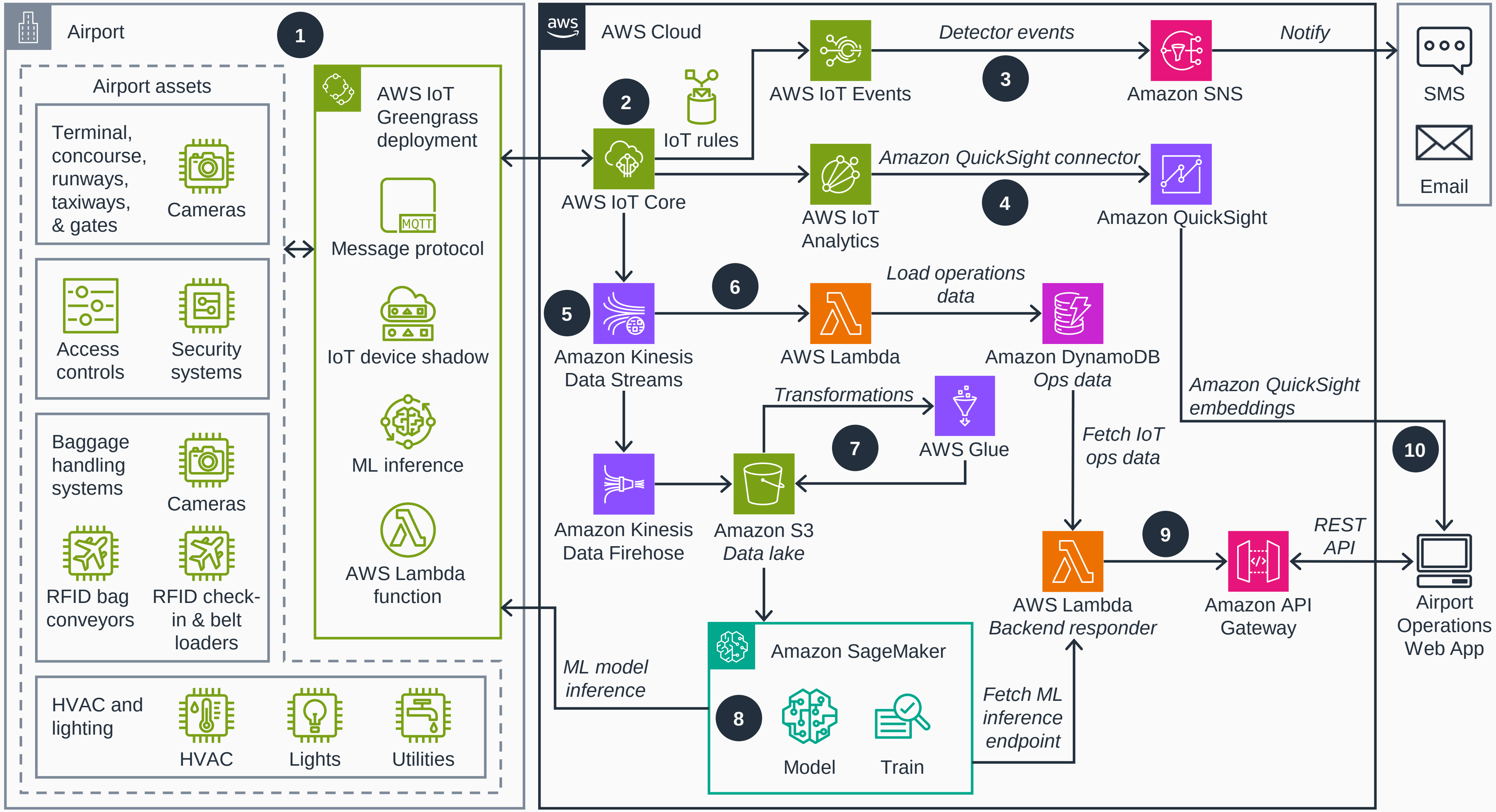
Guidance for Connected Airports Using Internet of Things on AWS

This architecture diagram shows how to build an Internet of Things (IoT)-connected airport asset monitoring system.



1 Use **AWS IoT Greengrass** core device to connect to, publish to, and subscribe to data from airport assets on the edge using the open standard Message Queuing Telemetry Transport (MQTT) protocol.

2 Use **AWS IoT Core** to maintain shadows of airport assets, connect to AWS, and manage messages from Internet of Things (IoT) sensors for further processing.

3 Create a detector model in **AWS IoT Events** with **AWS IoT Core** as the input source. Configure **Amazon Simple Notification Service (Amazon SNS)** in the detector model to send notifications by SMS or email when an unusual event occurs or when a sensor reaches your set thresholds.

4 Use **AWS IoT Analytics** to aggregate, transform, and analyze IoT messages from **AWS IoT Core**. Build an IoT analysis dashboard and visualize on **Amazon QuickSight**.

5 Configure an IoT rule to send messages from **AWS IoT Core** to **Amazon Kinesis Data Streams** for downstream processing.

6 Use an **AWS Lambda** function to process messages from **Kinesis Data Streams**, and store them on **Amazon DynamoDB**.

7 **Amazon Kinesis Data Firehose** reads data from **Kinesis Data Streams** and stores it in an **Amazon Simple Storage Service (Amazon S3)** data lake. Use **AWS Glue** to transform data and store it back on **Amazon S3**.

8 Use **Amazon SageMaker** to build, train, and validate machine learning (ML) models for predictive maintenance and anomaly detection of airport assets. Optionally, use this ML model inference with an **AWS IoT Greengrass** core device on the edge.

9 Use a **Lambda** function to process all IoT data stored on a **DynamoDB** table and fetch the ML model inference endpoint for predictions. Create a REST API with a **Lambda** function as a backend on **Amazon API Gateway**.

10 Develop an airport operations web application to centralize asset monitoring and predictive maintenance capabilities. Also, integrate a **QuickSight** dashboard using **QuickSight** embeddings.

Reviewed for technical accuracy January 8, 2024

AWS Reference Architecture