



SUMMIT  
ONLINE  
JAPAN



# Billion Transactions: Reaching new limits@PayPay

Harsh Prasad

SREチーム  
テックリーダー  
PayPay

Ayush Mittal

ペイメントチーム  
シニアソフトウェアエンジニア  
PayPay

Kuria Robert

プラットフォームチーム  
プラットフォームエンジニア  
PayPay



# Agenda

- PayPayについて
- PayPayにおけるパフォーマンス改善とスケーラビリティ
- システム移行から学んだこと
- Amazon Elasticsearch Serviceを利用したログ基盤

PayPayプロダクト部門

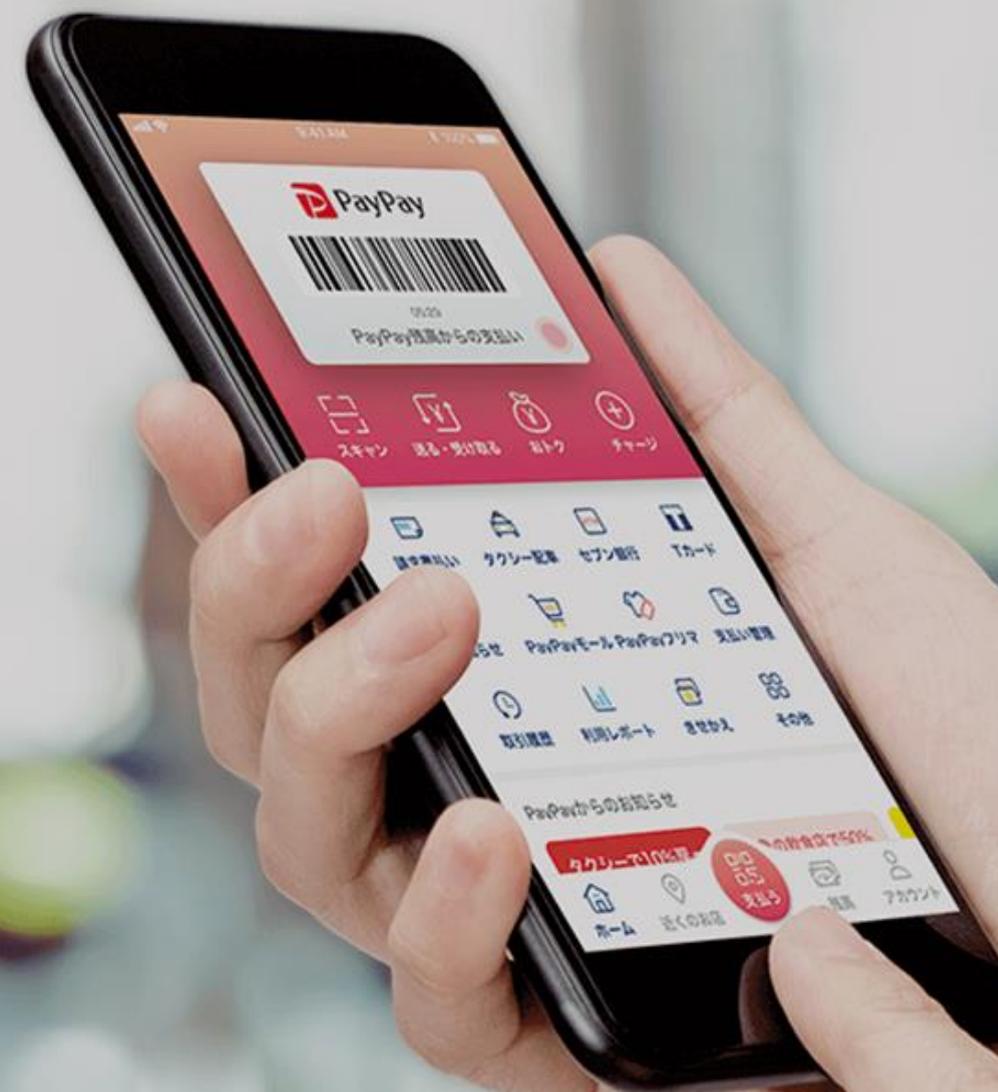
# Harsh Prasad

- インド出身
- 学士号および修士号 (コンピューター科学)
- 職歴:
  - Yahoo!JAPAN (2014年～2020年)
  - PayPayプロジェクト (2018年～現在)



# PayPayについて

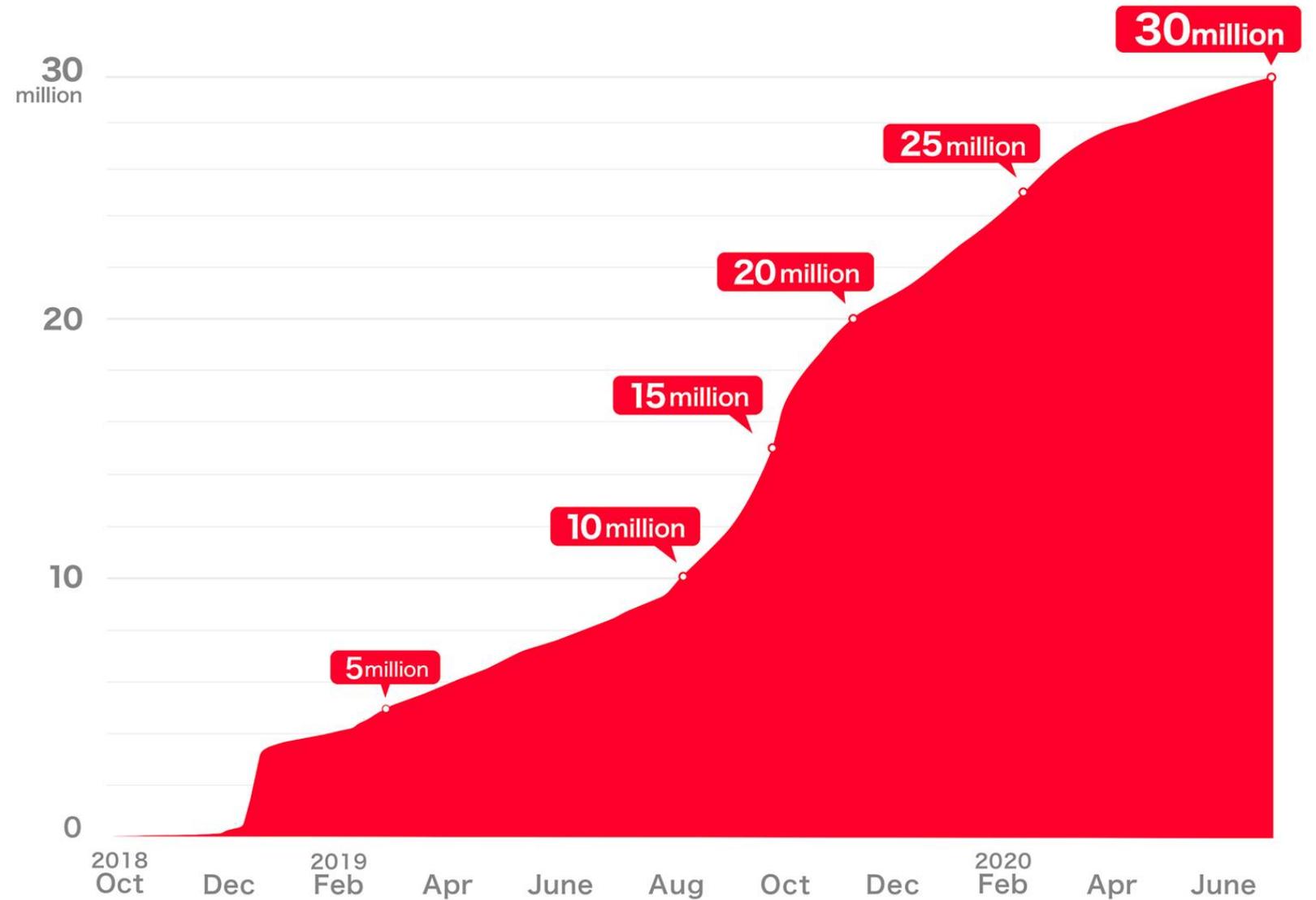
スマホひとつで  
かんたんにお支払いはPayPayで



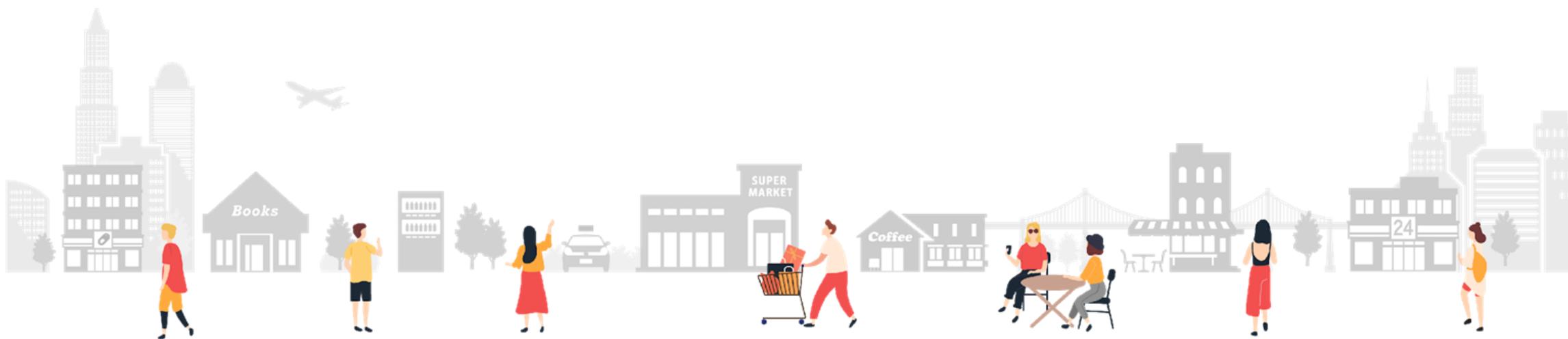
# 2018年10月5日に ローンチ

3000万ユーザー\*  
230万店舗\*

\*-2020年6月現在



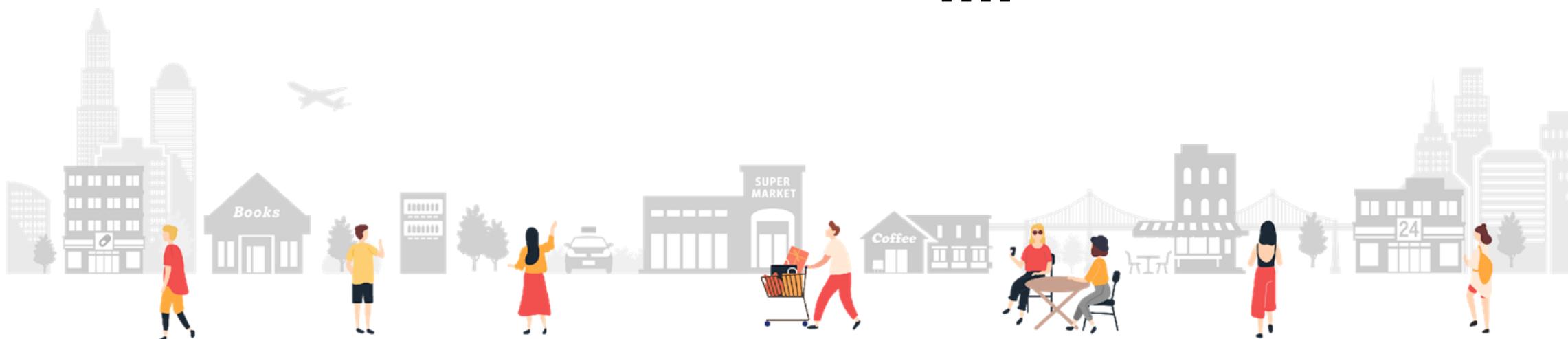
# PayPayは オンライン決済 だけではない



# PayPayは オフライン決済 だけではない

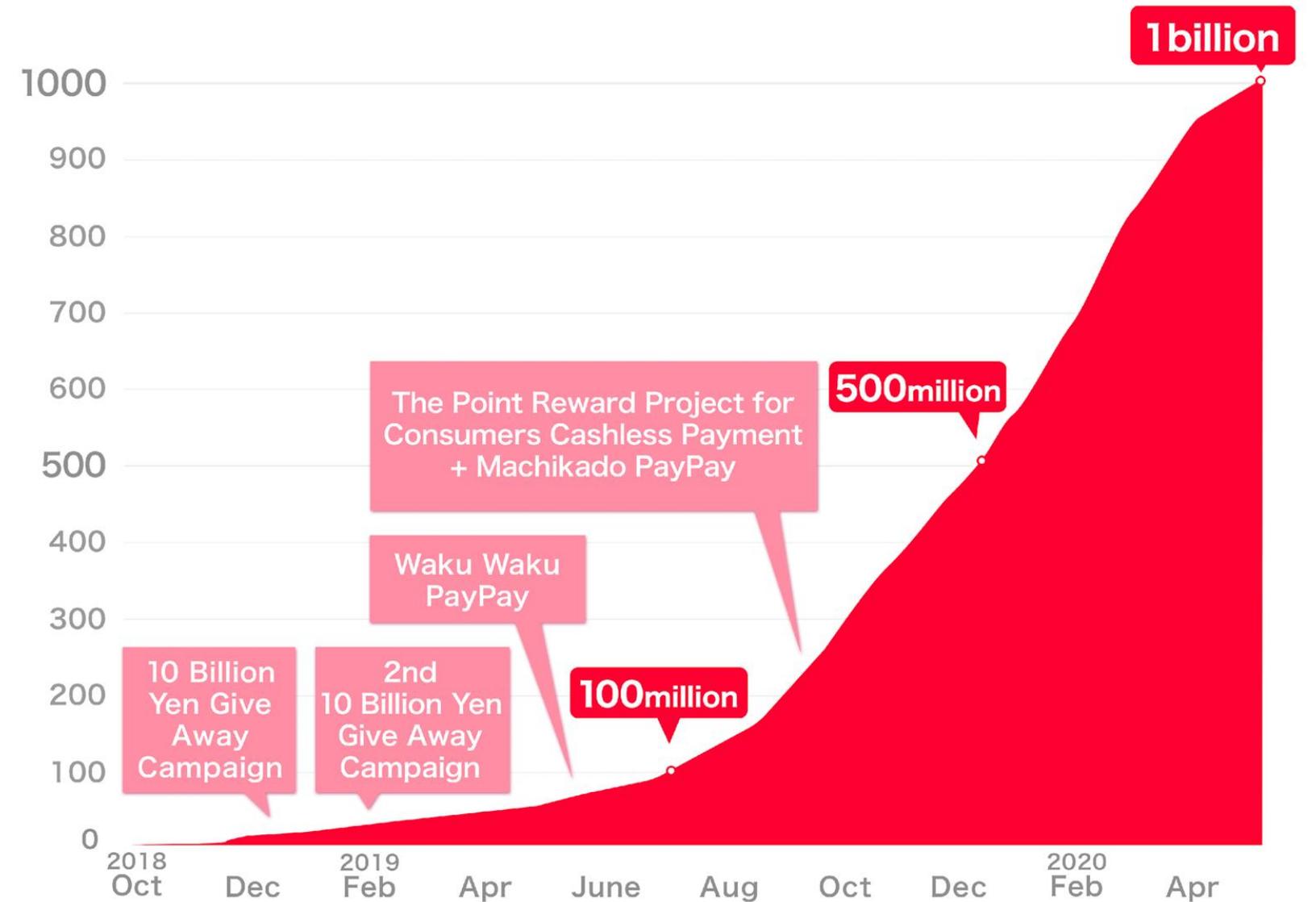
- ▶ オフライン加盟店
- ▶ オンライン加盟店
- ▶ 公共料金支払い
- ▶ P2P
- ▶ PayPayピックアップ

....



# PayPay

## 累計決済数 10億突破



この爆発的成長に  
PayPayのバックエンドは  
どのように対応したか？

# PayPayにおける パフォーマンス改善と スケーラビリティ

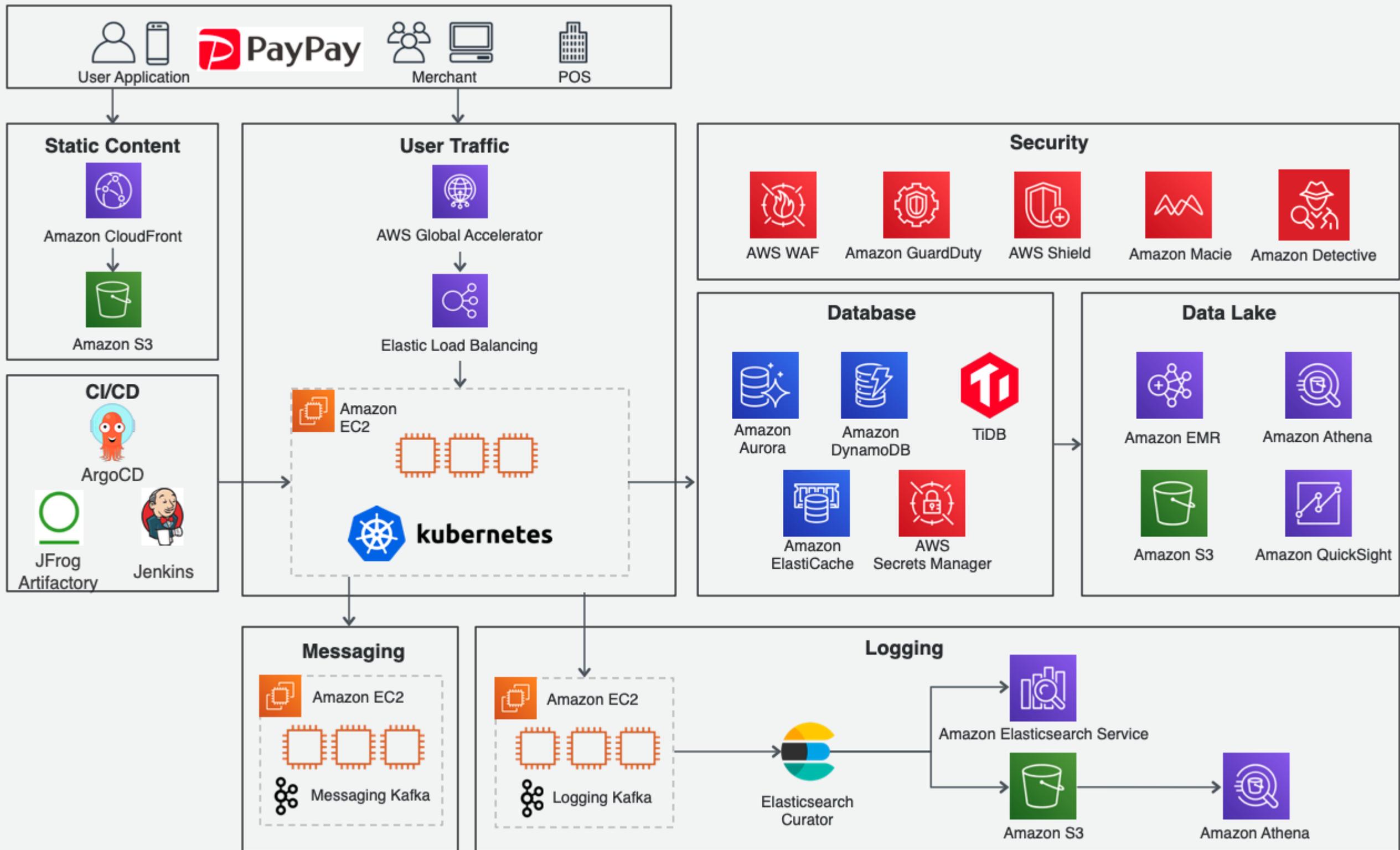
# 主要な課題

「解」より「問」が多い世界...

- どうやってキャパシティを測定する？
- ボトルネックはどこ？
- 最適なソリューションはどれ？

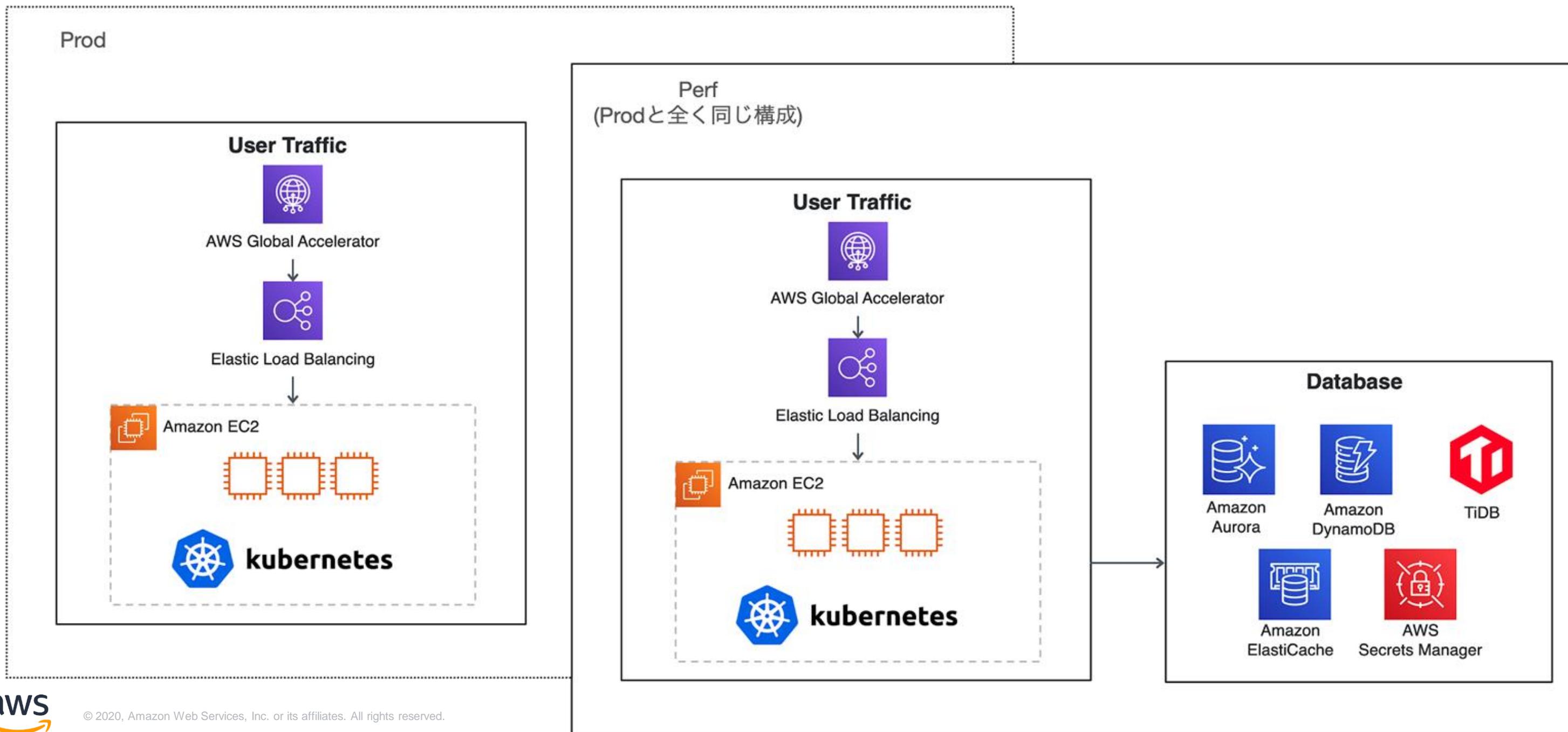


# PayPayのテック・スタッフ



どうやってシステムの  
キャパシティを測定するのか？

# キャパシティの測定



# 容量の測定

- パフォーマンステスト（社内）のために、本番環境と同規模のレプリカを構築
- どうやって?
  - Terraformでインフラを管理
  - Amazon Aurora/RDSインスタンスを定期的 to 実施するバックアップから自動リストア
  - 環境全体の自動生成によるコスト管理

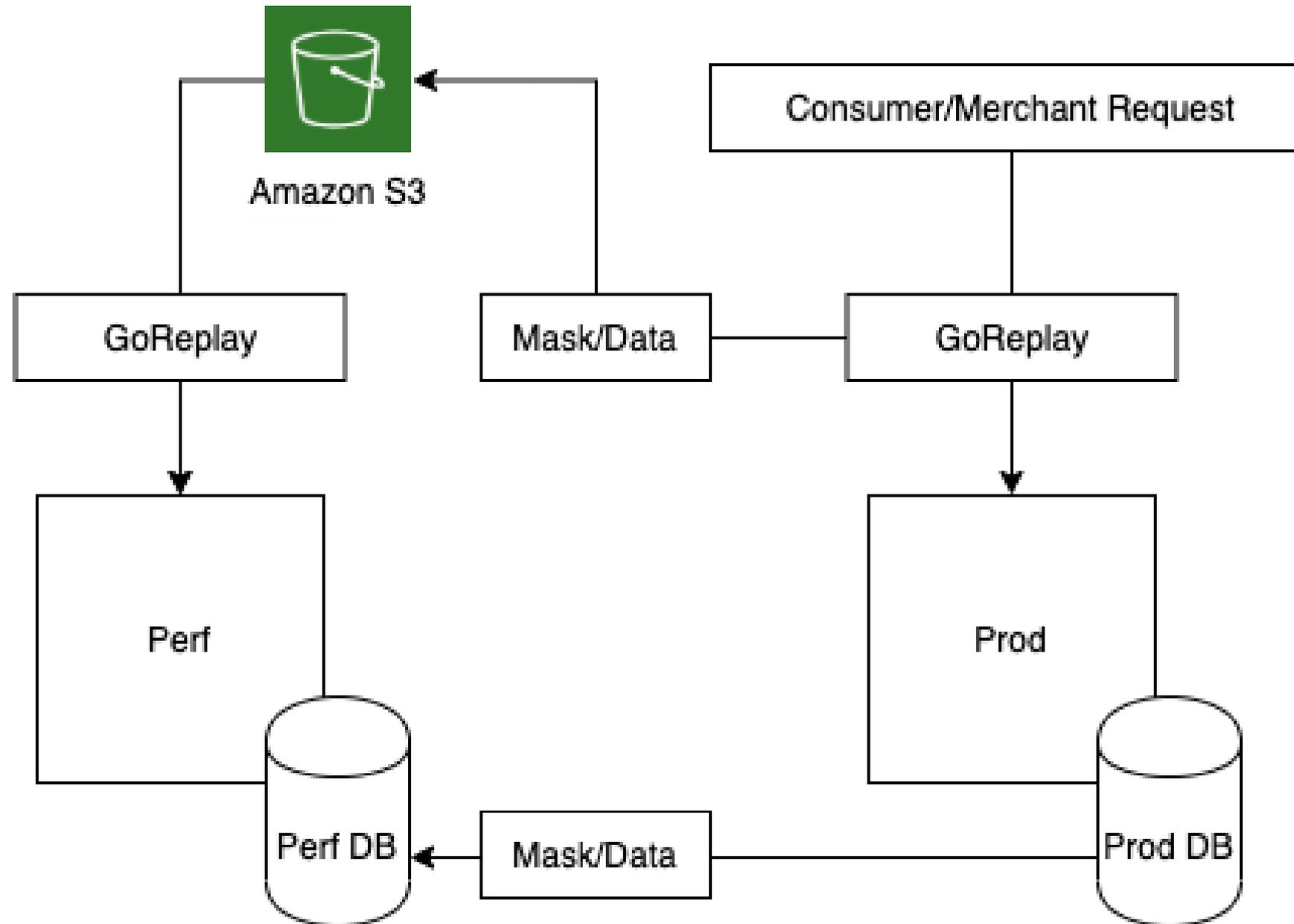
# ボトルネックの特定



# ボトルネックはどこ？

- ボトルネックを特定するには、インフラだけでなく負荷の再現も必要
- やってみたこと
  - GoReplayのようなツールを使ったトラフィック捕捉
  - Amazon S3とAmazon MSKのサポートにより、人の手でデータを触らなくても、捕捉されたトラフィックが保存・再利用される
  - ボトルネックを特定のため、保存されたユーザートラフィックを必要に応じてリプレイ

# GoReplayを使用したトラフィックの捕捉



# GoReplayとの経験

- 野心的なアプローチではあったが、本番のトラフィックを完全に複製する際に様々なハードルがあった
  - ユーザ情報をマスキングするためのロジック
  - マスキングされたユーザのoauthサービスの設定
  - UUIDのGET/PUTリクエストの処理

# ボトルネックはどこ？

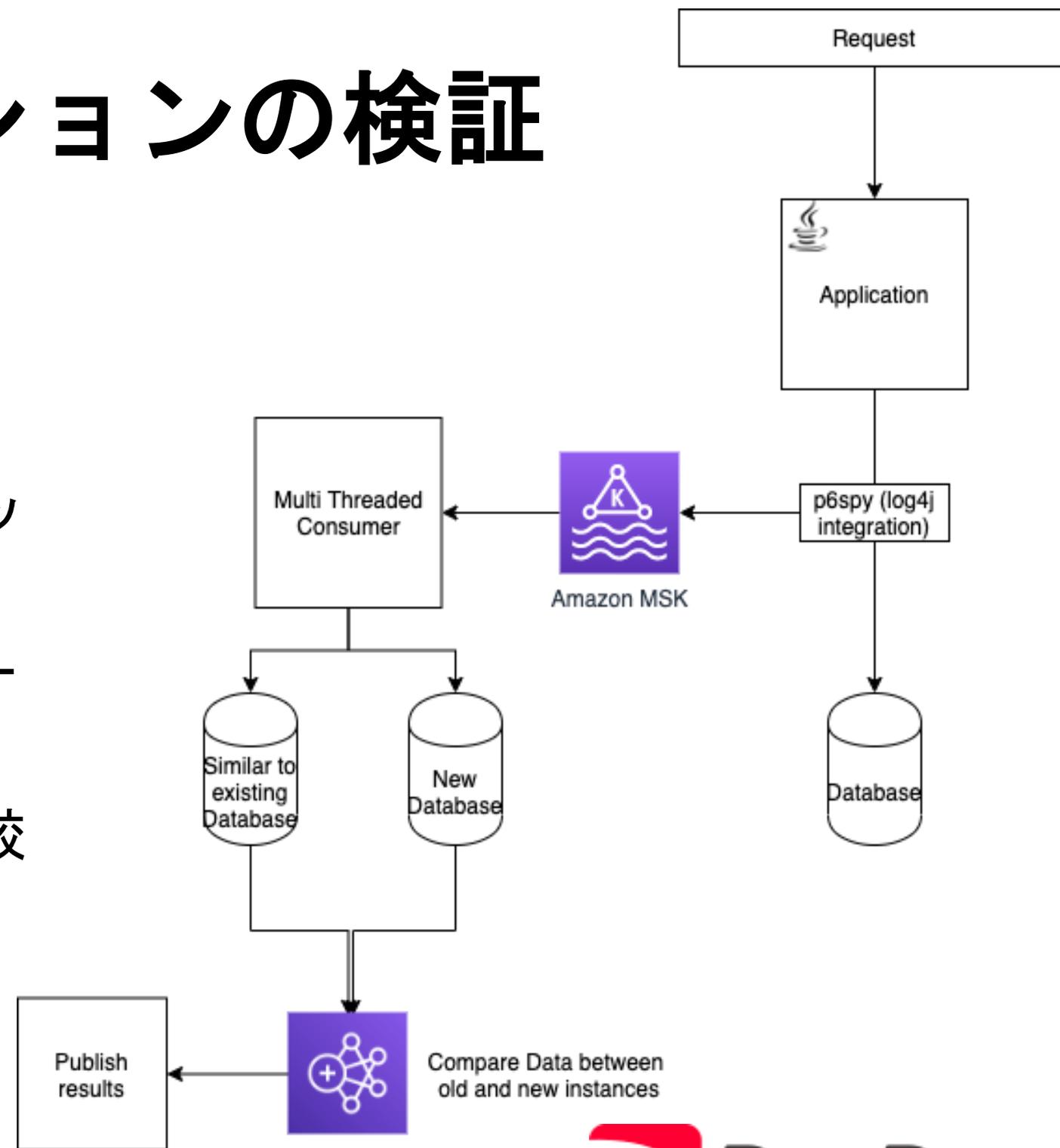
- アプリケーションとインフラの観測性の向上
- カスタムメトリックの導入による、アプリケーション監視のさらなる向上
- JMeterやGatlingなどの負荷テストツールの使用
- インフラのボトルネックを見つけるために自由に使えるAWSツール
  - Amazon CloudWatchのメトリクス
  - Amazon Auroraの Performance Insights
  - 複数のツールとの容易なインテグレーション

# 最適なソリューションとは？

- AWSは、インフラをon-the-flyで拡張・設定するための強力な機能を提供
- 改善されたかどうかの簡易検証
  - 本番リリース前にパフォーマンス環境で新しいアーキテクチャをテスト
  - 要件を満たす他のAWS Managed Resource (Super Reliable)を試してみる

# データベースソリューションの検証

- GoReplayと同様のアプローチ
- P6SPYを使用したデータベーストラフィックの再生
- 同じワークロードをさまざまなデータベースに容易に実行
- カスタムEMRジョブを使用したデータ比較



# 大容量データ向けAWSソリューション

- PayPayのデータ量は、事業成長に伴い何倍(1つDBにつき数十TB)にも増加している
- データの調整と検証
  - データ量の増加に伴う課題
  - アーキテクチャやプラットフォームの変更に対する信頼性を確保するために必要
    - 検証用のEMRストリームとDynamoDBストリーム
      - 強力でありながらセットアップが容易
      - アプリケーションとの容易なインテグレーション

# PayPayでのシステム移行

Ayush Mittal

# PayPay プロダクト部門 Ayush Mittal

- インド出身
- 職歴:
  - Paytm India (2015年～2019年)
  - PayPayプロジェクト (2019年～現在)



# 目次

- **PayPay利用レポート**

- 古いアーキテクチャ
- 課題
- 解決策
- システム移行

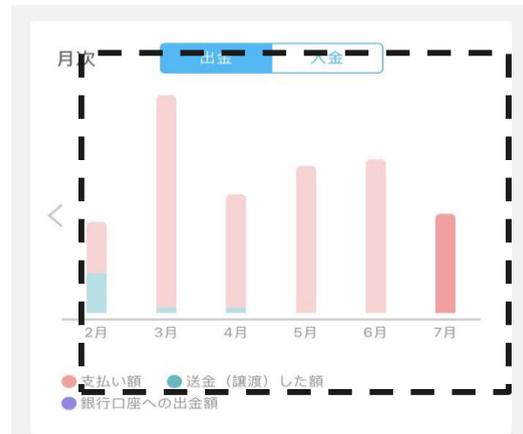
- **データ突合処理**

- 古いアーキテクチャ
- 新しいアーキテクチャ

# PayPay利用レポート



キャッシュバックの概要



月次の出金総額



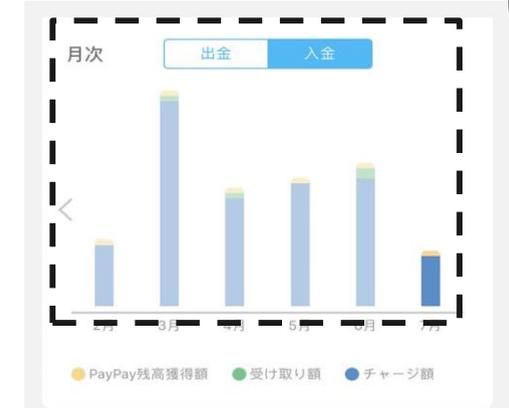
ローソンをはじめとする主要店舗



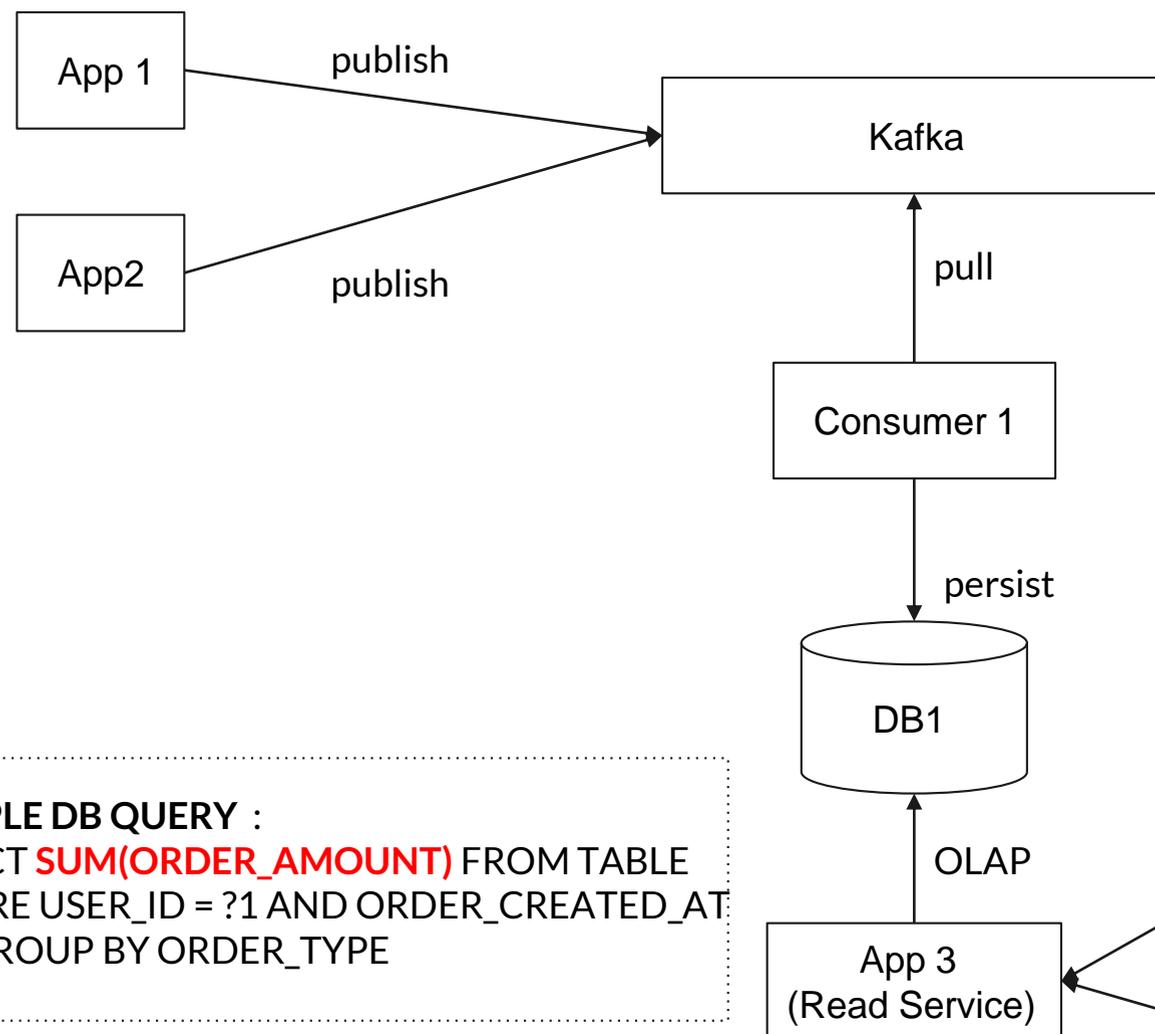
入金内訳



月次の入金総額



# 古いアーキテクチャ



SAMPLE DB QUERY :  
SELECT **SUM(ORDER\_AMOUNT)** FROM TABLE  
WHERE USER\_ID = ?1 AND ORDER\_CREATED\_AT  
<?2 GROUP BY ORDER\_TYPE

通常ユーザ (< 10k txns)

1. getデータを取得できる
2. OLAPクエリの方が時間はかからないが、毎回実行される

ホットユーザー (~100 K txns)

1. 大量のOLAPクエリ
2. すべてのリクエストで aggregation を毎回実行するため、dbリソースを消費する

# 課題

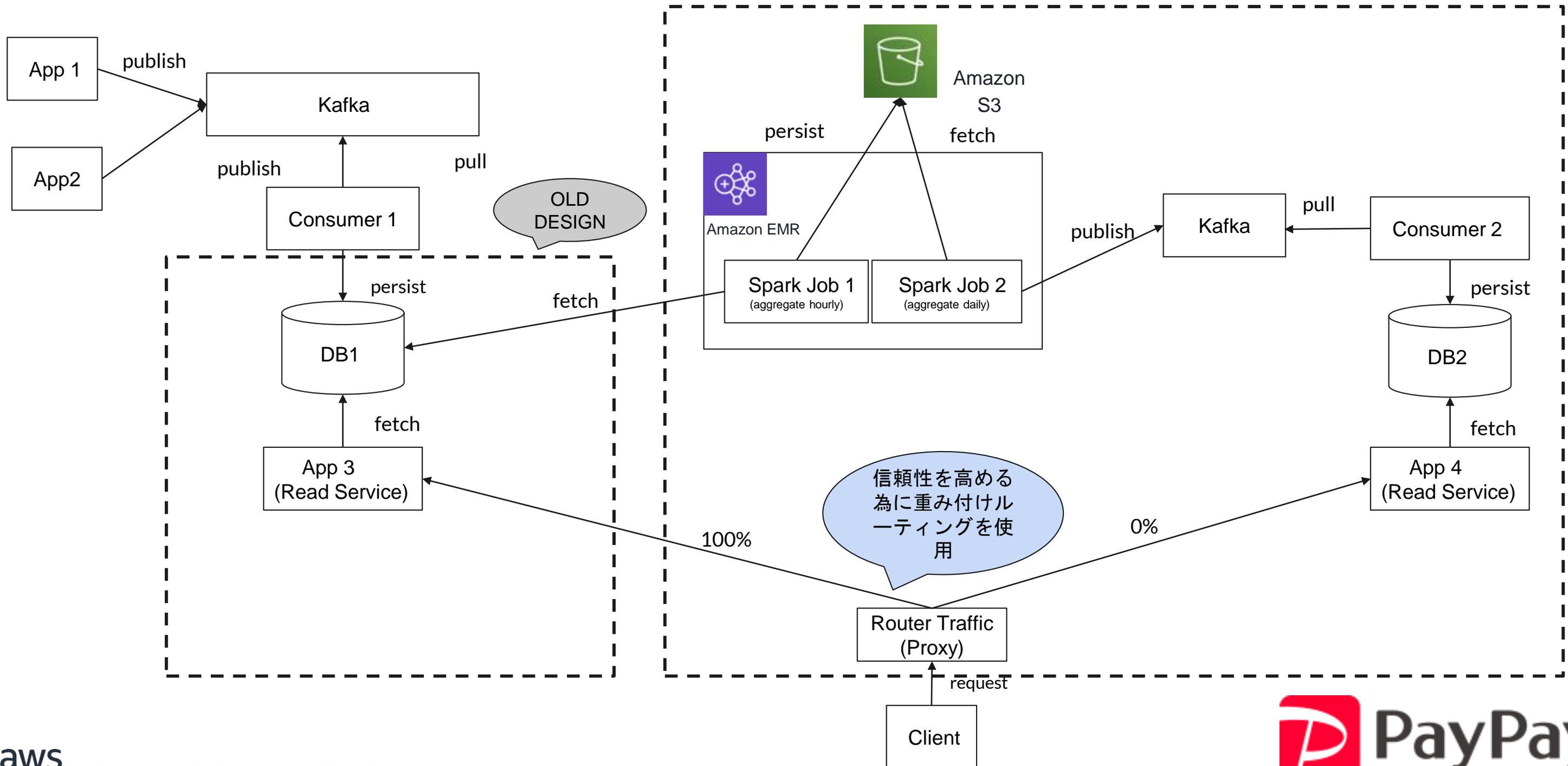
- 過去のデータサマリー/計算値は変わらないのに、各要求に対して毎回複数の重いOLAPクエリを実行している
- ホットユーザーの場合、スキャンされる行数が多いため、mysqlスレッドがビジー状態になり、読み取りがタイムアウトしてしまい、UXに影響を与える

# 解決策

- 前日のデータを**集計し、別のデータストアへ保存**することでクエリ頻度を下げる
- 増加するトランザクション数と、ホットユーザーに対応する**分散集約システム**を設計する。(EMRは最適なソリューションである)
- 中間データの保存には、一貫性のある永続的なストレージが必要である。(S3が良い。HDFSは一貫性という点で劣る。またS3はデバッグのためにデータを参照するためにも使うことができる。)
- **イベント駆動型モデル**を使用して、障害発生時にデータを再度集約できるようにし、再度フェッチするのにかかるDBコストを節約する。(Pub Subモデルとデータを使用して、ブローカ障害が発生した場合にS3から再度パブリッシュすることが可能)
- 信頼性を高め、ロールバック戦略として機能するように、トラフィック (1% ... 5% ... 10% ..... 100%) を新しいシステムへ**重み付けルーティングするルータシステム**を設計する。

# システム移行-1

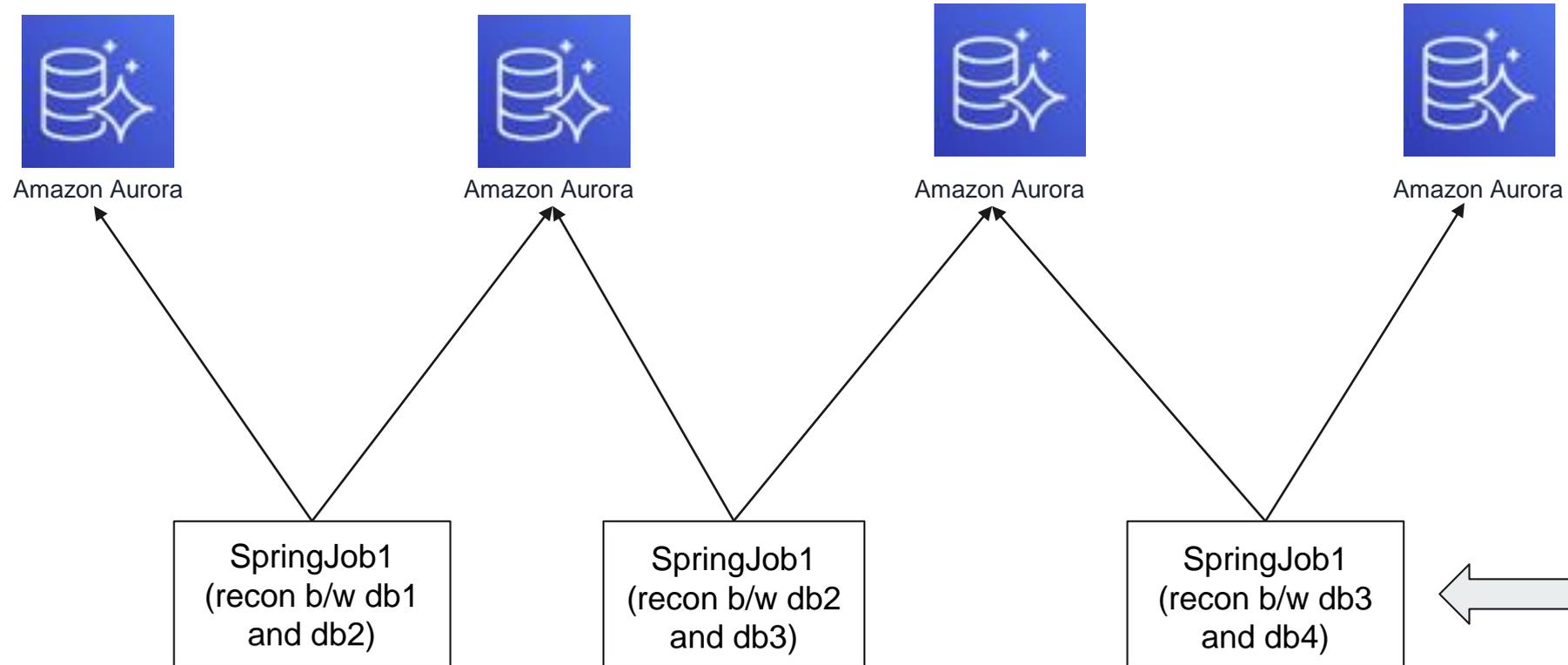
NEW DESIGN



# データ突合処理

- PayPayはイベントベースのモデルアーキテクチャを使用し、マイクロサービスのほとんどがメッセージに依存して対応するドメインロジックを実装している。
- 一貫性を維持するためには各マイクロサービスの支払い順序が常に更新されている必要があり、それには突合が非常に重要となる。

# 古いアーキテクチャ

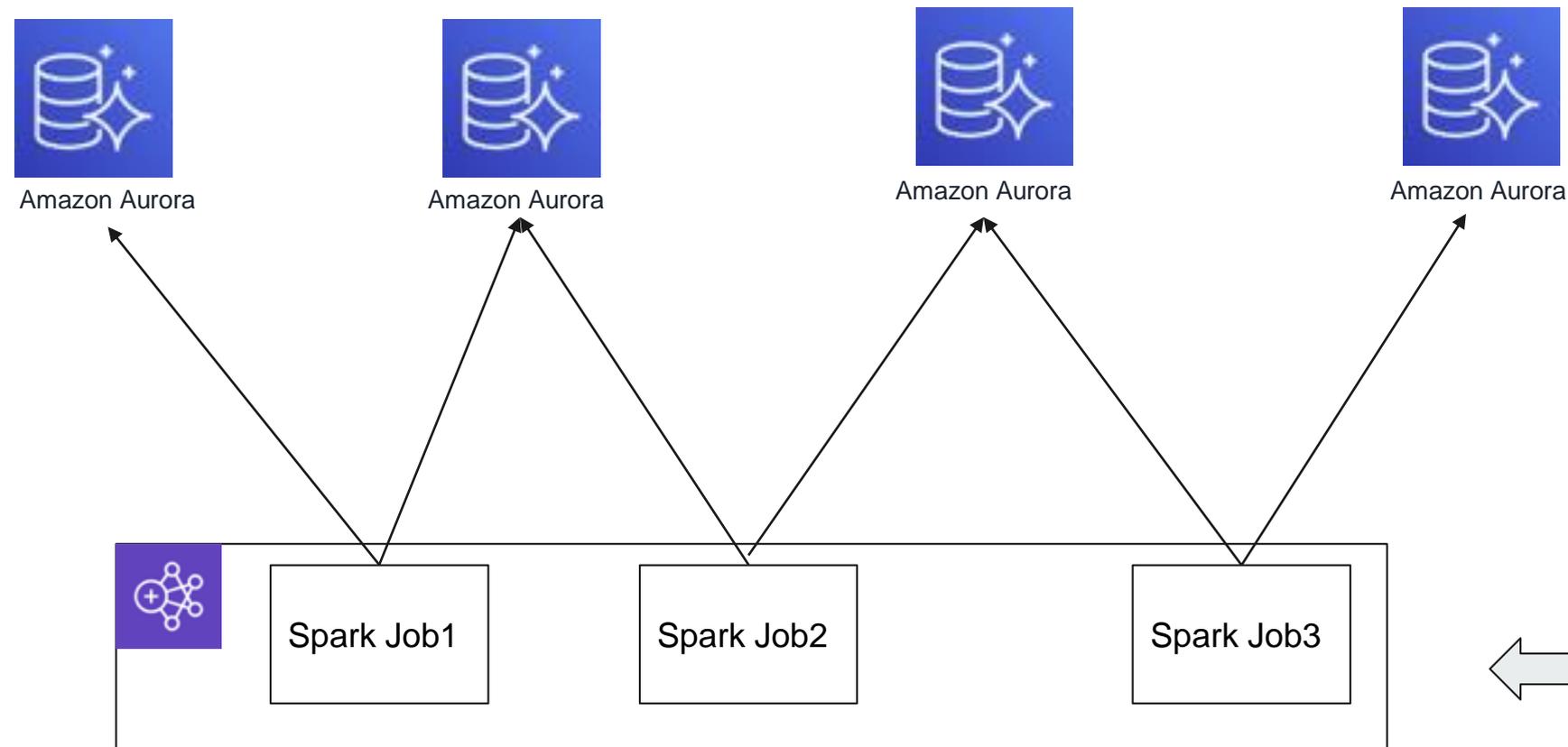


## 問題:

1. シリアル化された処理
2. 突合をするのがとても遅い
3. データベースのパフォーマンスとアプリケーションを維持するために、スキャンする必要がある行の数が少ない



# 新しいアーキテクチャ



改善点:

1. 分散処理
2. 大量のデータをスキャンできる
3. DBパフォーマンスの向上
4. AWSは、独自のスパーククラスタを維持するために、管理するプラットフォームタスクが非常に少なくなった

# Amazon Elasticsearch Service を利用したログ基盤

Robert Kuria

## [経歴]

- ケニア出身
- 日本の高等教育、文部科学省奨学金 (2008~)
- 電気電子工学学士号



## [職歴]

- Infraエンジニア @SoftBank Corp:2014~2018
- プラットフォームエンジニア @PayPayプロジェクト:2018~現在

# 目次

- **主なログの種類**
- **ログETLの概要**
- **クラスタとインデックスのサイズ**
- **課題**
- **Amazon Elasticsearch UltraWarm**
- **今後のESの取り組み**

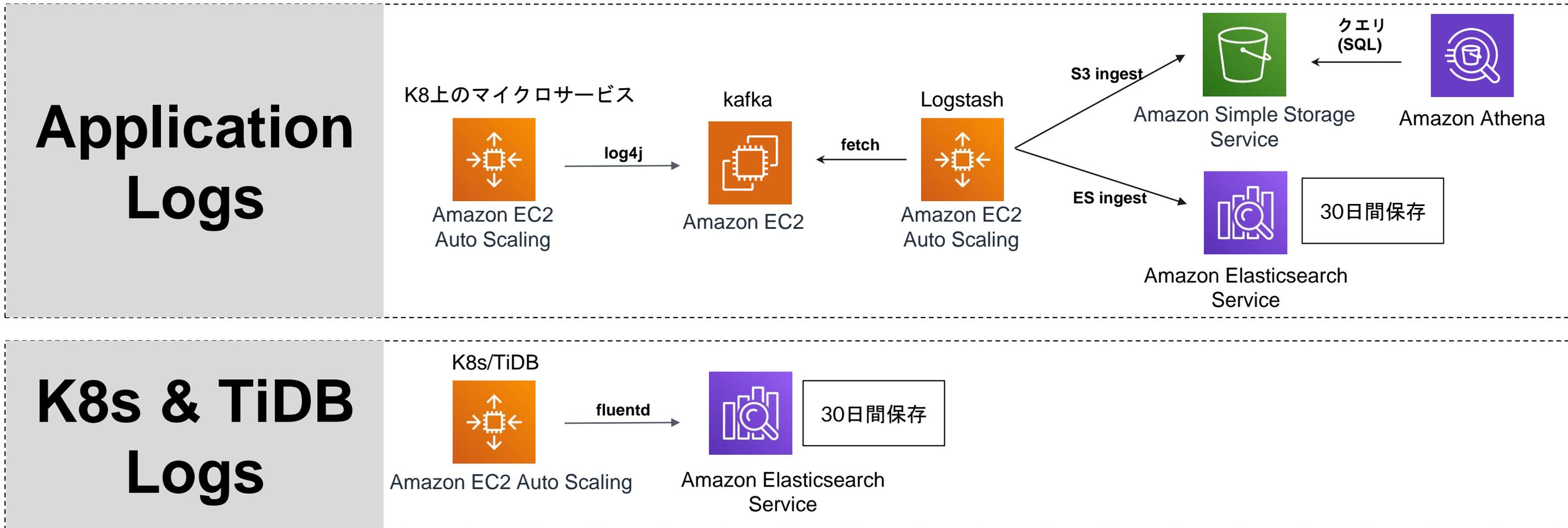
# 主なログの種類

**Application  
Logs**

**K8s Cluster  
Logs**

**TiDB  
Logs**

# Log ETLの概要



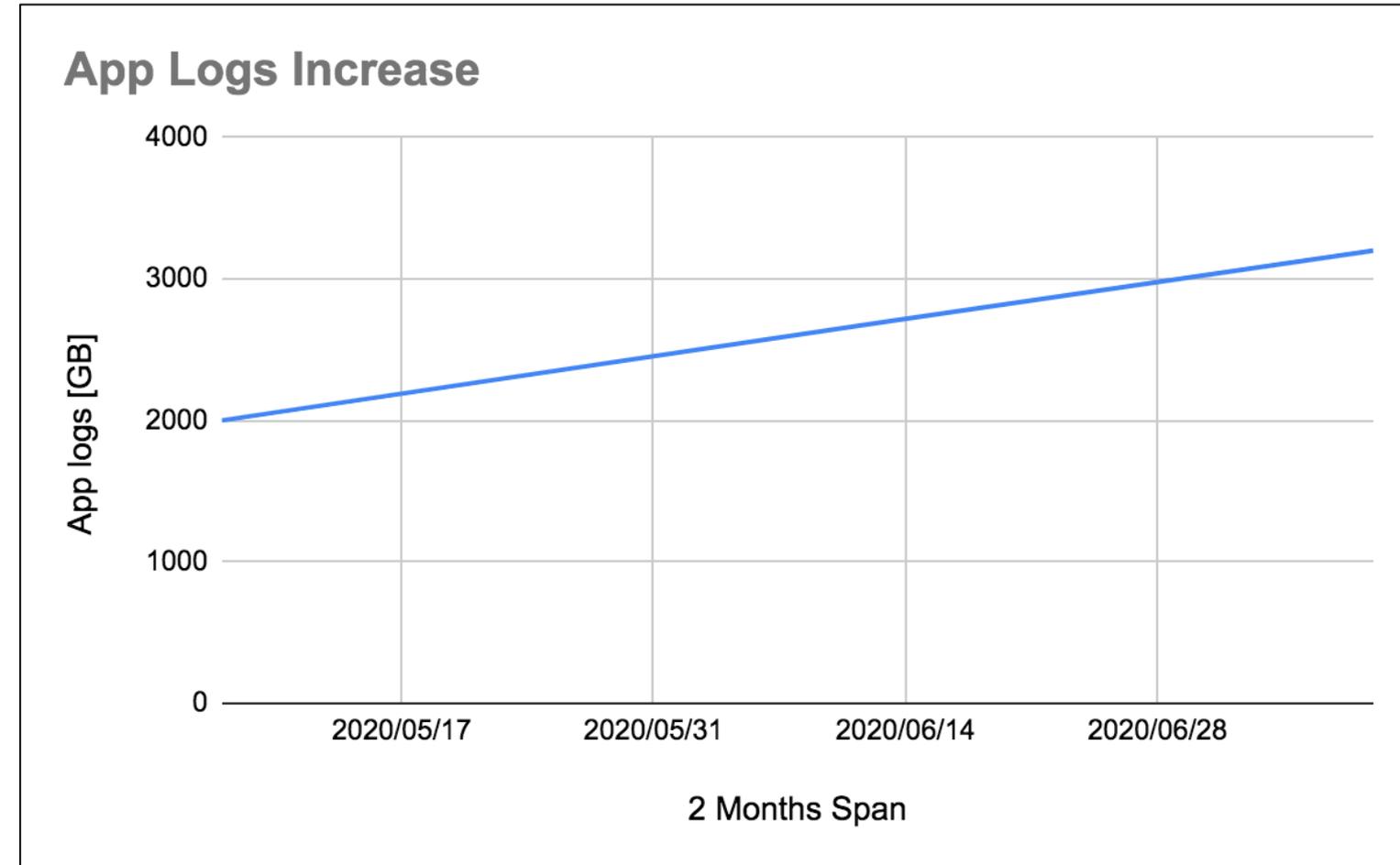
- ・アプリケーションログはビジネスサポートにも使用されるため、SLAが高く、アーカイブ検索も利用可能
- ・主に運用、システム監視、およびトラブルシューティングのためのK8sおよびTiDBログ

# Amazon ES クラスタとインデックスのサイズ

- **アプリケーション&K8sログクラスタ (共有)**
  - 120以上のマイクロサービスをログ連携
  - 57ノード(3つのマスターノード、54のデータノード)
  - 329 TBのクラスタ容量
  - アプリケーションログインデックス:約2.8 TB/日(レプリカ数1なので実質5.6TB)
  - K8sログインデックス:約500 GB/日
  
- **TiDBログクラスタ (専用)**
  - 1つのTiDBクラスタからのログ
  - 9ノードクラスタサイズ(3つのマスターノード、6つのデータノード)
  - 36 TBの総容
  - TiDBログインデックス:約100 GB/日(レプリカ数1なので実質200GB)

# 課題

- **ログ量の増加**
  - 書き込みを分散するための頻繁なインデックスのチューニング
  - ログ管理コストの増加
  - ES内ログ保存期間の短縮
  - 構成の変更が困難
- **AthenaとKibanaで異なるログ検索の経験**
  - Athenaは全文検索エンジンではないため、Kibanaと同じように使うと効率的ではない
- **ドメイン構成の変更/BGデプロイメント時のダウンタイムの増加**
  - Kibanaダウンタイムが長くなる

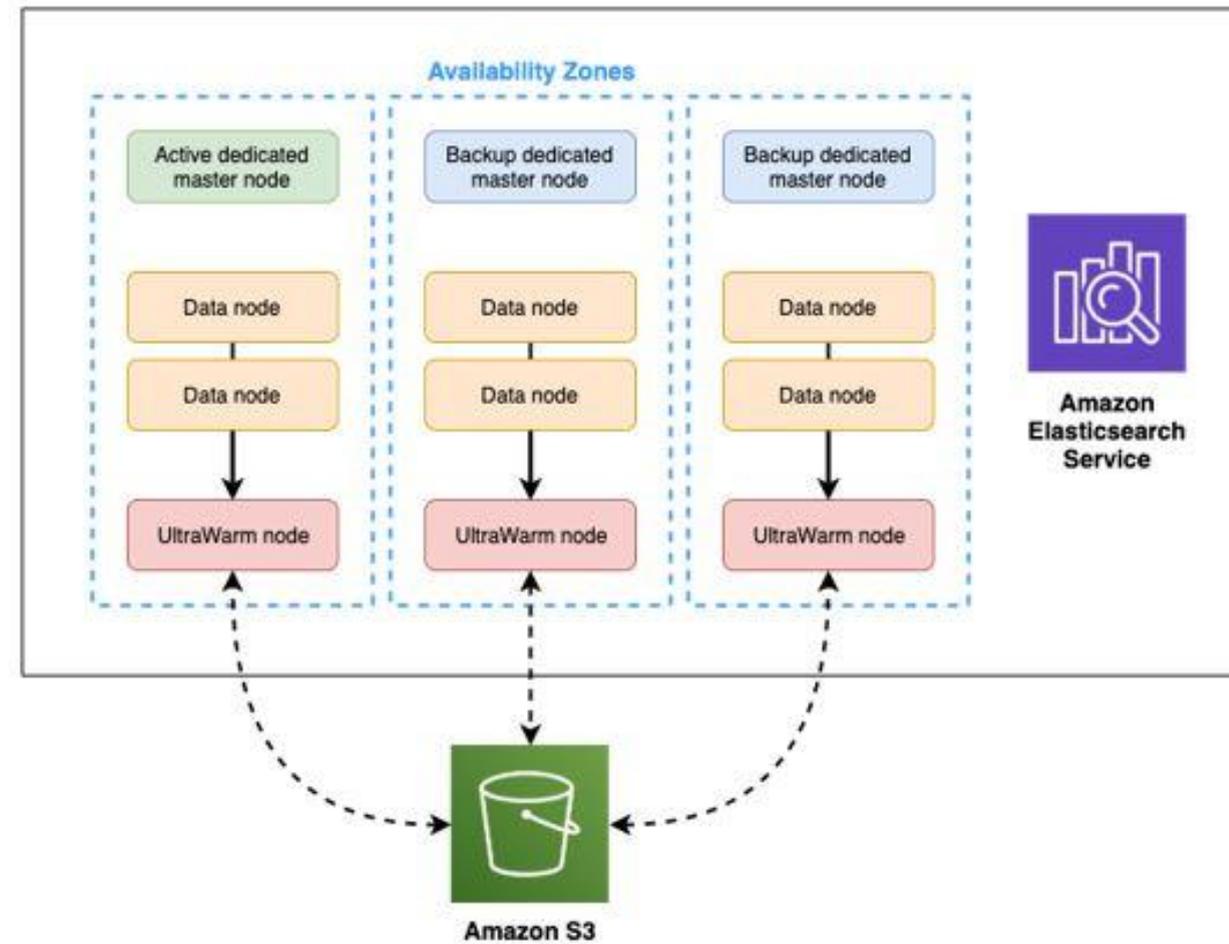


# Amazon Elasticsearch Service UltraWarm

## UltraWarmとは?

Amazon Elasticsearch Serviceの新しい読み取り専用Warmストレージティア、2020年5月5日から東京リージョンで利用可能

- **大容量ノード**  
ultrawarm1.large.elasticsearch:20 TB
- **階層間でのインデックスの移動が容易**  
`POST_ultrawarm/migration/my-index/_warm`  
`POST_ultrawarm/migration/my-index/_hot`
- **インデックス階層の移行の自動化**  
Index State Management(ISM)



UltraWarmは日々急激に増加するログボリュームの  
パーフェクトなソリューション

# UltraWarm

## UltraWarm導入前

- 57クラスタノード
- Kibanaで最大1ヶ月検索可能

## UltraWarm導入後

- **20クラスタノード**  
マスターノード:3  
Hot Tier:6  
Warm Tier:11
- **Kibanaで3ヶ月検索可能に**  
Hot Tier:1週間  
Warm Tier:3ヶ月

月々のコストを  
約40%削減

# 今後のESの取り組み

- Athenaクエリのパフォーマンスを向上させるために、ファイル形式をtextからParquetに変更する
- OktaとAmazon CognitoでKibanaへのSSOを実現する
- Amazon ESの新バージョンと新機能を試す
- ログドキュメントを最適化する (explicit mapping)

最後に



# We are hiring

- フロントエンドエンジニア
- バックエンドエンジニア
- Androidエンジニア
- iOS エンジニア
- QA Engineer
- Data Engineer
- SRE / Platform
- Product Security Engineer
- QA マニュアルテスト マネージャー
- DBA
- エンタープライズシステム開発  
PM/PMO
- 不正対策エンジニア
- セキュリティエンジニア
- 業務推進エンジニア
- プロダクトデザイナー

20カ国以上から集まった200名以上のメンバー



# We are hiring



採用ページはこちらから



<https://about.paypay.ne.jp/career/>

For open positions contact: [momoko.hayakawa@paypay-corp.co.jp](mailto:momoko.hayakawa@paypay-corp.co.jp)



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



ご清聴ありがとうございます