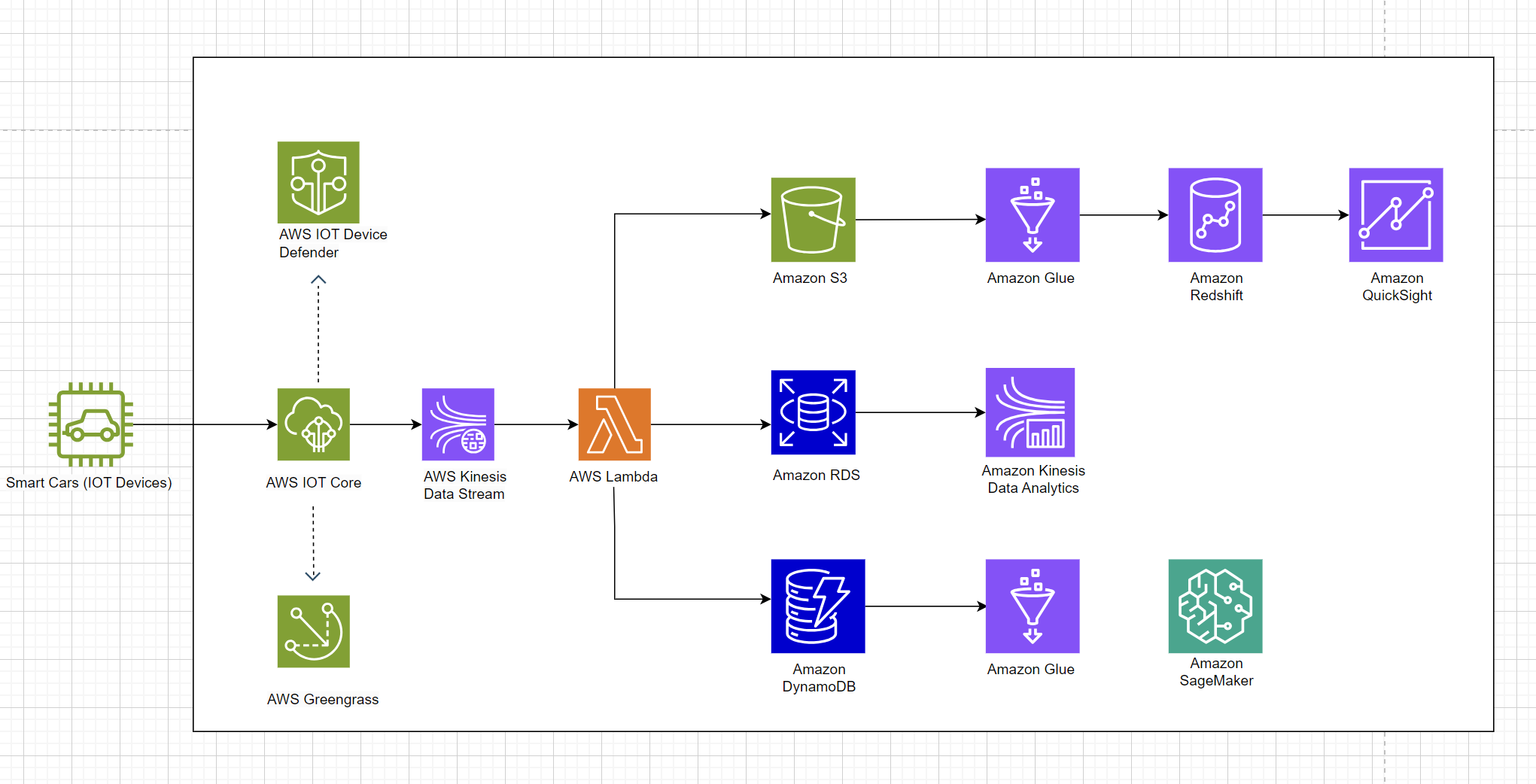
Cloud Architecture Design



Link Architecture : <https://drive.google.com/file/d/1zJnpcR87OM2fQDJLYO8e5IHGQuW8ivzQ/view?usp=sharing>

Components :

1. Smart Cars (IoT Devices)

* Smart cars equipped with IoT devices send data to the cloud

2. AWS IoT Core

* Receives data from IoT devices (smart cars) and manages communication between devices and the cloud
* Connected to AWS IoT Device Defender to secure IoT devices from security threats
* Connected to **AWS Greengrass** for local data processing and software updates on IoT devices

3. AWS Kinesis Data Stream

* Captures and processes streaming data from AWS IoT Core
* The streaming data is then sent to various AWS services for further processing and analysis

4. AWS Lambda

* Automatically processes data as new data arrives from Kinesis Data Stream
* Acts as an integration point for various storage and data analysis services

5. Amazon S3

* Stores raw data received from IoT devices for long-term storage and further analysis

6. Amazon RDS

* Stores structured data that has been processed for further analysis and reporting

7. Amazon DynamoDB

* Stores unstructured data with low latency for quick and efficient access

8. Amazon Glue

* ETL (Extract, Transform, Load) service to prepare data for further analysis and processing

9. Amazon Redshift

* Data warehouse that enables large-scale data analysis and reporting

10. Amazon Kinesis Data Analytics

* Analyzes streaming data in real-time to gain actionable insights

11. Amazon QuickSight

* Data visualization service for creating informative reports and dashboards

12. Amazon SageMaker

* Platform for building, training, and deploying machine learning models to enhance the features and capabilities of smart cars

**Workflow :**

1. Data Collection

* Data is collected from smart cars and sent to AWS IoT Core.
* AWS IoT Core manages device communication and security via AWS IoT Device Defender.

2. Data Processing

* Data from AWS IoT Core is forwarded to AWS Kinesis Data Stream for initial processing.
* AWS Lambda automatically processes the data and routes it to various storage services.

3. Data Storage

* Raw data is stored in Amazon S3
* Structured data is stored in Amazon RDS
* Unstructured data with quick access needs is stored in Amazon DynamoDB.

4. Data Analysis

* Amazon Glue performs ETL on the data for further analysis
* Amazon Redshift is used for large-scale data analysis
* Amazon Kinesis Data Analytics performs real-time streaming data analysis
* Amazon QuickSight provides data visualization for creating reports and dashboards.

5. Machine Learning

* Amazon SageMaker is used to build, train, and deploy machine learning models to enhance smart car features.

Improving Product Features via OTA :

AWS Greengrass enables local data processing and software updates on IoT devices (smart cars), facilitating OTA (Over-The-Air) updates to improve product features and capabilities remotely

4. AWS Services for Image Analysis

* Amazon Rekognition

**Reason :**

* Pay-as-you-go pricing model ensures you only pay for the images analyzed, reducing upfront costs.
* Highly available and managed service with built-in redundancy

**Implementation and Optimization :**

* Implement pre-processing to filter out unnecessary images before sending them to Rekognition, reducing the number of API calls
* Use multiple AWS regions to ensure availability and disaster recovery
* AWS Lambda

**Reason :**

* Serverless architecture means you only pay for the compute time you consume
* Automatically scales based on the workload, ensuring consistent performance

**Implementation and Optimization :**

* Optimize Lambda functions to minimize execution time and memory usage, and use AWS Lambda's free tier to handle low-traffic periods
* Set up retries and dead-letter queues to handle failed invocations
* Amazon S3

**Reason :**

* Cost-effective storage with different storage classes (e.g., Standard, Infrequent Access, Glacier) to optimize costs based on data access patterns
* Designed for 99.999999999% (11 9's) of durability and 99.99% availability

**Implementation and Optimization :**

* Store frequently accessed images in Standard and archival images in Glacier
* Enable versioning and cross-region replication for data resilience
* Amazon API Gateaway

**Reason :**

* Only pay for the number of API calls and data transfer
* Provides high availability and fault tolerance

**Implementation and Optimization :**

* Use caching to reduce the number of API calls for frequently requested data
* Deploy APIs across multiple regions and set up custom domain names with regional endpoints
* Amazon RDS or DynamoDB

**Reason :**

* Managed services with on-demand scaling and pricing models suited to various workloads
* Managed databases with automated backups, snapshots, and multi-AZ deployments for high availability

**Implementation and Optimization :**

* Choose the appropriate instance size and storage type based on usage patterns. Use DynamoDB's on-demand mode to pay per request if the access patterns are unpredictable
* Use Multi-AZ deployments in RDS or enable DynamoDB Global Tables for multi-region redundancy

Example workflow of architecture :

1. User submits an image through the chatbot interface.
2. Amazon API Gateway receives the image upload request and stores the image in Amazon S3.
3. AWS Lambda is triggered by the S3 upload event to process the image and invoke Amazon Rekognition.
4. Amazon Rekognition analyzes the image and returns the results.
5. AWS Lambda processes the results and stores them in Amazon RDS/DynamoDB.
6. The chatbot retrieves analysis results from Amazon RDS/DynamoDB and responds to the user.