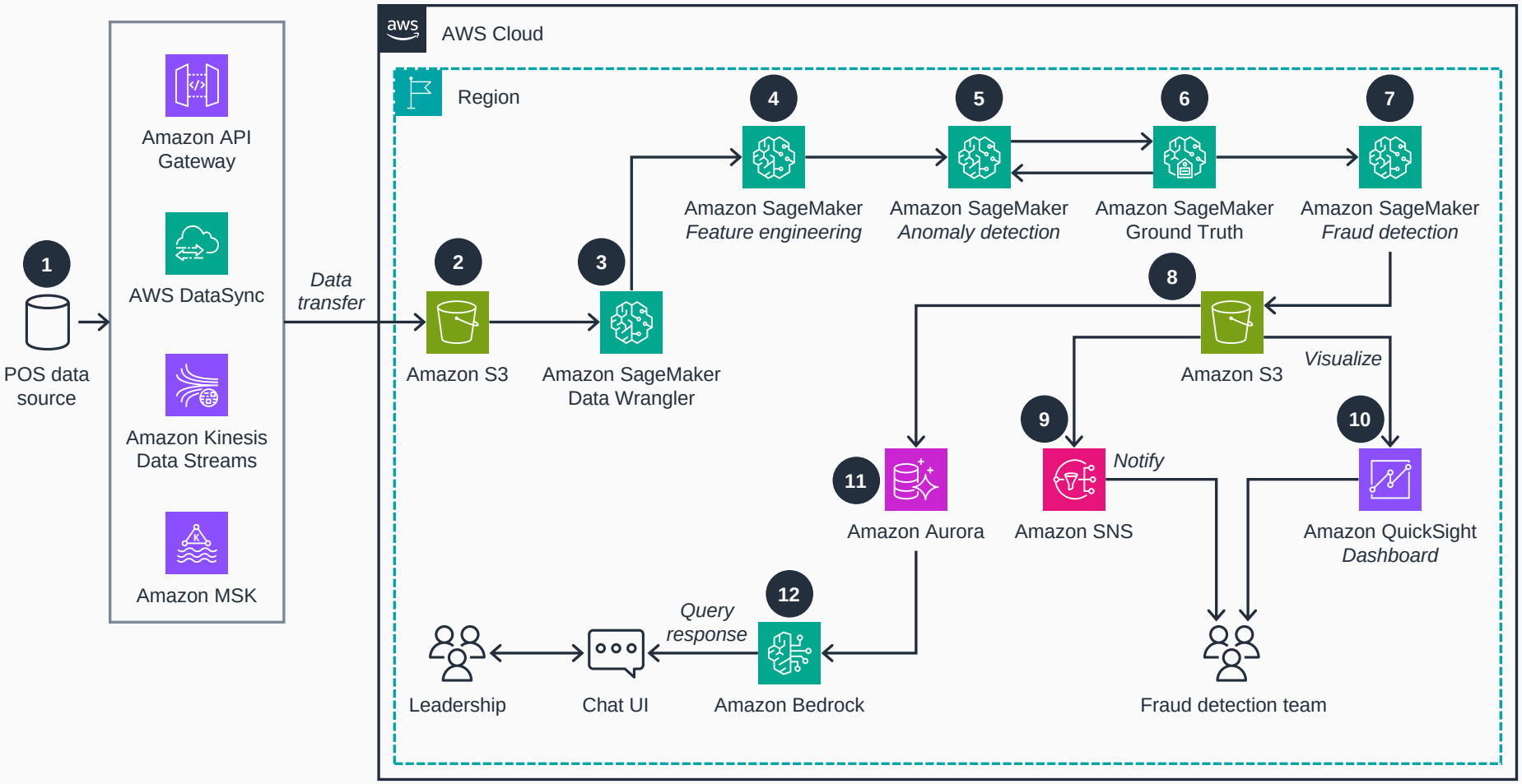
Guidance for Detecting Point-of-Sale Fraud Using Amazon SageMaker

This architecture diagram shows how T&H businesses can effectively detect fraudulent transactions at POS terminals and quickly gain insights from fraud prediction datasets by using Amazon SageMaker and Amazon Bedrock.



1. Ingest POS data to AWS using various data transfer services, including **Amazon API Gateway**, **AWS DataSync**, **Amazon Kinesis Data Streams**, or **Amazon Managed Streaming for Apache Kafka (Amazon MSK)**.
2. Store all raw ingested data in **Amazon Simple Storage Service (Amazon S3)**.
3. Clean and analyze the dataset using **Amazon SageMaker Data Wrangler**. For example, identify missing data, remove duplicates, or fix data types, then extract insights.
4. Denormalize the datasets using Jupyter notebooks through **Amazon SageMaker**. You can extract insights from the combined dataset and generate relevant features for accurate prediction.
5. Identify unusual datapoints, using an anomaly detection algorithm to highlight atypical behaviors in transactions.
6. Label the dataset using **Amazon SageMaker Ground Truth** based on domain expertise. Establish a feedback loop with the anomaly detection step to check model outputs against your labels.
7. Predict fraudulent transactions, hypertune parameters, and evaluate models using **SageMaker**.
8. Store model outputs in **Amazon S3**.
9. Based on your predictions, notify users of potential fraud using **Amazon Simple Notification Service (Amazon SNS)**.
10. Visualize results using **Amazon QuickSight**.
11. Combine and convert data into vectors using the embedding model through **Amazon Bedrock** and store the data in **Amazon Aurora**.
12. Questions asked by leadership, such as C-level executives, are embedded using **Amazon Titan in Amazon Bedrock**, then matched with similar vectors in **Aurora** PostgreSQL. The question and relevant context go to **Amazon Bedrock**, which generates a response for the chatbot.

Reviewed for technical accuracy November 1, 2024

AWS Reference Architecture