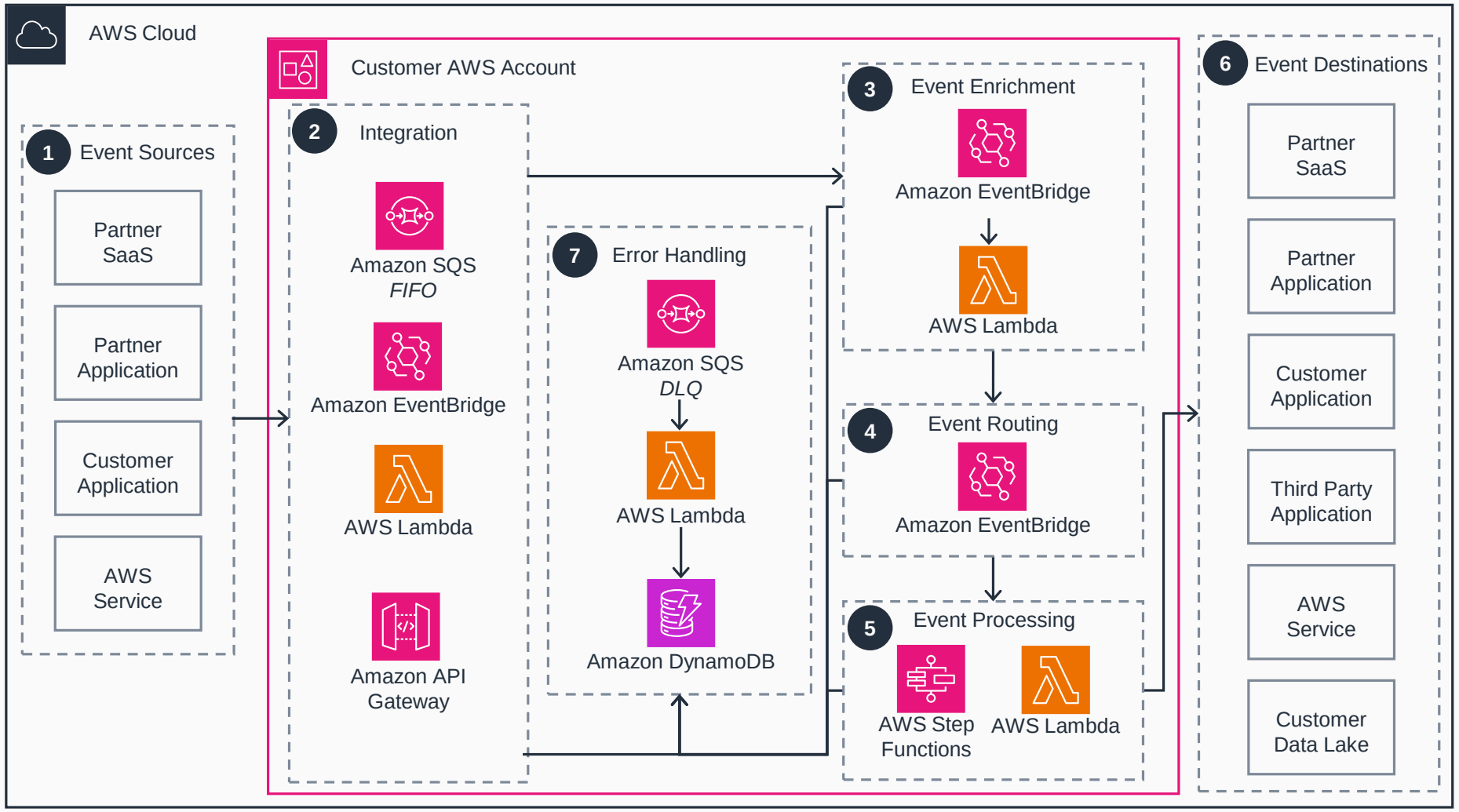
Guidance for Event-Driven Media Workflow Automation on AWS

There are six architecture diagrams that make up this Guidance. This diagram gives an overview of how to integrate event-driven third-party partner services and applications into your AWS account. The subsequent slides show how to integrate this Guidance with various third-party events.



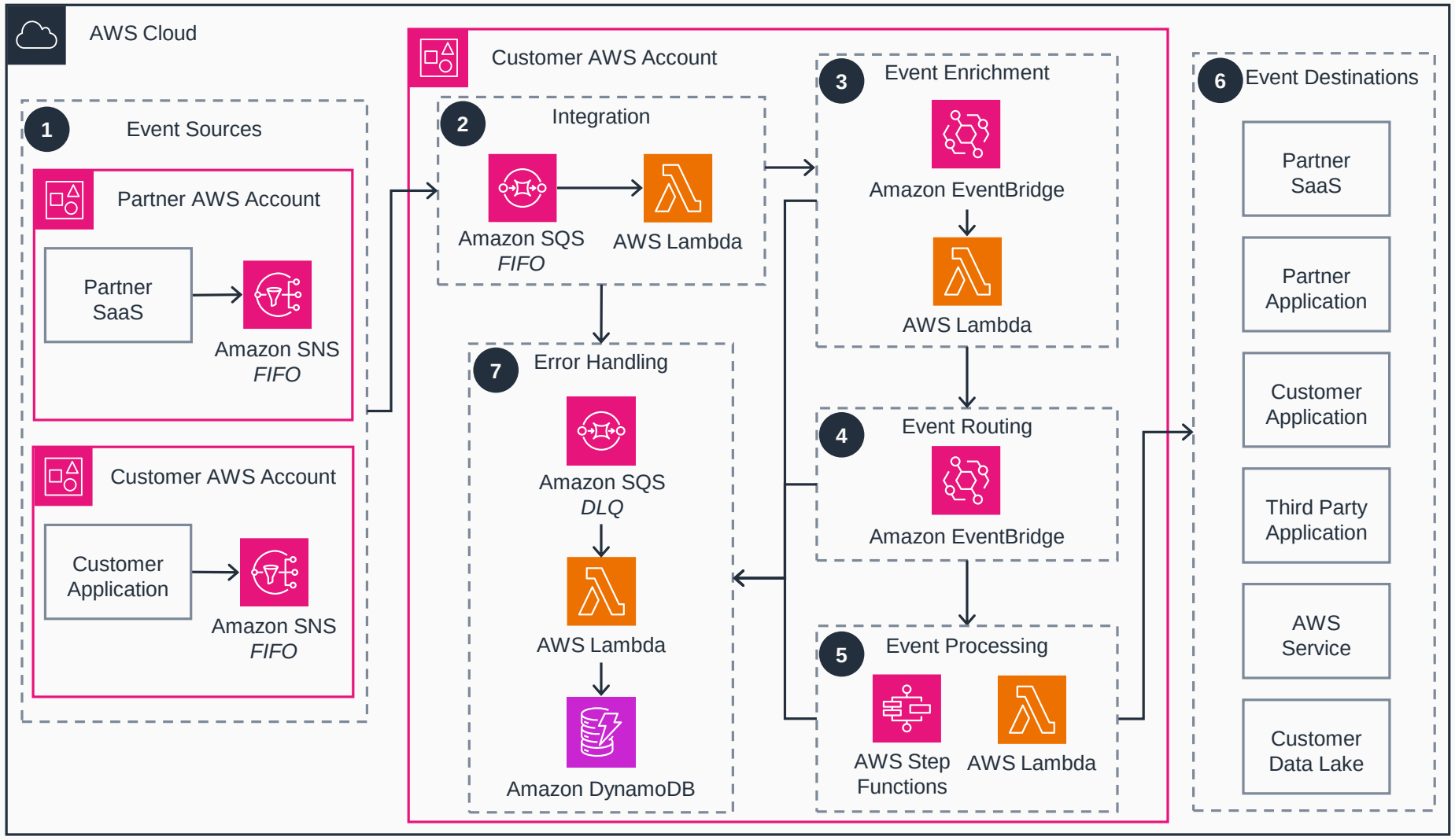
1. Partner Software as a Service (SaaS) applications, partner applications, your application, or an AWS service send events to your AWS account.
2. **Amazon Simple Queue Service** (Amazon SQS), **AWS Lambda**, **Amazon API Gateway**, or **Amazon EventBridge** receive events produced from step one.
3. **EventBridge** rules route events to **Lambda** to be enriched and transformed. Skip this step if there is no need for enrichment or transformation.
4. **EventBridge** rules route events to a destination. Rules use event pattern matching to route events based on their contents.
5. **Lambda** sends event data to downstream services. If multiple steps are required, **AWS Step Functions** starts a multi-step workflow using the event. Skip this step if there is no need for additional processing.
6. Downstream services, such as a partner SaaS, a locally deployed partner application, your application, a third-party application, an AWS service, or your data lake, receive the event data.
7. **Amazon SQS** dead-letter queue (DLQ) receives events not successfully processed from previous steps. **Amazon SQS** will re-drive and process the events. If re-driven events are unsuccessful, **Lambda** writes the event to an **Amazon DynamoDB** table for manual review.

Reviewed for technical accuracy March 212024

AWS Reference Architecture

Guidance for Event-Driven Media Workflow Automation on AWS

This diagram shows how to integrate partner events into your AWS account that are produced from an Amazon Simple Notification Service (Amazon SNS) FIFO (first in, first out) topic. These topics are delivered to a subscribed Amazon Simple Queue Service (Amazon SQS) FIFO queue.



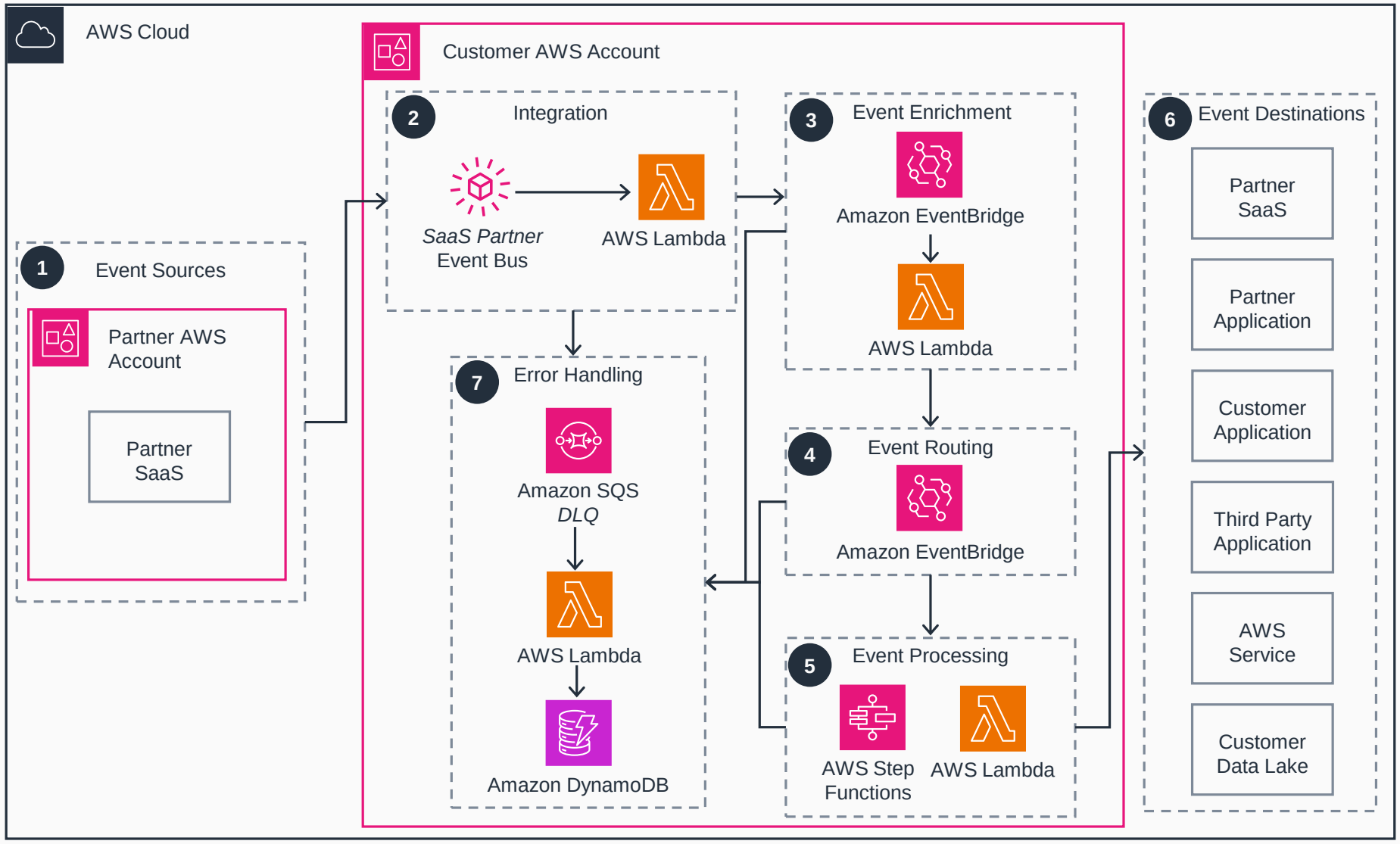
1. Partner SaaS applications, partner applications, or your application send events to your AWS account.
2. An **Amazon SQS** queue or **Amazon Simple Notification Service** (Amazon SNS) topic receives events from a third-party service. Use FIFO (first in, first out) queues with FIFO topics for ordered processing. Ensure the queue and topic types match—either FIFO or standard.
3. **EventBridge** rules route events to **Lambda** to be enriched and transformed. Skip this step if there is no need for enrichment or transformation.
4. **EventBridge** rules route events to a destination. Rules use event pattern matching to route events based on their contents.
5. **Lambda** sends event data to downstream services. If multiple steps are required, **Step Functions** starts a multi-step workflow using the event. Skip this step if there is no need for additional processing.
6. Downstream services, such as partner SaaS, a locally deployed partner application, your application, a third-party application, an AWS service, or your data lake, receive the event data.
7. **Amazon SQS** DLQ receives events not successfully processed from previous steps. **Amazon SQS** will re-driven and process the events. If re-driven events are unsuccessful, **Lambda** writes the event to a **DynamoDB** table for manual review.

Reviewed for technical accuracy March 212024

AWS Reference Architecture

Guidance for Event-Driven Media Workflow Automation on AWS

This architecture diagram shows how to integrate partner events that are produced from an Amazon EventBridge partner event source and delivered to an EventBridge Event Bus in your AWS account.



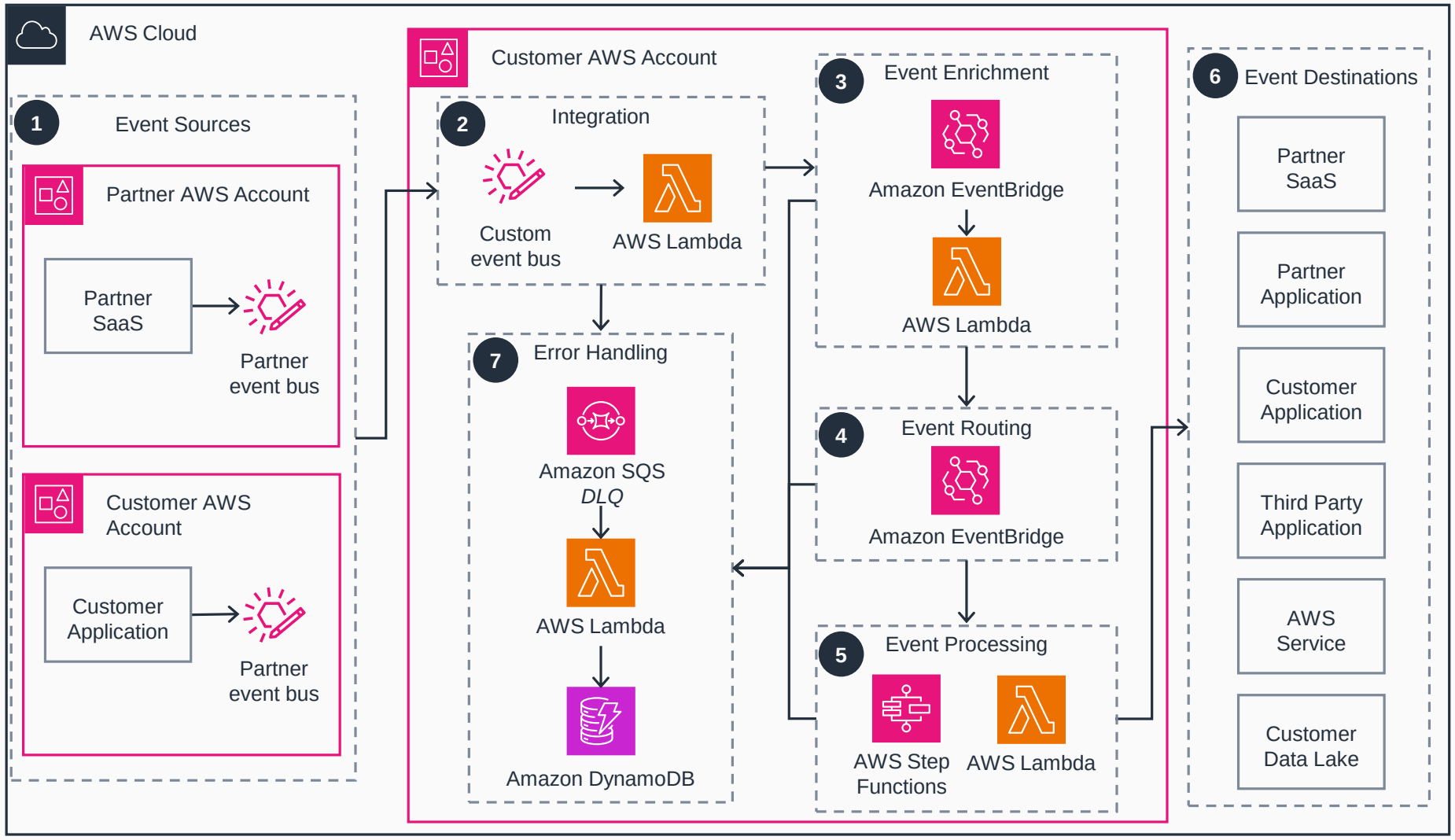
1. Partner SaaS applications send events to your AWS account.
2. The **EventBridge p**artner event source provisions a SaaS partner event bus into your account. The event bus receives partner events, and if necessary, an **EventBridge** rule routes the events to **Lambda** for transformation.
3. **EventBridge** rules route events to **Lambda** to be enriched and transformed. Skip this step if there is no need for enrichment or transformation.
4. **EventBridge** rules route events to a destination. Rules use event pattern matching to route events based on their contents.
5. **Lambda** sends event data to downstream services. If multiple steps are required, **Step Functions** starts a multi-step workflow using the event. Skip this step if there is no need for additional processing.
6. Downstream services, such as partner SaaS, a locally deployed partner application, your application, a third-party application, an AWS service, or your data lake, receive the event data.
7. **Amazon SQS** DLQ receives events not successfully processed from previous steps. **Amazon SQS** will re-driven and process the events. If re-driven events are unsuccessful, **Lambda** writes the event to a **DynamoDB** table for manual review.

Reviewed for technical accuracy March 212024

AWS Reference Architecture

Guidance for Event-Driven Media Workflow Automation on AWS

This architecture diagram shows how to integrate partner events that are produced from a partner's Software as a Service (SaaS) or your application. The events are delivered to an Amazon EventBridge custom event bus in your AWS account.



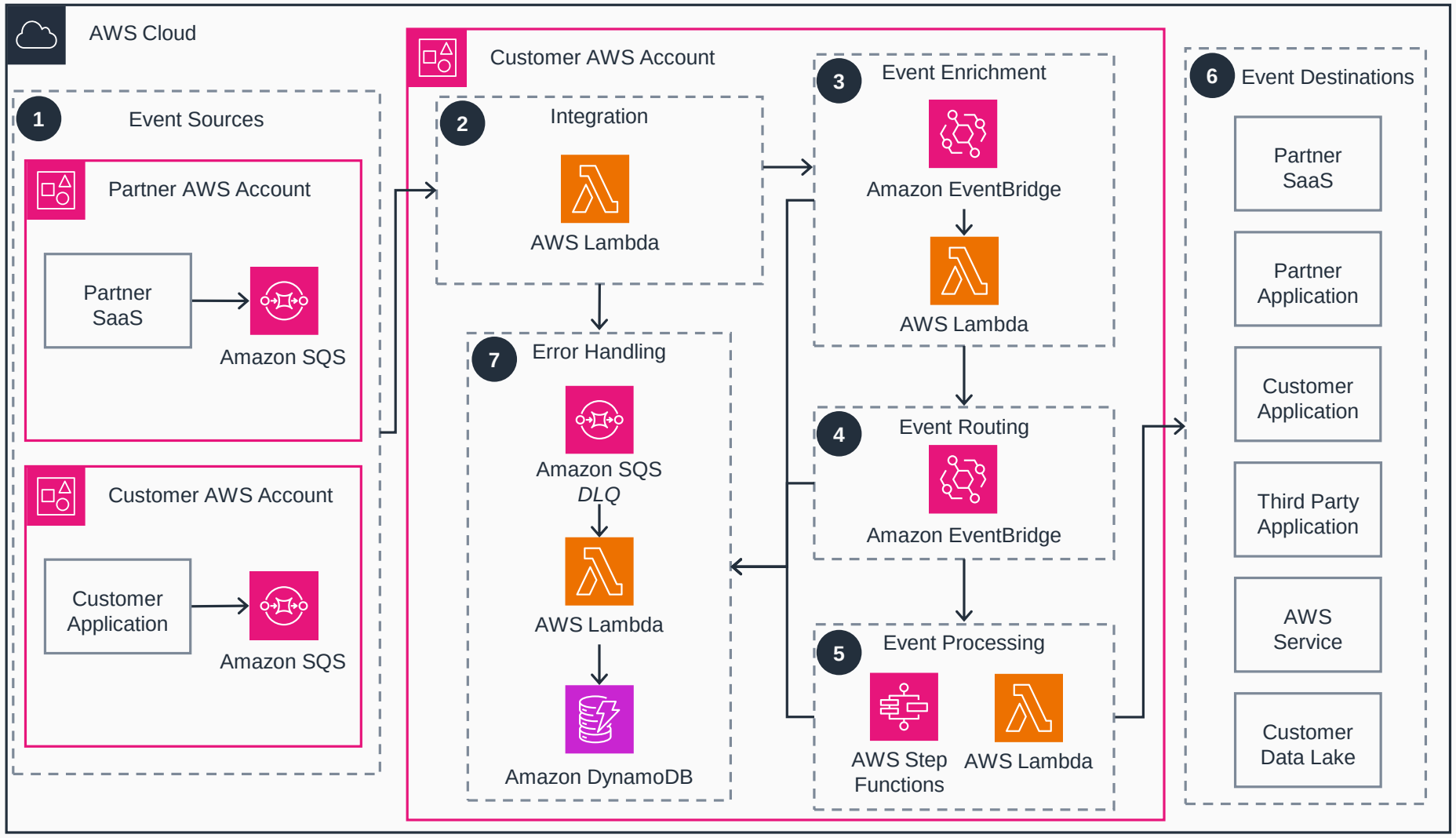
1. Partner SaaS applications, partner applications, or your application send events to your AWS account.
2. An AWS Partner SaaS platform or AWS Partner custom app writes events to the **EventBridge** custom event bus. The event bus receives partner events, and if necessary, an **EventBridge** rule routes the events to **Lambda** for transformation.
3. **EventBridge** rules route events to **Lambda** to be enriched and transformed. Skip this step if there is no need for enrichment or transformation.
4. **EventBridge** rules route events to a destination. Rules use event pattern matching to route events based on their contents.
5. **Lambda** sends event data to downstream services. If multiple steps are required, **Step Functions** starts a multi-step workflow using the event. Skip this step if there is no need for additional processing.
6. Downstream services, such as partner SaaS, a locally deployed partner application, your application, a third-party application, an AWS service, or your data lake, receive the event data.
7. **Amazon SQS** DLQ receives events not successfully processed from previous steps. **Amazon SQS** will re-driven and process the events. If re-driven events are unsuccessful, **Lambda** writes the event to a **DynamoDB** table for manual review.

Reviewed for technical accuracy March 212024

AWS Reference Architecture

Guidance for Event-Driven Media Workflow Automation on AWS

This architecture diagram shows how to integrate partner events that are produced by an Amazon SQS queue with an AWS Lambda function in your AWS account.



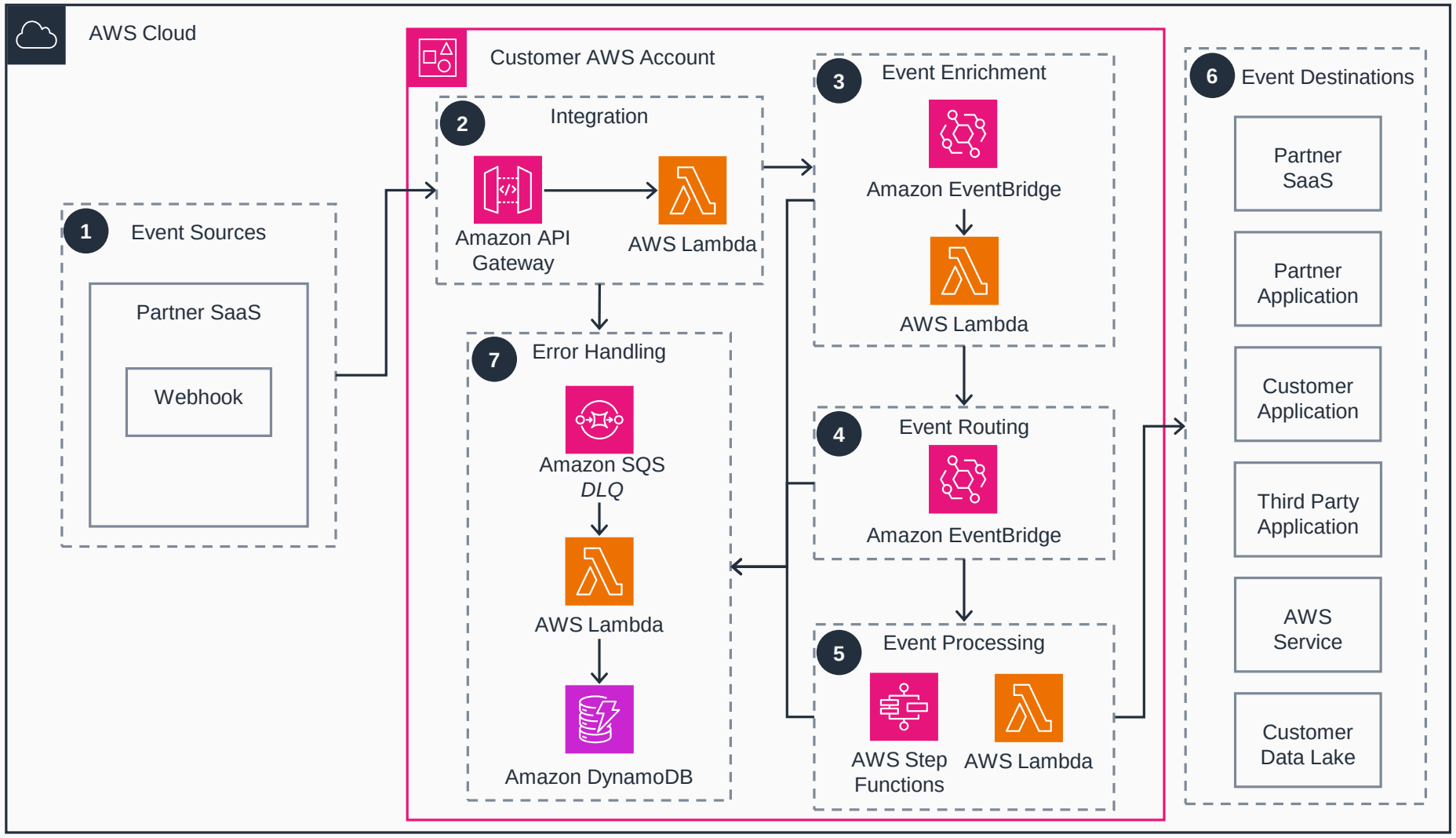
1. Partner SaaS applications, partner applications, or your application send events to your AWS account.
2. **Lambda** polls an AWS Partner SaaS or an AWS customer application's **Amazon SQS** queue. **Lambda** receives events from the **Amazon SQS** subscription and performs any needed transformation.
3. **EventBridge** rules route events to **Lambda** to be enriched and transformed. Skip this step if there is no need for enrichment or transformation.
4. **EventBridge** rules route events to a destination. Rules use event pattern matching to route events based on their contents.
5. **Lambda** sends event data to downstream services. If multiple steps are required, **Step Functions** starts a multi-step workflow using the event. Skip this step if there is no need for additional processing.
6. Downstream services, such as partner SaaS, a locally deployed partner application, your application, a third-party application, an AWS service, or your data lake, receive the event data.
7. **Amazon SQS** DLQ receives events not successfully processed from previous steps. **Amazon SQS** will re-driven and process the events. If re-driven events are unsuccessful, **Lambda** writes the event to a **DynamoDB** table for manual review.

Reviewed for technical accuracy March 212024

AWS Reference Architecture

Guidance for Event-Driven Media Workflow Automation on AWS

This architecture diagram shows how to integrate partner events produced by a partner webhook with an Amazon API Gateway endpoint.



1. Partner SaaS applications send events to your AWS account.
2. **API Gateway** receives HTTP API calls from third-party applications. **Lambda** receives the HTTP API call, processes the request, and responds to the third-party application.
3. **EventBridge** rules route events to **Lambda** to be enriched and transformed. Skip this step if there is no need for enrichment or transformation.
4. **EventBridge** rules route events to a destination. Rules use event pattern matching to route events based on their contents.
5. **Lambda** sends event data to downstream services. If multiple steps are required, **Step Functions** starts a multi-step workflow using the event. Skip this step if there is no need for additional processing.
6. Downstream services, such as partner SaaS, a locally deployed partner application, your application, a third-party application, an AWS service, or your data lake, receive the event data.
7. **Amazon SQS** DLQ receives events not successfully processed from previous steps. **Amazon SQS** will re-driven and process the events. If re-driven events are unsuccessful, **Lambda** writes the event to a **DynamoDB** table for manual review.

Reviewed for technical accuracy March 212024

AWS Reference Architecture