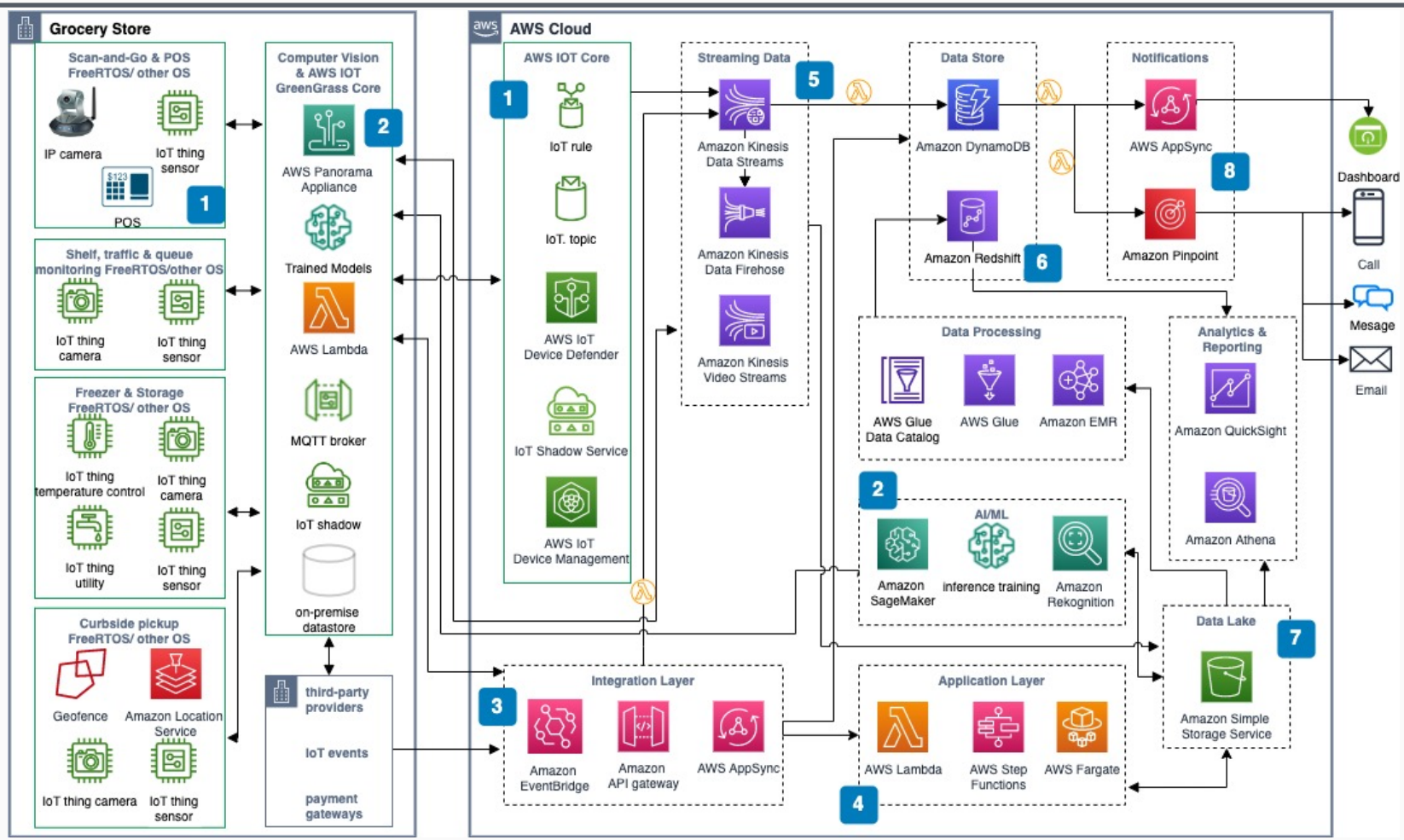
Smart Grocery with Scan-and-Go, Computer Vision, and IoT capability

This is a reference architecture for building a Smart Grocery with Scan-and-Go shopping; traditional point-of-sale (POS); self-checkout; IoT connectivity and orchestration around freezers and storage; and all-in computer vision using AWS Panorama for shelf stock intelligence, traffic patterns, queue analysis, and curbside fulfillment automation.



1. Use **AWS IoT Greengrass core** to help manage connections and data aggregation from various in-store sensors and smart retail devices using open standard Message Queuing Telemetry Transport (MQTT) protocol.

2. Use the **AWS Panorama** on-premise appliance to efficiently apply machine learning and artificial intelligence (AI/ML) learning models to data (from existing in-store IP cameras to support of Smart Grocery picking and fulfillment workflows).

3. **Amazon API Gateway**, **Amazon EventBridge,** and **AWS AppSync** are used as a integration layer between customer-facing digital ordering applications, associate-facing order-picking and fulfillment applications, and their required business logic and data access sources.

4. Core Smart Grocery application layer is built using serverless platforms and services like **AWS Lambda** and **AWS Step Functions**, and helps orchestrate order management workflows and connections to transaction, customer, and inventory data sources.

5. **Amazon Kinesis Data Streams** and **Amazon Kinesis Data Firehose** are used to stream in-store smart device/IoT data. **Amazon Kinesis Video Streams** is used to optimize IP camera video feeds for computer vision-based Smart Grocery applications.

6. **Amazon DynamoDB** is used for storing events and connecting to smart grocery application services, and to generate notifications related to transactional data.

7. Scalable data lake is used to store all sources of in-store and digital data including transactional, sensor, and telemetry sources in **Amazon S3.**

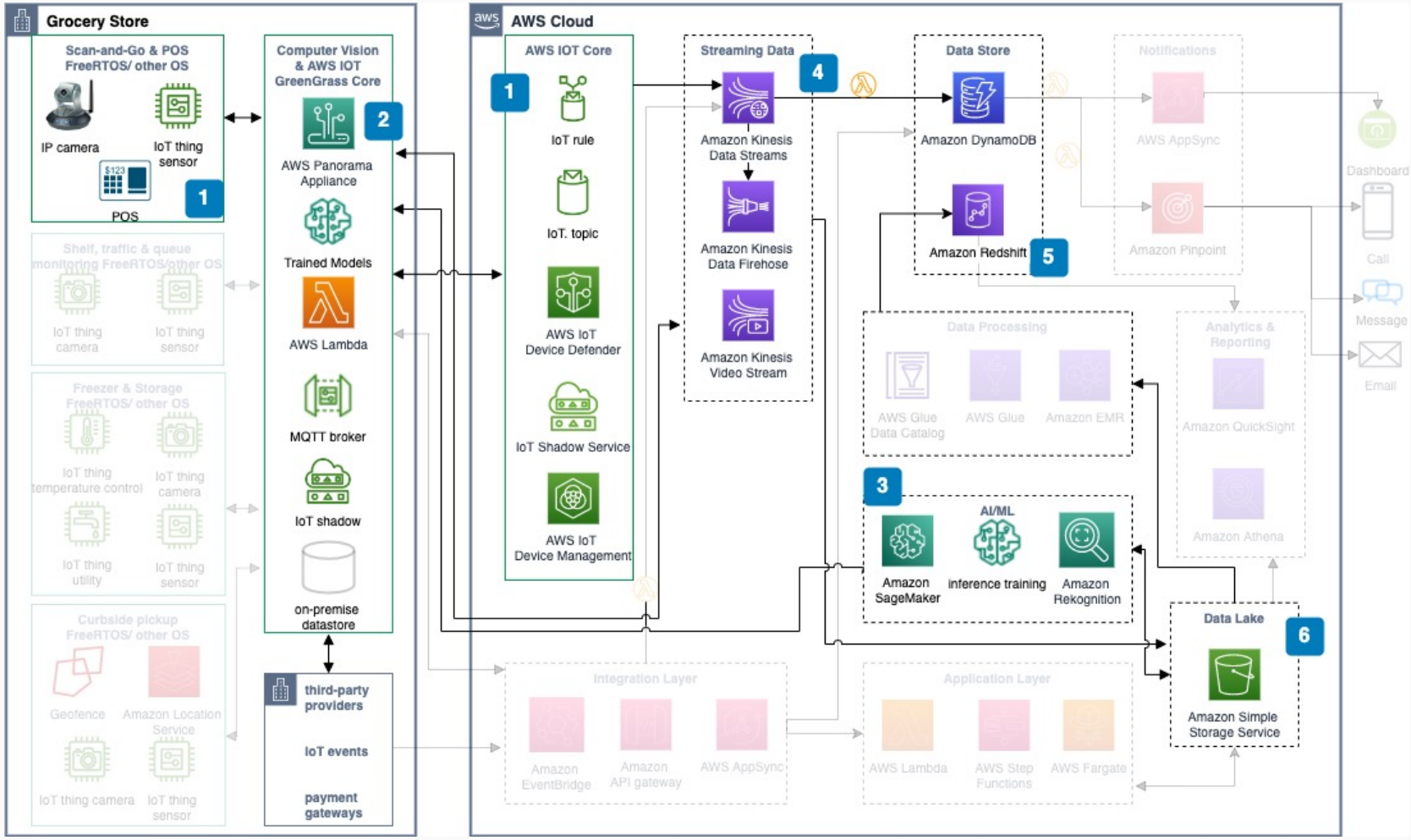
8. Custom build a real-time operations dashboard integrating Smart Grocery-defined services using **AWS AppSync**. Leverage **Amazon Pinpoint** to delivers targeted, location-based customer and associate messaging via multiple channels.

Reviewed for technical accuracy May 18th, 2022

AWS Reference Architecture

Use case 1: Scan-and-Go

This is a reference architecture for a Scan-and-Go use case with an AWS Panorama appliance connected to in-store cameras and running ML models in the store.



1 IP cameras positioned in-store capture real-time videos of the checkout process, and along with sensors and POS, help identify the product and make the checkout process seamless for the customer.

2 This video data is processed on-premise by **AWS Panorama** using optimized AI/ML models. The resulting analytic data can then be integrated into specific transactions within customer ordering and associate fulfillment applications that support Smart Grocery platforms.

3 Processed videos from the **AWS Panorama** appliance are sent to **Amazon SageMaker Ground Truth** and **Amazon SageMaker** inference training in support of optimizing computer vision training models. This helps improve customer behavior analysis, and digital order and fulfillment efficiency.

4 Ingestion of in-store sensor and retail device data is managed by **Amazon Kinesis Data Streams** and the processed data is directed into an **Amazon S3** data lake using **Amazon Kinesis Data Firehose**. **Amazon Kinesis Video Streams** is used to optimize multiple IP camera feeds, and the resulting data used by **AWS Panorama** computer vision applications.

5 Sensor telemetry data captured from smart in-store devices such as digital signage or cooling systems are stored in **Amazon DynamoDB,** with trigger-based notifications generated for use by Smart Grocery customer support teams. **Amazon Redshift** is used as a core data warehouse platform in support of key data science and analytics functions.

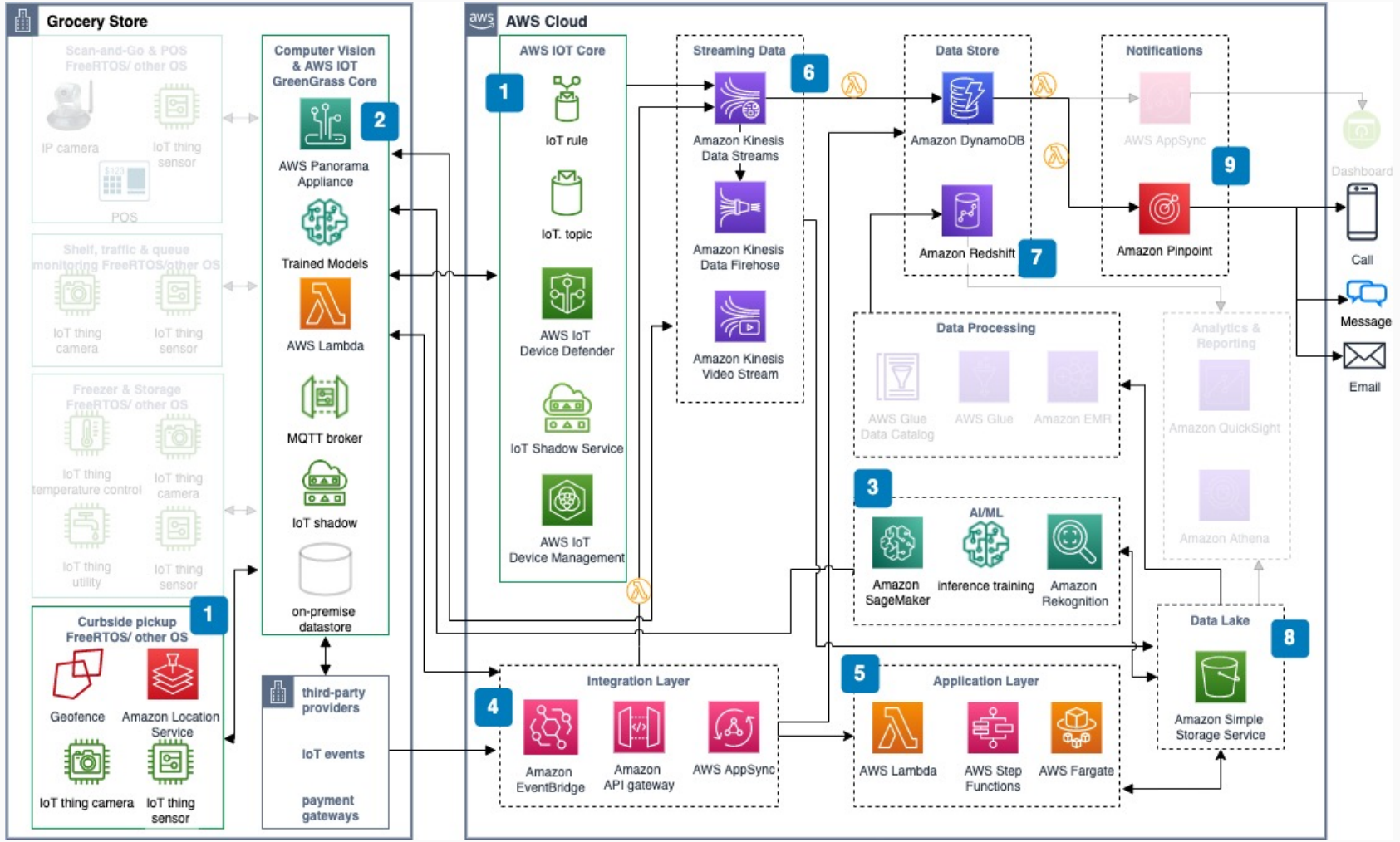
6 Use a scalable data lake on **Amazon S3** to store raw data from in-store devices, and also to store curated processed data which can include images, shopper analytics, marketing intelligence, and digital commerce details.

Reviewed for technical accuracy May 18th, 2022

AWS Reference Architecture

Use case 2: Curbside Pickup

This is a reference architecture for a Curbside Pickup use case with cameras and sensors in the parking lot and an AWS Panorama appliance running ML models in the store.



1. **Amazon Location Service** helps with **Geofence** and notifies in-store associate once customers enter the parking lot. Cameras and sensors monitor the vehicle movement and reads the tag number.

2. An application and pre-trained ML model deployed to the **AWS Panorama** appliance will process the video from cameras. Custom application then notifies the in-store customer support team for curbside delivery for the particular customer.

3. Processed videos are sent to **Amazon SageMaker Ground Truth** and **Amazon SageMaker** inference training for training models.

4. **Amazon API Gateway**, **Amazon EventBridge**, and **AWS AppSync** are used as a primary integration layer between customer-facing digital ordering applications and associate-facing order-picking and fulfillment applications.

5. Core Smart Grocery application layer is built using serverless platforms and services like **AWS Lambda** and **AWS Step Functions**, and helps orchestrate order management workflows and connections to transaction, customer, and inventory data sources.

6. Ingest sensor data using **Amazon Kinesis Data Streams** and load the processed data into Amazon S3 data. **Amazon Kinesis Video Streams** is used to stream camera feeds.

7. Events captured from sensors/cameras are stored in **Amazon DynamoDB**, and notifications are generated for in-store customer support team.

8. Build a data lake on **Amazon S3** to store raw data from devices and to store curated processed data which includes images, video, customer details, and shopping information.

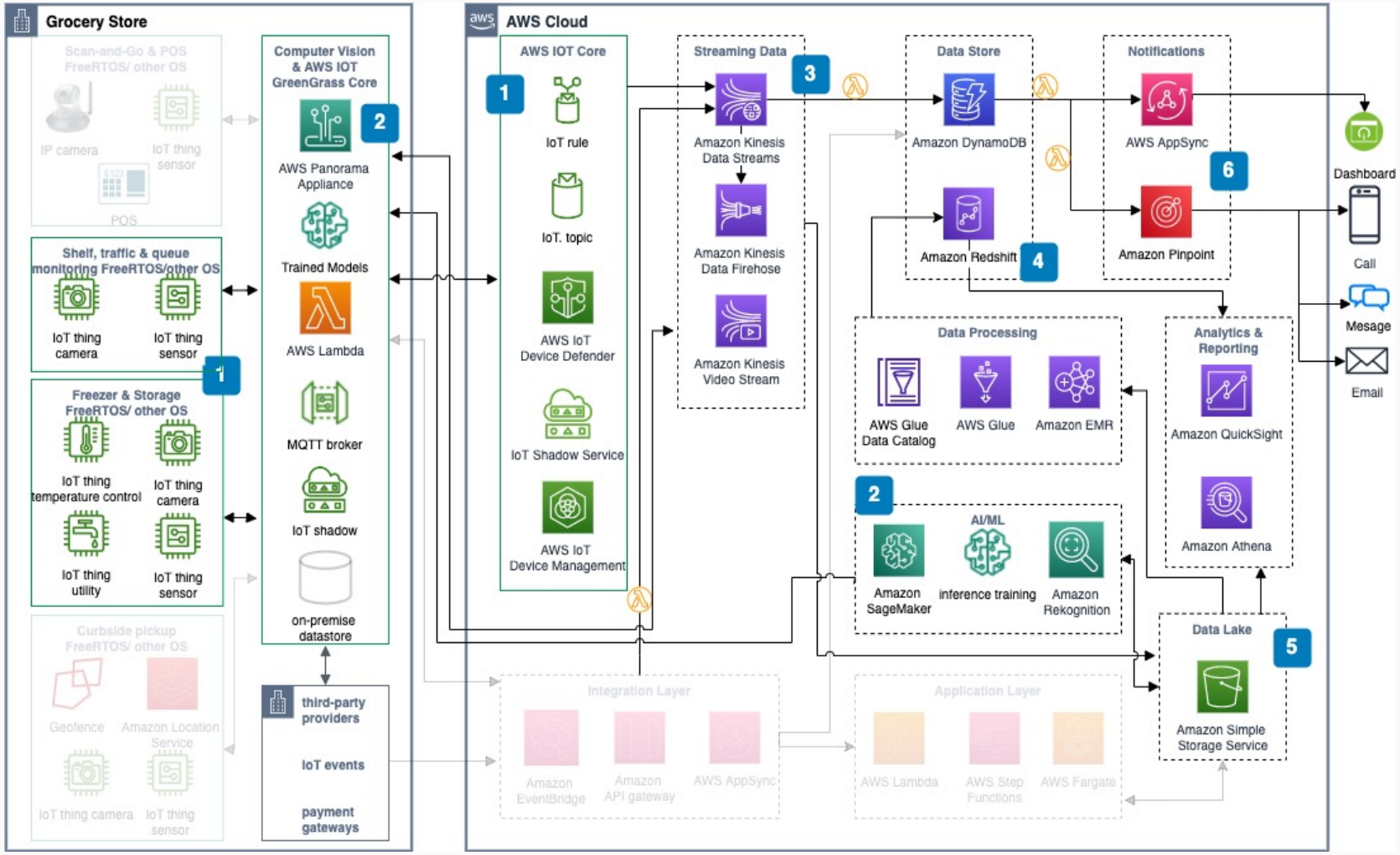
9. Build a real-time operational dashboard using microservices and **AWS AppSync**. Delivers alerts to multiple channels using **Amazon Pinpoint.**

Reviewed for technical accuracy May 18th, 2022

AWS Reference Architecture

Use case 3: In-Store Monitoring

This is a reference architecture for monitoring shelves, queue, the temperature control of freezers, utility, and other grocery storage using AWS Panorama appliance, and also alerting store operations teams for further action.



1. Use **AWS IOT Greengrass core** to help manage connections and data aggregation from various in-store sensors and smart retail devices using open standard MQTT protocol, and use **AWS IoT core** to manage all in-store smart devices.

2. Use the **AWS Panorama** on-premise appliance to efficiently apply AI/ML learning models to data (from existing in-store IP cameras to support of Smart Grocery picking and fulfillment workflows). Use **Amazon SageMaker Ground Truth** and **Amazon SageMaker** inference training to build and maintain related computer vision learning models.

3. **Amazon Kinesis Data Streams** and **Amazon Kinesis Data Firehose** are used to stream (at scale) in-store smart device/IoT data for storage into Amazon S3 data storage locations. **Amazon Kinesis Video Streams** is used to optimize IP camera video feeds into **AWS Panorama** for computer vision-based smart grocery applications.

4. **Amazon DynamoDB** is used for storing events, connecting to Smart Grocery application services, and generating notifications related to transactional data. **Amazon Redshift** is used as core data warehouse platform in support of key data science and analytics functions.

5. Use a scalable data lake to store all sources of in-store and digital data including transactional, sensor, and telemetry sources in **Amazon S3** using **AWS Glue** and **Amazon EMR**. Use **Amazon Athena** to support Smart Grocery reporting and analytic requests from the data lake.

6. Custom build a real-time operations dashboard integrating Smart Grocery-defined services using **AWS AppSync**. Leverage **Amazon Pinpoint** to delivers targeted, location-based customer and associate messaging via multiple channels.

Reviewed for technical accuracy May 18th, 2022

AWS Reference Architecture