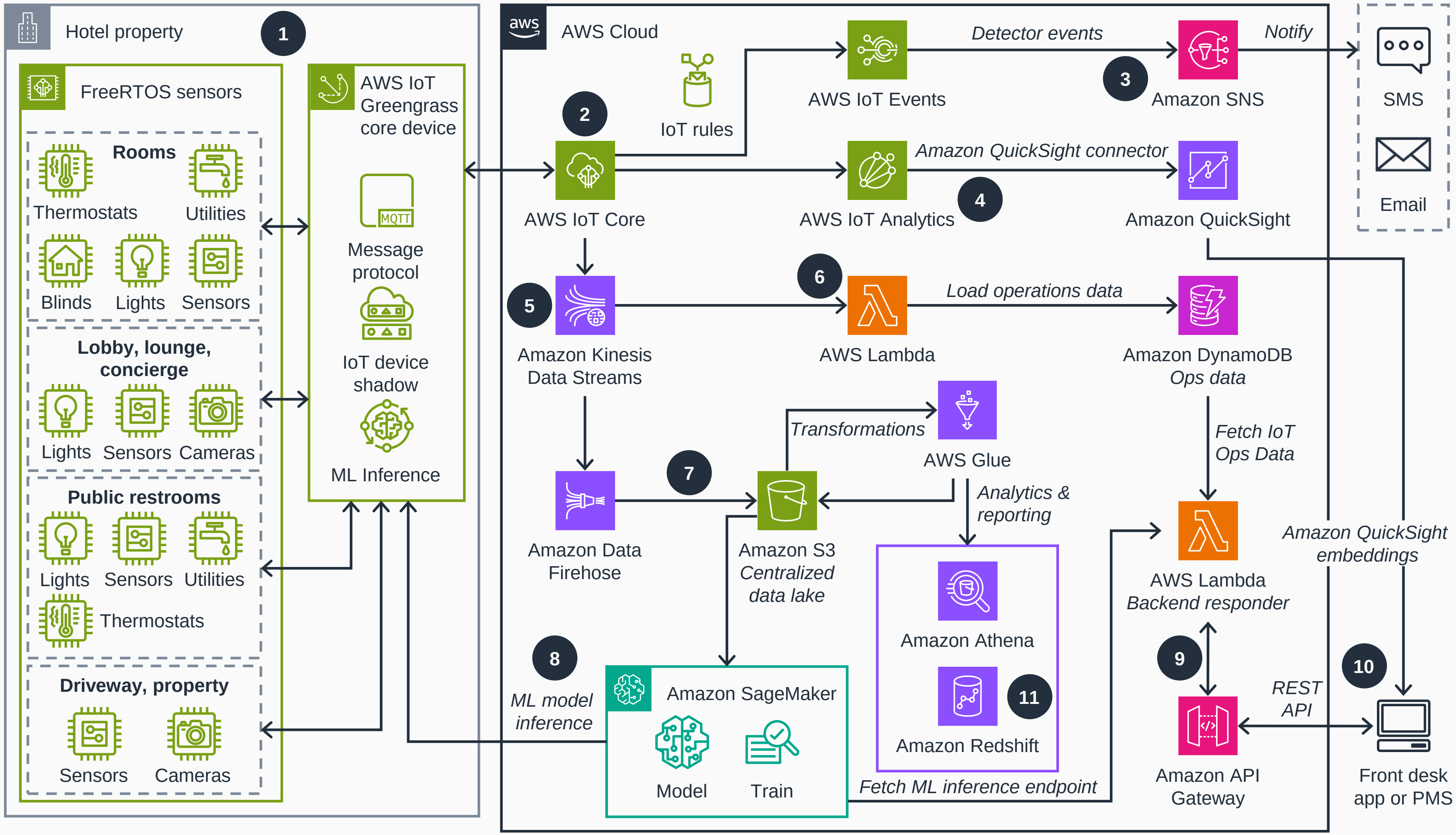
Guidance for Implementing Connected Lodging Properties on AWS

This architecture diagram shows how to implement a connected IoT lodging solution that enhances guest experiences, optimizes resource utilization, and streamlines operations. This slide details steps 1-8; steps 9-11 are detailed on the next slide.



1. Use an **AWS IoT Greengrass** core device to connect, publish, and subscribe to data from your hotel property device IoT sensors on the edge using the open-standard MQTT protocol.

2. Use **AWS IoT Core** to maintain shadows of all IoT devices, connect to AWS, and manage messages from 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 a sensor reaches set thresholds.

4. Use **AWS IoT Analytics** to aggregate, transform, and analyze IoT messages from **AWS IoT Core**. Build an IoT analysis dashboard and visualizations 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 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, then store it back on **Amazon S3**.

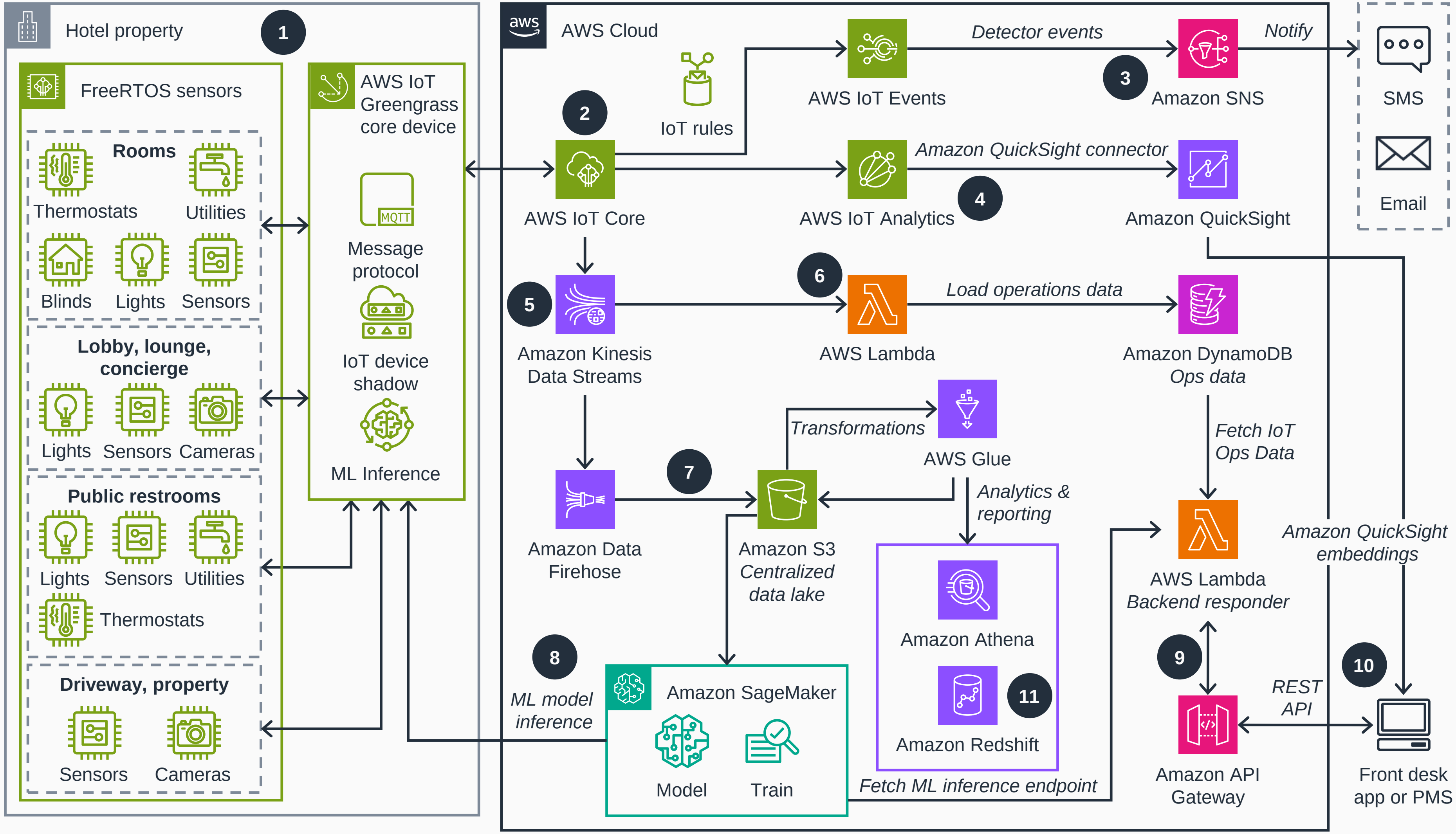
8. Use **Amazon SageMaker** to build, train, and validate ML models for predictive maintenance and anomaly detection for your kitchen equipment. Optionally, use this ML model inference with an **AWS IoT Greengrass** core device on the edge.

Reviewed for technical accuracy December 11, 2024

AWS Reference Architecture

Guidance for Implementing Connected Lodging Properties on AWS

Steps 9-11



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

10 Create a property management system (PMS), such as a kitchen operations app or front desk app, that centralizes equipment monitoring and predictive maintenance capabilities. Also, integrate a **QuickSight** dashboard using **QuickSight** embeddings.

11 Optionally, get deeper insights through reporting and one-time analytics using **Amazon Redshift** and **Amazon Athena**.

Reviewed for technical accuracy December 11, 2024

AWS Reference Architecture