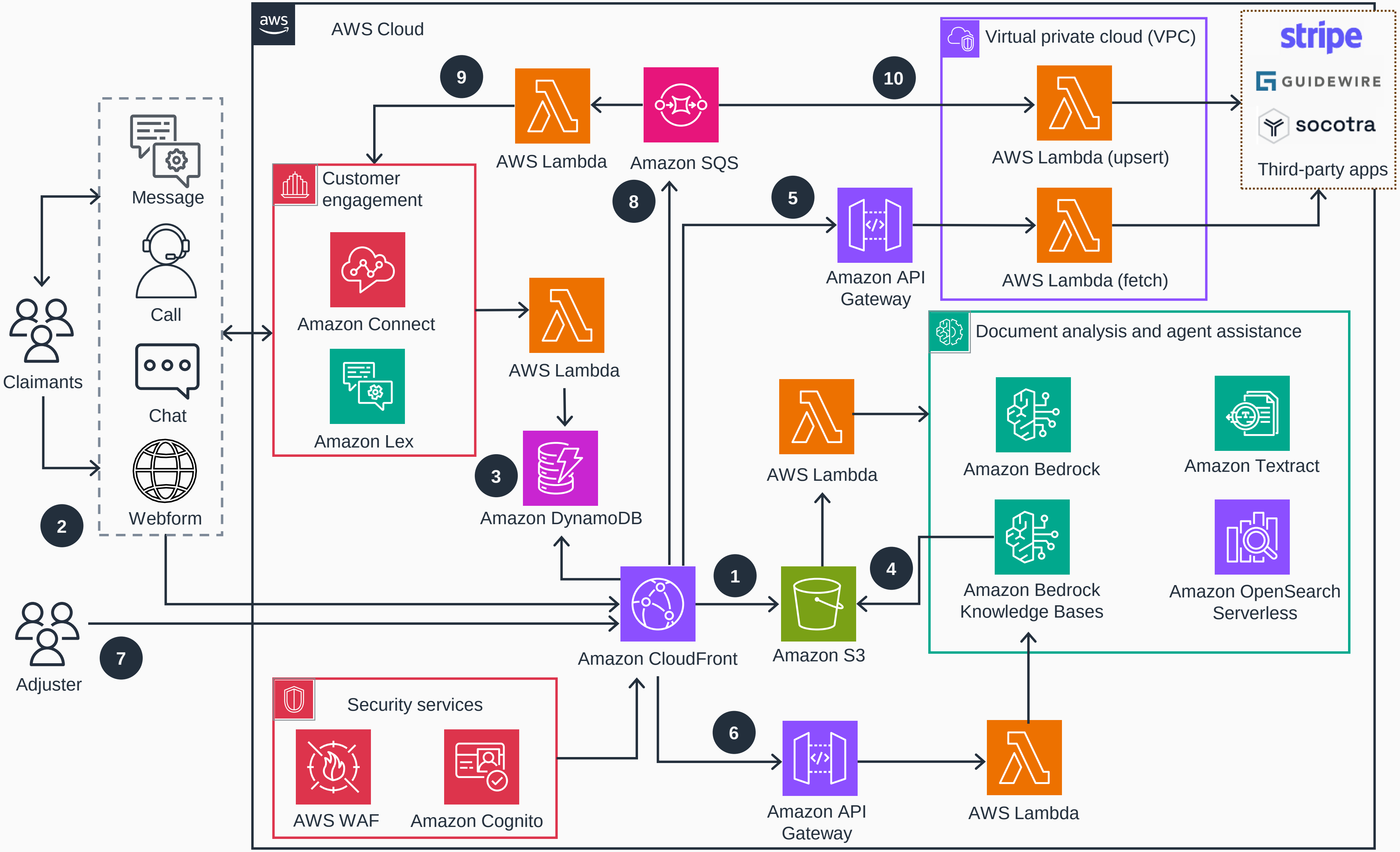
Guidance for Omnichannel Claims Processing Powered by Generative AI on AWS

This architecture diagram illustrates how to effectively support claims processing automation across multiple channels on AWS.



1 **Amazon CloudFront** serves the Claims Processing React Web Application, including an **Amazon Connect** chat interface. **Amazon Cognito** and **AWS WAF** protect **CloudFront.**

2 Initiate First Notice of Loss (FNOL) communication through call, SMS, and chat using **Amazon Connect** and **Amazon Lex** and webform using the Claims Processing Web Application.

3 An **Amazon DynamoDB** table stores claims request details.

4 **Amazon Simple Storage Service (Amazon S3)** stores claims documents through the Claims Processing Web Application. **Amazon S3** events trigger an **AWS Lambda** function, which invokes **Amazon Textract** to analyze documents, such as driver's licenses. The **Lambda** function also invokes the **Amazon Nova Pro** large language model (LLM) using **Amazon Bedrock** to analyze images of vehicle damages. **Lambda** updates the generated insights, including potential costs to replace and repair the coverable to existing claims records in the **DynamoDB** table.

5 **Amazon API Gateway** and **Lambda** integrate third-party application data to the Claims Processing Web Application.

6 The adjuster leverages **Amazon Bedrock Knowledge Bases** to search for information using **API Gateway** and **Lambda**. **Amazon S3** stores knowledge articles for the **Amazon Bedrock Knowledge Bases**. **Amazon OpenSearch Serverless** is used as the vector database.

7 The adjuster reviews and adjudicates the claim request using the web application.

8 The adjuster decision is sent to an **Amazon Simple Queue Service (Amazon SQS)** queue.

9 **Lambda** picks the messages from **Amazon SQS** and notifies the claimant with the status of the claim request using **Amazon Connect.**

10 **Lambda** picks the messages from **Amazon SQS** and updates third-party applications for further downstream processing (if required).

Reviewed for technical accuracy March 7, 2025

AWS Reference Architecture