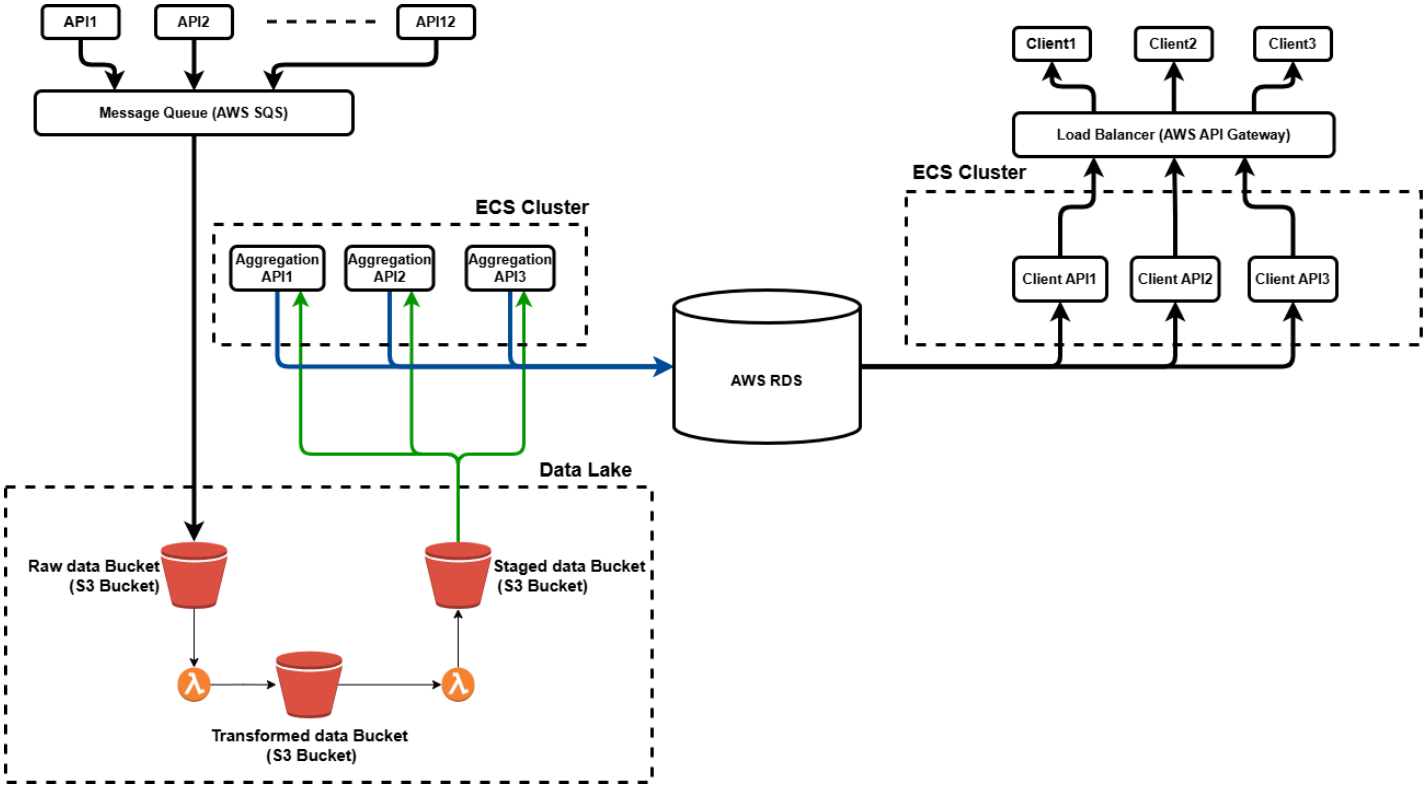

System Components

Data Collection:

12 APIs collect incoming data from various sources.These APIs send the raw data to an AWS Message Queue (SQS). SQS sits as a buffer between the data lake and the 12 APIs. This decouples the API from S3 allows the API to focus solely on receiving requests and sending messages, without being burdened by the complexities of S3 interactions. It also ensures reliability by persisting messages until they are successfully processed and written to S3. Rate limiting can also be implemented at the SQS.

Data processing:

Data from the SQS queue is sent to a data lake that consists of 3 S3 buckets:

1. Raw Data Bucket which stores unprocessed data.
2. Transformed Data Bucket which stores cleaned data.
3. Staged Data Bucket which stores fully structured data, ready for ingestion.

The steps in the data lake are controlled by two lambda functions:

- **Transformation Lambda -** Triggers when new raw data arrives and processes it into transformed data.

- **Staging Lambda -** Triggers when transformed data is available and prepares it for processing.

The Aggregator API Instances retrieve staged data, apply logic to the staged data then store it in an AWS RDS database. This data is in the right structure for use by non-technical people. These API instances exist as containers that are auto-scaleable. The volume of incoming data fluctuates, requiring different levels of compute at different times. ECS Auto Scaling ensures instances are only provisioned when needed. A minimum and maximum instance limit can be set to prevent uncontrolled scaling.

At the data consumption side, we have an ECS cluster of client-side APIs that post data to the front-end for visualization and monitoring. Auto-scaling of these API instances is handled in a similar manner to that used to auto-scale the aggregator API instances.

For load balancing on the client-side, we have an AWS API gateway to distribute the client-side traffic amongst the client-side API instances.