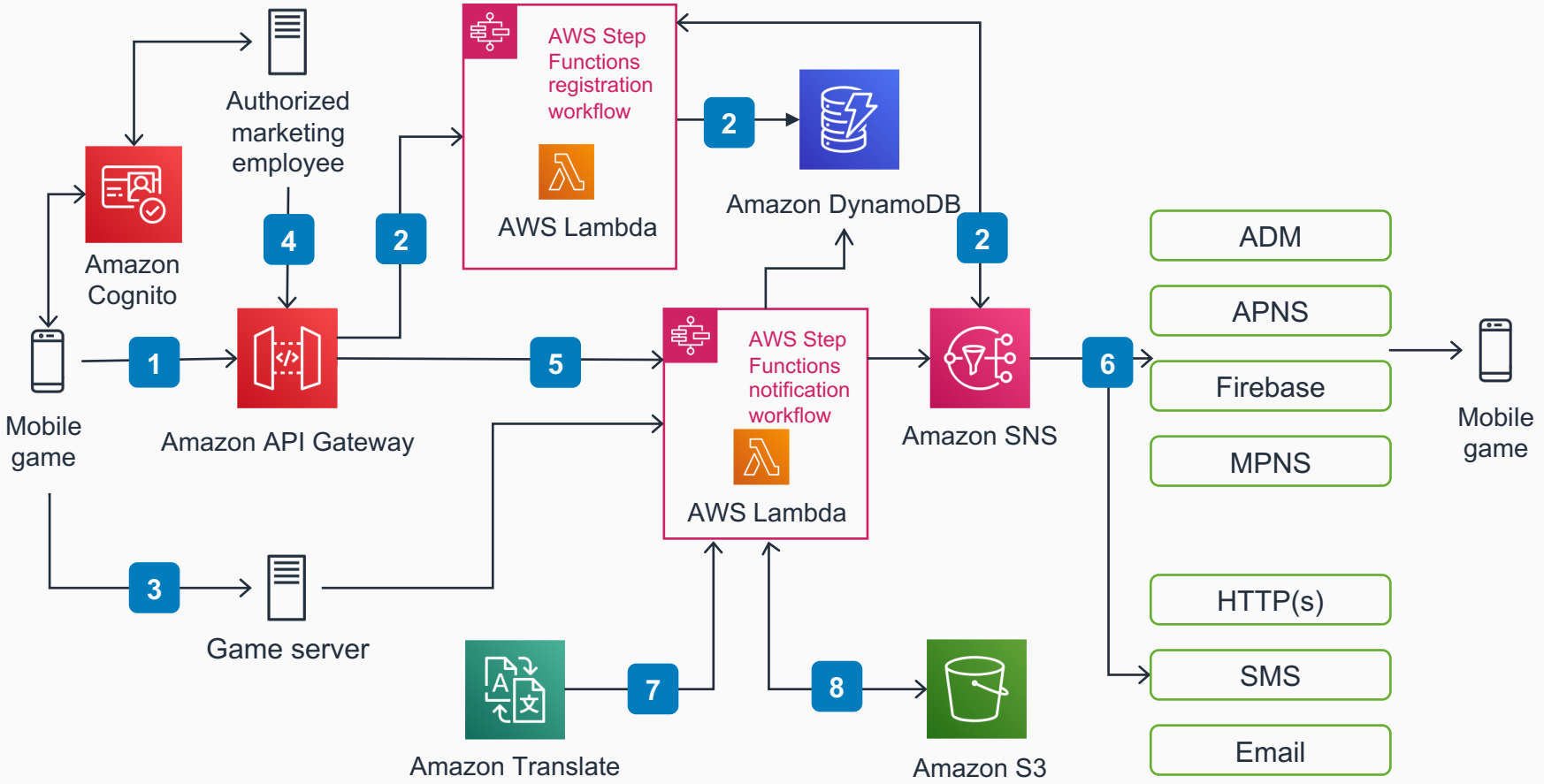
Serverless Notifications for Mobile Games

Build a Scalable Pipeline with Simultaneous Transaction Processing

Create a serverless data flow to ingest, store, process, and perform push notifications to subscribers. Use this pipeline for flash offers, in-game event/promotion notifications, campaign specific notifications targeting segmented audience with efficiency tracking and processing, and storage of notifications data for further analysis. The services shown in this architecture are scalable and support processing large volumes of transactions simultaneously.



1. When the game client launches, it sends registration information to an **Amazon API Gateway** REST endpoint. The game client can also send direct notification information via REST.
2. **Amazon API Gateway** requests step functions registration flow that performs additional logic like device registration and storing **Amazon SNS** endpoint data in **Amazon DynamoDB**.
3. As game client sends information to game server, the server triggers the **AWS Step Functions** workflow that processes messages.
4. To run campaigns or send single notifications, marketing employee can also manually request **AWS Lambda** step functions flow via **Amazon API Gateway**. Authorization is done via **AWS Cognito**.
5. Notification flow performs actions like translation, message payload enrichment, and storing and obtaining endpoint data from Amazon DynamoDB. This data is used to define direct users or segments to send messages to.
6. **Amazon SNS** uses push notifications service via provided credentials and mobile device tokens. After receiving the message, all topic subscribers or direct clients receive the push notification message via providers like Firebase or Apple push notification service.
7. **AWS Step Functions** flow uses **Amazon Translate** for quick translation.
8. Raw data is sent to **Amazon S3** for cold data analytics.

AWS Reference Architecture