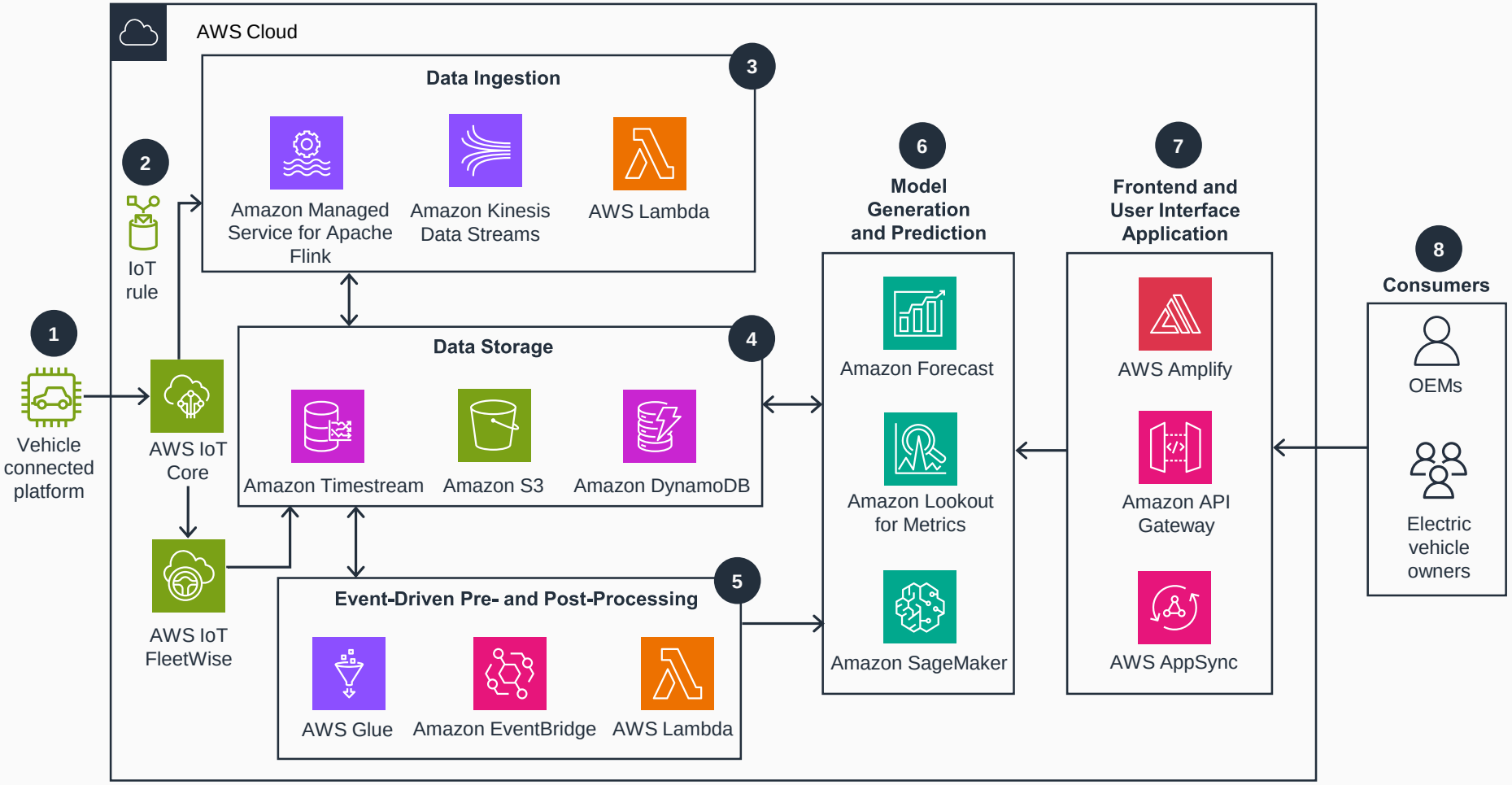
Guidance for Battery Digital Twin on AWS

This architecture diagram helps illustrate the battery digital twin for different use-cases involving fault detection using AWS services for real-time and historic data collection, state-of-health of battery forecasting, and real-time data analysis.



1. The vehicle sends telemetry data, such as operating voltage, current, temperature, and cell-level data, to **AWS IoT Core**.
2. An **AWS IoT rule** differentiates data based on whether the data is required for real-time analytics or batch analysis.
3. **Amazon Kinesis Data Streams** and **Amazon Managed Service for Apache Flink** ingest telemetry data for diagnostic trouble code (DTC) detection. **AWS Lambda** initiates the pre-processing job in **AWS Glue** to transform battery health data into csv format**.**
4. Real-time telemetry data, such as current, temperature, and charge data, is sent to **Amazon Timestream** for threshold model detection. Batch data is sent to **Amazon Simple Storage Service (Amazon S3)** for anomaly model training, and **Amazon DynamoDB** stores metadata.
5. **AWS Glue** pre-processes data by adding context to data stored in **Amazon S3**. **AWS Glue** post-processes timeseries data to visualize in the frontend application. **Amazon EventBridge** orchestrates the event workflow for model prediction**.**
6. **Amazon Forecast** predicts the state of health of the battery using a pre-trained model. **Amazon SageMaker** trains a prediction model based on the batch battery data in **Amazon S3**.
7. **AWS Amplify** deploys and hosts the frontend application. **AWS AppSync** enriches data using custom data sources. **Amazon API Gateway** manages APIs securely.
8. Original equipment manufacturers (OEMs) and electric vehicle owners can access this application securely through **API Gateway**.

Reviewed for technical accuracy May 7, 2024

AWS Reference Architecture