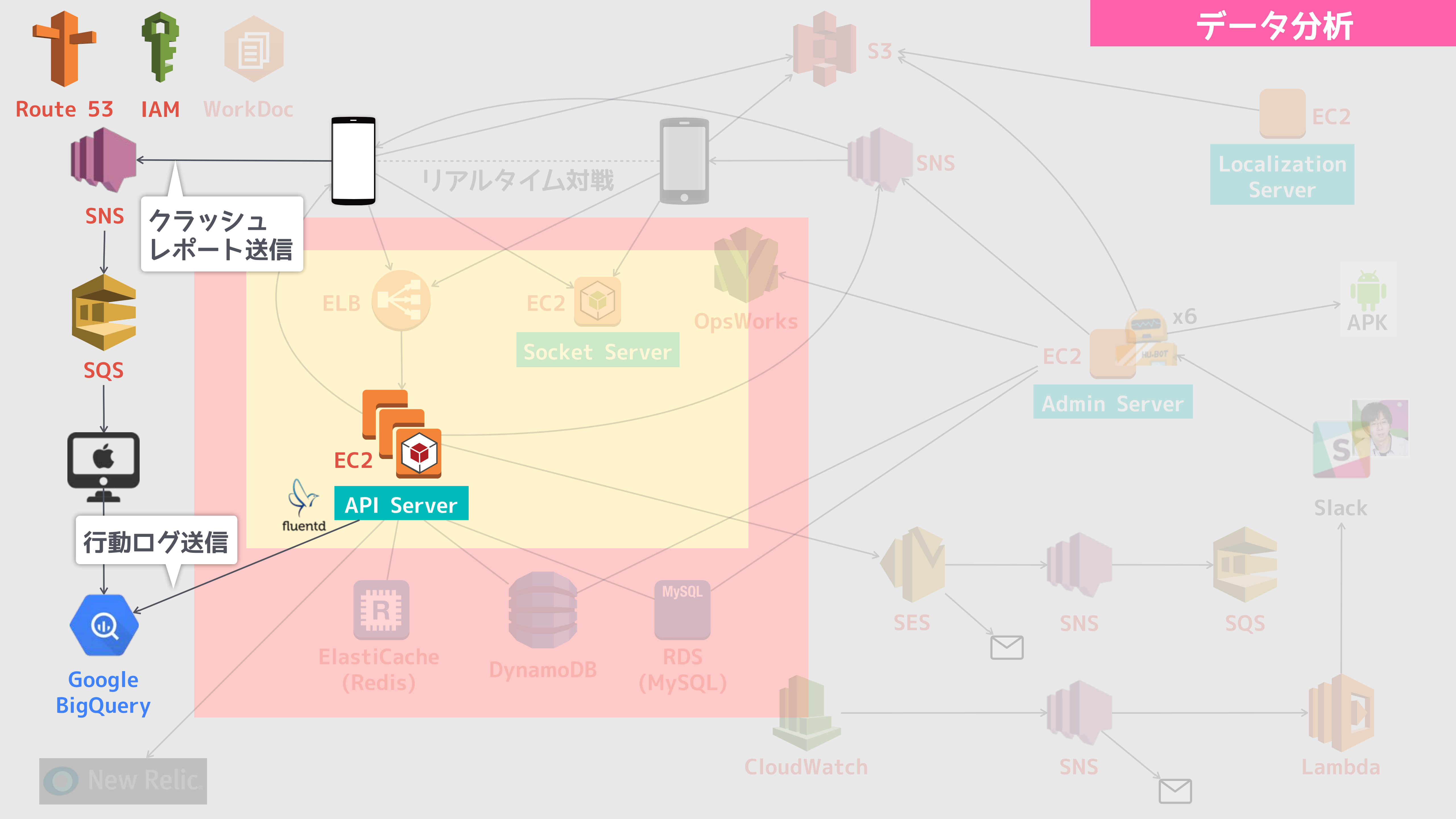


# BrainWars

## データ分析

## のアーキテクチャ

# データ分析



# データ分析

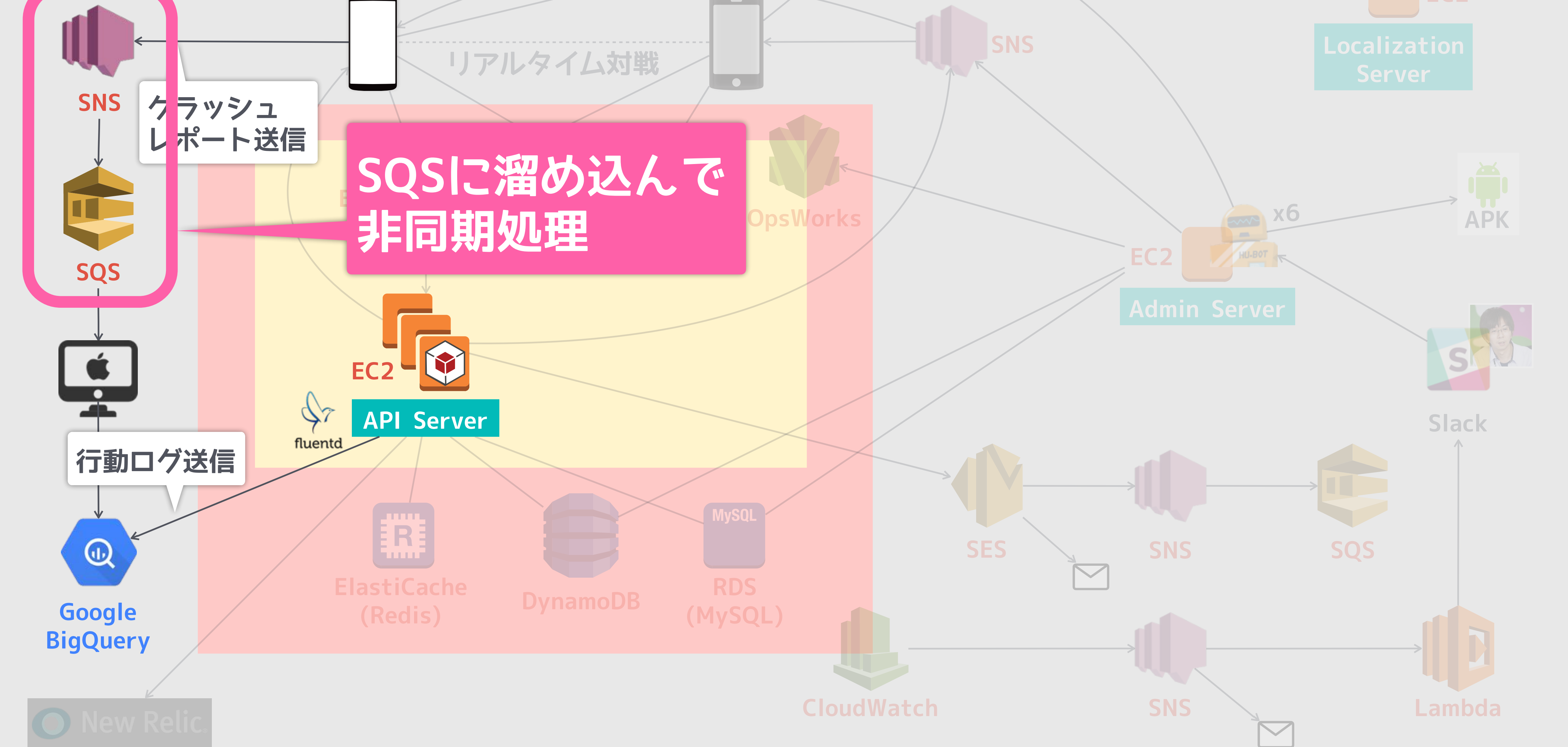
SQSに溜め込んで  
非同期処理

リアルタイム対戦

クラッシュ  
レポート送信

行動ログ送信

Route 53 IAM WorkDoc



# SQSの活用

- クラッシュレポートはひとまずSQSにキューイング
- そこまでリアルタイム性は求めていない
- デザイナーさん用iMacの常駐プログラムがSQSからキューを取り出して解析、BigQueryに投入  
(Mac上でsymbolicateを行うため)

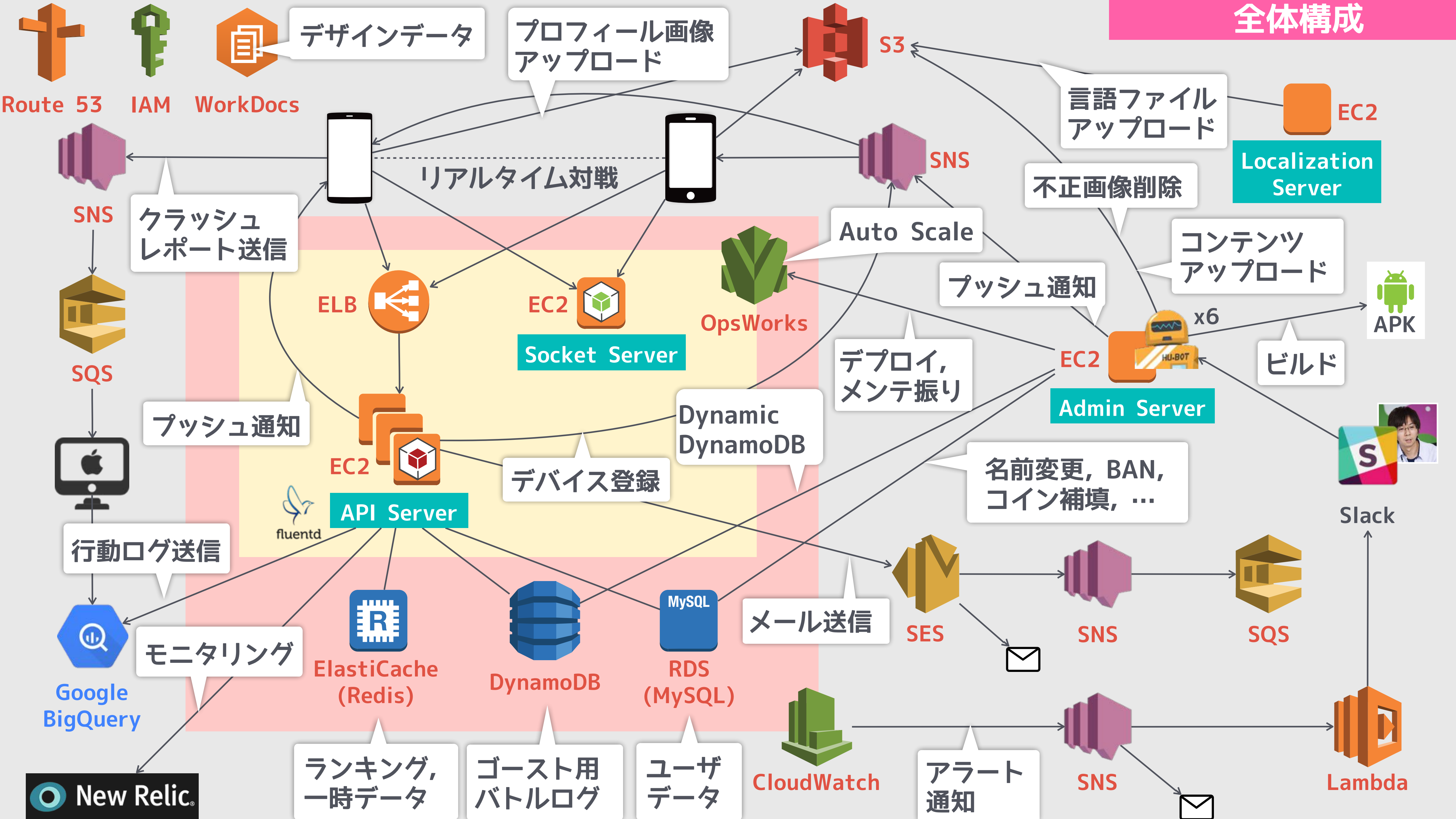




BrainWars

全体アーキテクチャ

# 全体構成



BrainWars概要

アーキテクチャ基本方針

BrainWarsとAWSサービスたち

まとめ

どのサービスも  
使いこなせれば強力！

ただし、  
事例と同じことをしても  
プロダクトや環境次第では  
うまくいかない

向いていない使い方は  
身を滅ぼします  
( ;´Д`)

大事なものは、  
それぞれのサービスの  
特徴を見極めること

機能面だけでなく、  
利用後のコスト感や  
運用イメージは  
ハッキリしていますか？



**Happy AWS Life !!!**

宣伝



# ただいま メンバーを募集中！

- エンジニア
- プロデューサー
- CFO/COO候補
- デザイナー
- プランナー

世界に響く

サービスをつくる



# 世界をターゲットに 億超えダウンロードを目指す！



5億DL～



5億DL～



17億DL～

バランスボールや  
スタンディングデスク



社内勉強会 T-Cube  
(Translimit Tech Talk)

# お待ちしております！

エンジニア

<1200万ダウンロード突破>世界に響くサービスを一緒に作りませんか

株式会社トランスリミット

埋め込み いいね! 25 ツイート 5 Bookmark 0



<http://translimit.co.jp/recruit.html>

<https://www.wantedly.com/projects/21002>