



アーキテクチャ図 ゲーム技術者

AWS でのモバイルアプリ内購入の検証



AWS でのモバイルアプリ内購入の検証: アーキテクチャ図 ゲーム技術者

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

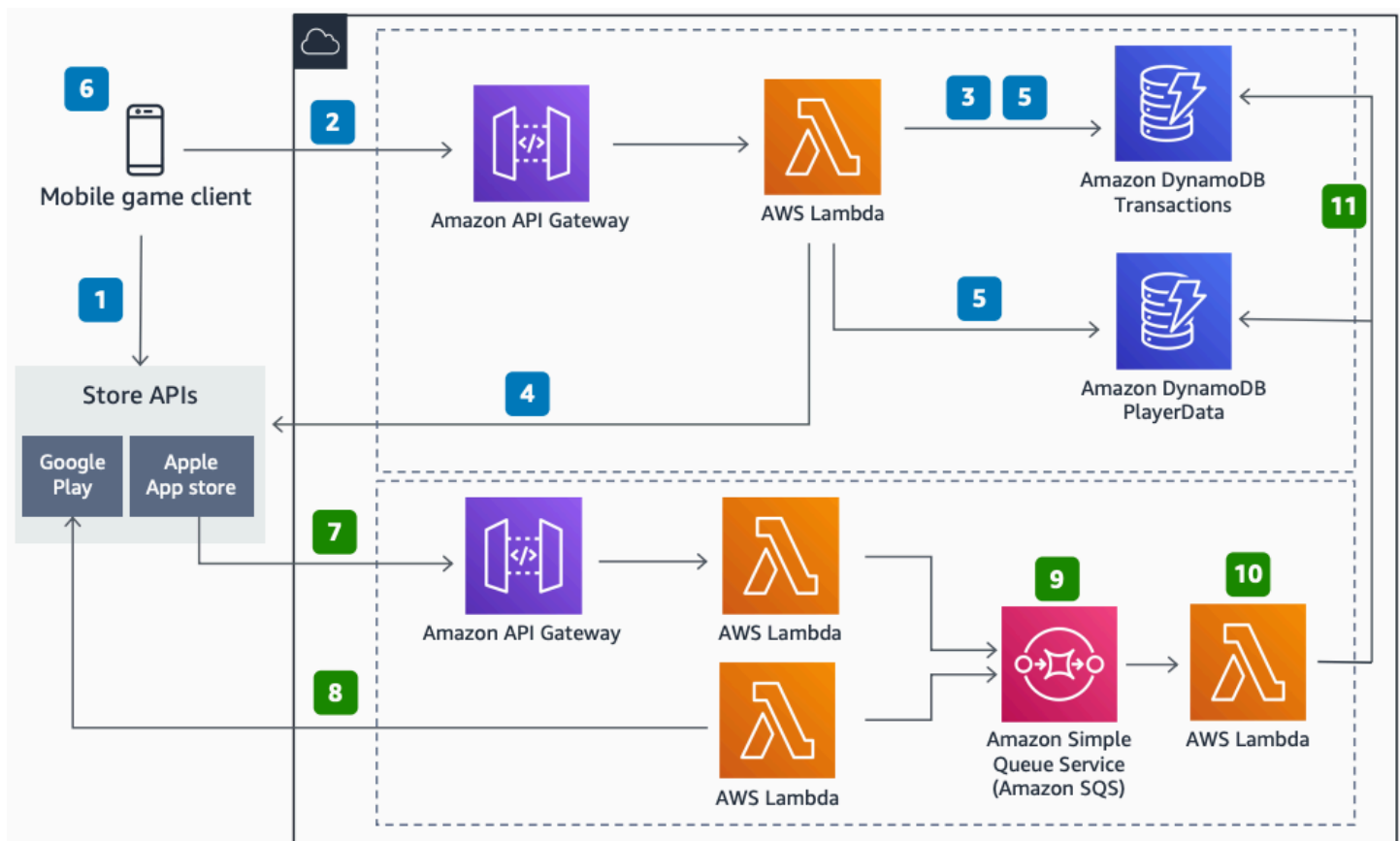
アプリ内購入の検証	i
AWS 図表のモバイルアプリ内購入の検証	1
詳細情報	2
図の履歴	2
.....	iv

でのモバイルアプリ内購入の検証 AWS

公開日: 2021 年 1 月 1 日 ([図の履歴](#))

このアーキテクチャは、Google Play Store および からのアプリ内購入を検証しApple App Store、返金を管理するためのサーバーレスソリューションを提供します。このアーキテクチャでは、[Amazon API Gateway](#)[AWS Lambda](#)、および [Amazon DynamoDB](#) を使用して、トランザクションを大規模に安全に処理します。

AWS 図表のモバイルアプリ内購入の検証



次のステップでは、購入検証フローについて説明します。

1. モバイルゲームクライアントは、OS 固有の SDK を使用して購入を行い、受信をローカルに保存します。
2. ゲームクライアントは API Gateway に API リクエストを行い、受信を検証して購入したアイテムを受け取ります。

3. Lambda 関数は、DynamoDB のトランザクションテーブルをチェックして、受信が以前に使用されていないことを確認します。
4. Lambda 関数は、Google Play Storeまたは Apple App Store API を通じて受信を検証します。
5. 検証後、Lambda 関数は DynamoDB の PlayerData テーブルに項目を追加し、追加のメタデータを含む受信をトランザクションテーブルに追加します。
6. クライアントは成功メッセージを受信し、購入した項目データをローカルで同期して、ローカル受信コピーを削除します。

次の手順では、返金フローについて説明します。

7. API Gateway は、iOS で返金されたトランザクションに関する通知を受け取ります。スケジュールされた Lambda 関数は、Android で返金されたトランザクションをクエリします。
8. 返金されたトランザクションは、Lambda 関数によって [Amazon Simple Queue Service](#) キューにプッシュされます。
9. Lambda 関数は、プレイヤーアカウントの閉鎖などの追加のアクションを含む返金を処理します。
10. Lambda 関数は DynamoDB テーブルを更新して項目を削除し、トランザクションを返金済みとしてマークします。

詳細情報

詳細については、次のリソースを参照してください。

- [AWS アーキテクチャアイコン](#)
- [AWS アーキテクチャセンター](#)
- [AWS Well-Architected](#)

図の履歴

このリファレンスアーキテクチャ図の更新について通知を受け取るには、RSS フィードをサブスクライブします。

変更

説明

日付

初版発行

リファレンスアーキテクチャ 2021 年 1 月 1 日
図が最初に公開されました。

Note

RSS 更新をサブスクライブするには、使用しているブラウザで RSS プラグインが有効になっている必要があります。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。