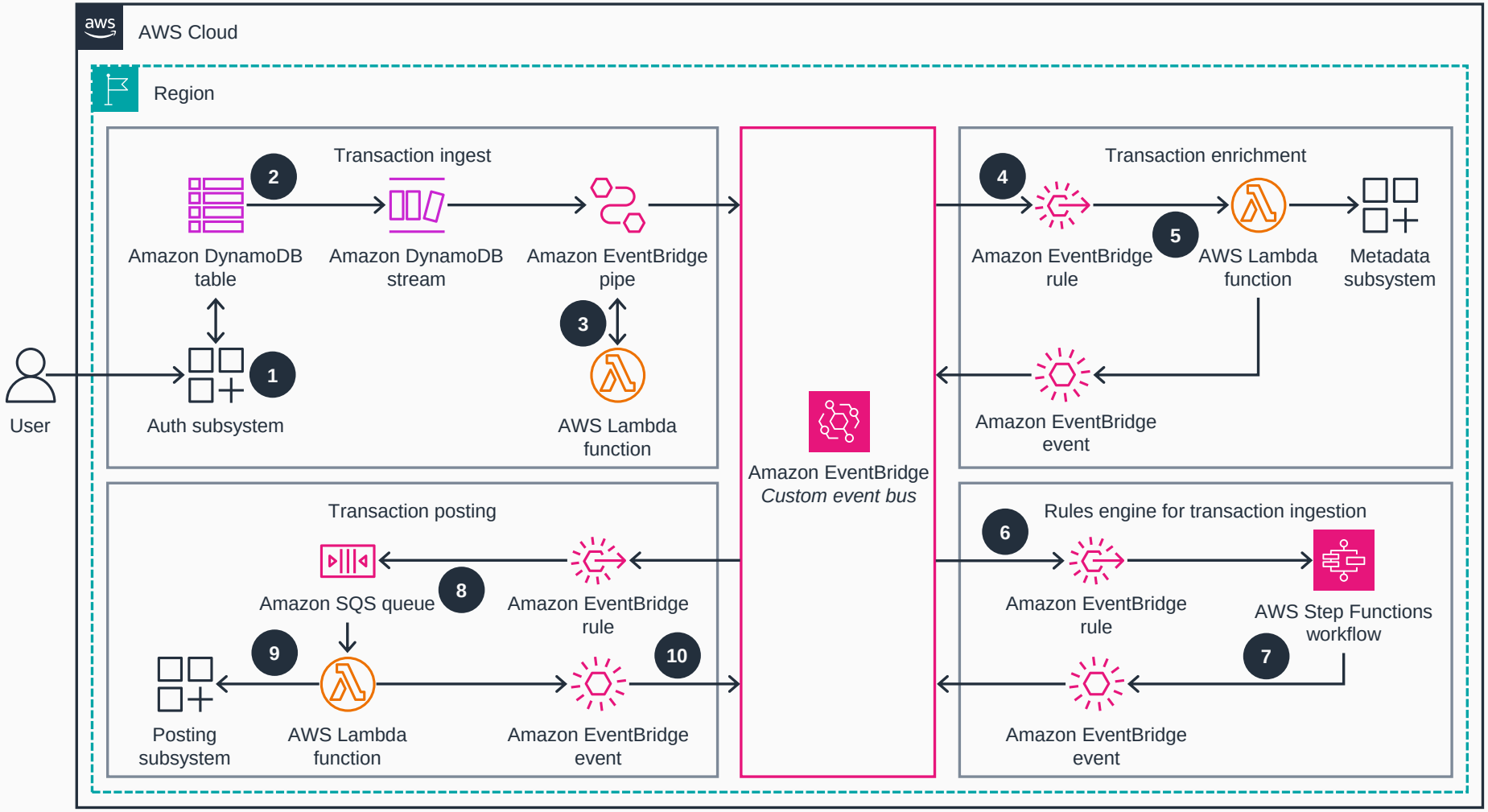
Guidance for Building Payment Systems Using Event-Driven Architecture on AWS

This architecture diagram shows how to enable near real-time processing of account posting events in payment systems.



1. Your user initiates a payment, which the authorization application approves and persists to an **Amazon DynamoDB** table.
2. An **Amazon EventBridge** pipe reads the approved authorization records from the **DynamoDB** table stream and publishes events to an **EventBridge** custom event bus.
3. You can add duplicate checking logic to the **EventBridge** pipe through an **AWS Lambda** deduplication function.
4. An **EventBridge** rule invokes an enrichment **Lambda** function for matching events to add context like account type and bank details.
5. The **Lambda** function queries the metadata and publishes a new event containing the extra info to the **EventBridge** custom event bus.
6. An **EventBridge** rule watching for enriched events invokes an **AWS Step Functions** workflow to apply business rules to the event as part of a rules engine.
7. When an event passes all business rules, the **Step Functions** workflow publishes a new event back to the **EventBridge** custom event bus.
8. An **EventBridge** rule adds a message to an **Amazon Simple Queue Service (Amazon SQS)** queue as a buffer to avoid overrunning the downstream posting subsystem.
9. A **Lambda** function reads from the **Amazon SQS** queue and invokes the downstream posting subsystem to post the transaction.
10. The **Lambda** function publishes the final event back to the **EventBridge** custom event bus.

Reviewed for technical accuracy May 23, 2024

AWS Reference Architecture