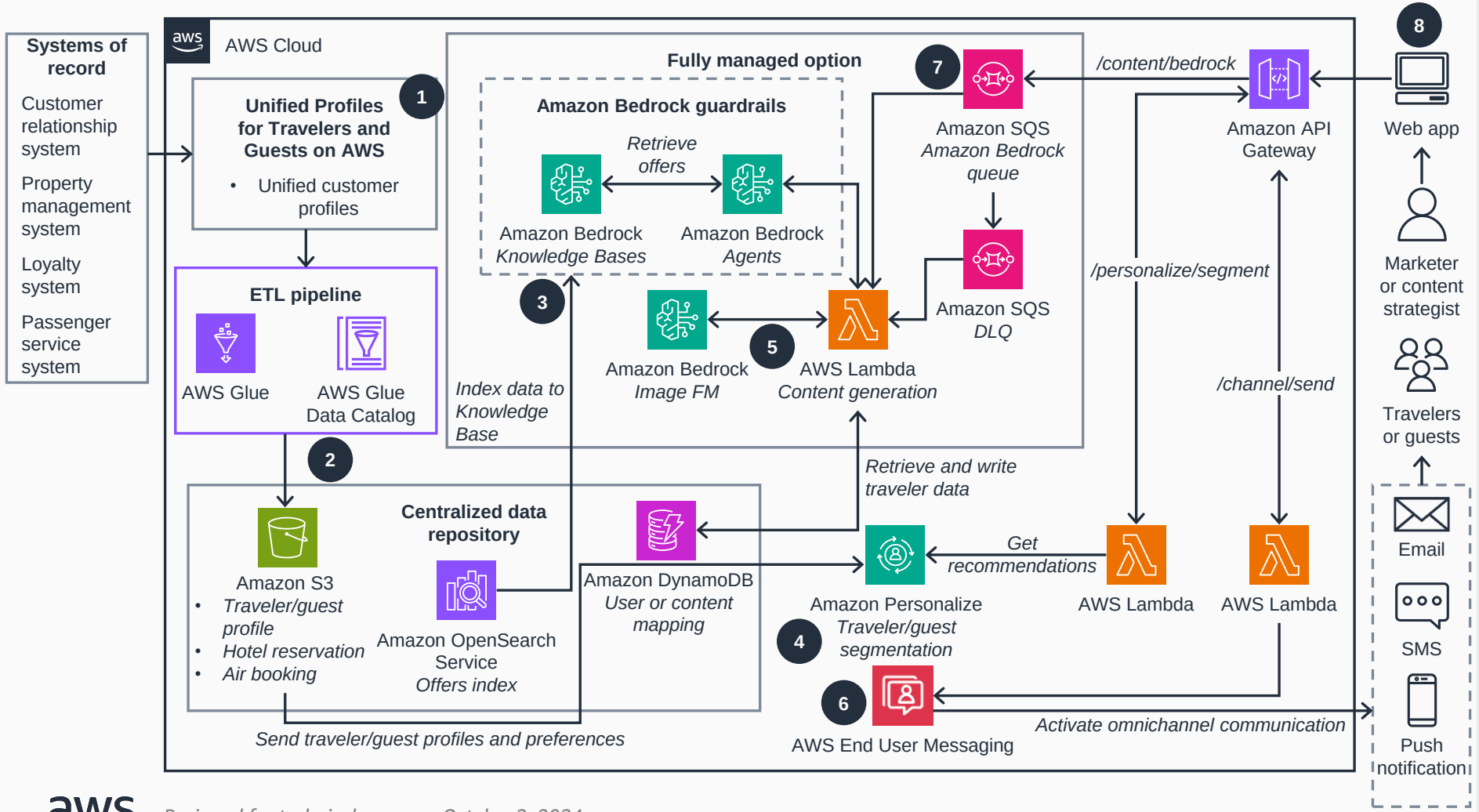
Guidance for Hyperpersonalized Marketing with Amazon Personalize and Amazon Bedrock

This architecture diagram shows how to implement a comprehensive travel and hospitality (T&H) customer engagement platform that uses Unified Profiles for Travelers and Guests on AWS to generate hyperpersonalized marketing content.



1. Using connectors, Unified Profiles for Travelers and Guests on AWS ingests booking, stay, and loyalty data from industry systems and creates unified profiles of travelers or guests.
2. Catalog traveler/guest metadata using **AWS Glue Data Catalog**. Use **AWS Glue** to transform this data from JSON to comma-separated values (CSV) and store it in an **Amazon Simple Storage Service (Amazon S3)** bucket.
3. Your marketing team indexes the promotion and offers data in **Knowledge Bases for Amazon Bedrock** stored on **Amazon OpenSearch Service**.
4. **Amazon Personalize** creates a segmentation model from traveler/guest booking, stay, and loyalty data stored on **Amazon S3**.
5. Use **AWS Lambda** to call an **Amazon Bedrock** foundation model (FM) API and generate hyperpersonalized marketing content using traveler /guest preferences. Store user/content mapping in an **Amazon DynamoDB** table.
6. Activate campaigns on **AWS End User Messaging** to send tailored promotions and offers to travelers/guests using SMS, WhatsApp, email, and push notifications.
7. A **Lambda** function acts as the **Amazon API Gateway** backend, retrieving and batch processing messages from the **Amazon Simple Queue Service (Amazon SQS)** queue. **Amazon SQS** decouples the **API Gateway** endpoint from **Lambda**, providing a request buffer and preventing throttling. Batch processing improves efficiency and reduces overhead.
8. The web application, utilizing a REST API hosted on **API Gateway** with **Lambda** as the backend, enables your marketers or content strategists to access traveler/guest profiles, segmentation, and content generation services. Requests are queued in **Amazon SQS** for reliable messaging.

Reviewed for technical accuracy October 2, 2024

AWS Reference Architecture