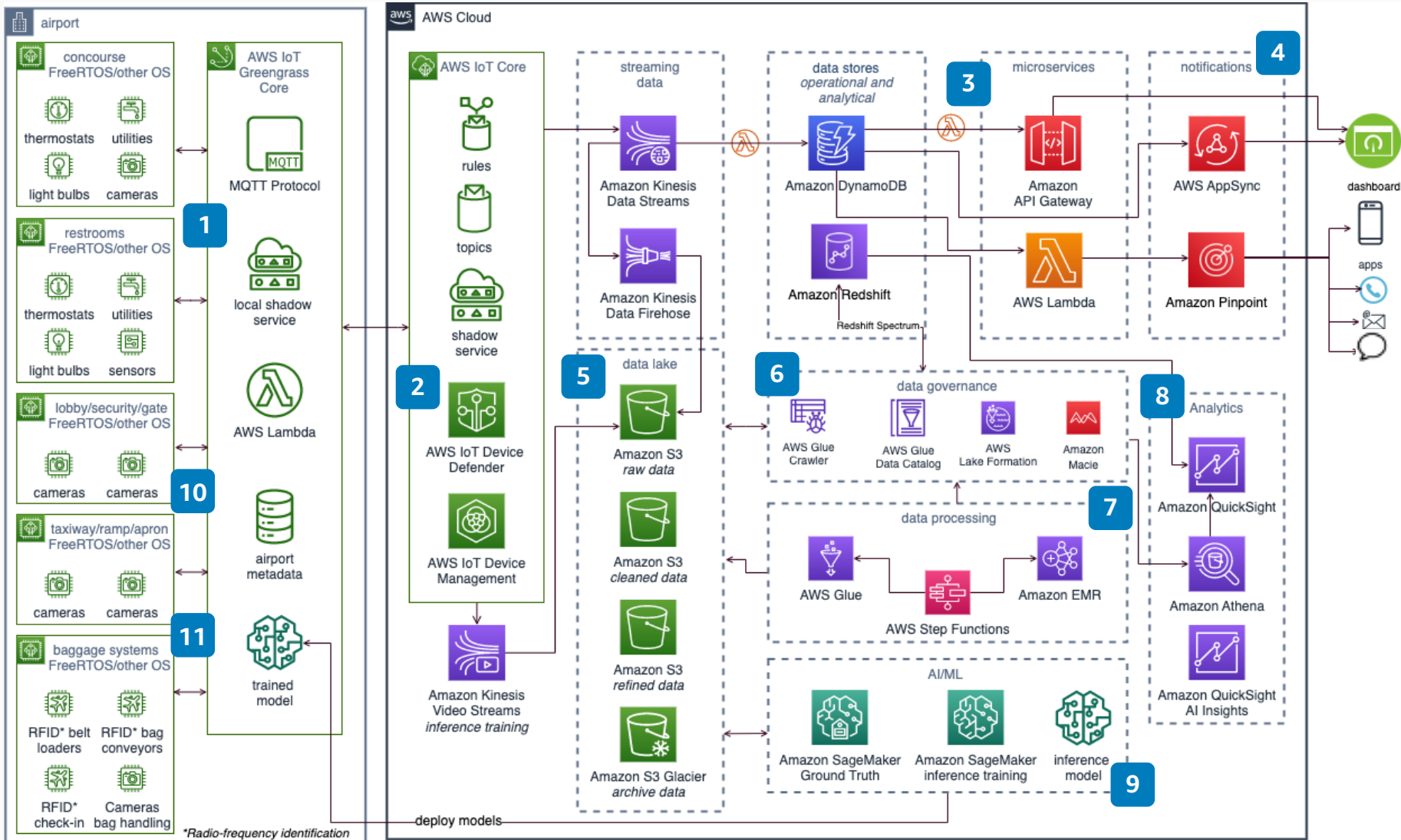# Connected Airports Using IoT and AI/ML

Build a smart, connected airport that uses IoT and artificial intelligence/machine learning (AI/ML) to generate near real-time data for aircraft movement, gate turns, baggage tracking, queue depth, and passenger traffic, and to implement compliance measures like social distancing and security.

airport

concourse FreeRTOS/other OS
- thermostats
- utilities
- light bulbs
- cameras

restrooms FreeRTOS/other OS
- thermostats
- utilities
- light bulbs
- sensors

lobby/security/gate FreeRTOS/other OS
- cameras
- cameras

taxiway/ramp/apron FreeRTOS/other OS
- cameras
- cameras

baggage systems FreeRTOS/other OS
- RFID* belt loaders
- RFID* bag conveyors
- RFID* check-in
- Cameras bag handling

*Radio-frequency identification

AWS IoT Greengrass Core
- MQTT Protocol
- local shadow service
- AWS Lambda
- airport metadata
- trained model

AWS Cloud

AWS IoT Core
- rules
- topics
- shadow service
- AWS IoT Device Defender
- AWS IoT Device Management
- Amazon Kinesis Video Streams *inference training*

streaming data
- Amazon Kinesis Data Streams
- Amazon Kinesis Data Firehose

data stores *operational and analytical*
- Amazon DynamoDB
- Amazon Redshift — Redshift Spectrum

microservices
- Amazon API Gateway
- AWS Lambda

notifications
- AWS AppSync
- Amazon Pinpoint

dashboard
apps

data lake
- Amazon S3 *raw data*
- Amazon S3 *cleaned data*
- Amazon S3 *refined data*
- Amazon S3 Glacier *archive data*

data governance
- AWS Glue Crawler
- AWS Glue Data Catalog
- AWS Lake Formation
- Amazon Macie

data processing
- AWS Glue
- AWS Step Functions
- Amazon EMR

AI/ML
- Amazon SageMaker Ground Truth
- Amazon SageMaker inference training
- inference model

Analytics
- Amazon QuickSight
- Amazon Athena
- Amazon QuickSight AI Insights

deploy models

1 Leverage **AWS IoT Greengrass Core** to connect, publish, and subscribe data using open standard MQTT protocol with Internet of Things (IoT) devices running on **FreeRTOS** and other operating systems (OS').

2 Leverage **AWS IoT Core** to maintain shadows of all IoT devices, connect to **AWS Cloud**, manage devices, update over-the-air (OTA), and secure the devices.

3 Use purpose-built databases like **Amazon DynamoDB** and serverless architecture to store events, deliver microservices, and generate events for the operational data store.

4 Build a real-time operational dashboard using microservices and **AWS AppSync**. Deliver alerts to multiple channels using **Amazon Pinpoint**.

5 Build the data lake to store raw data and to create curated processed data in **Amazon Simple Storage Service** (Amazon S3) using **AWS Glue** and **Amazon EMR**.

6 Discover and govern the data in **Amazon S3** using **AWS Glue** crawlers, **AWS Glue Data Catalog**, and **AWS Lake Formation**. Additionally deploy **Amazon Macie** to detect any sensitive data.

7 Use **AWS Glue** jobs and **Amazon EMR** to perform any transformation or enrichment of the data.

8 Use **Amazon Redshift**, **Amazon Athena**, and **Amazon QuickSight** for analytics. Optionally, build data marts in **Amazon Redshift** for heavily used analytics. For one-time requirements, publish the data catalog and use **Amazon Athena** or **Amazon Redshift Spectrum** for direct analysis using the data lake.

9 Use **Amazon SageMaker** to build, train, and deploy inference models. Optionally, deploy edge models on **AWS IoT Greengrass Core**.

10 Use the Aircraft Turn Tracking solution to passively collect and use aircraft gate turn events.

11 Use the Facilitate Social Distancing and Queue Depth Management solutions for compliance and enhanced customer experience.

AWS Reference Architecture