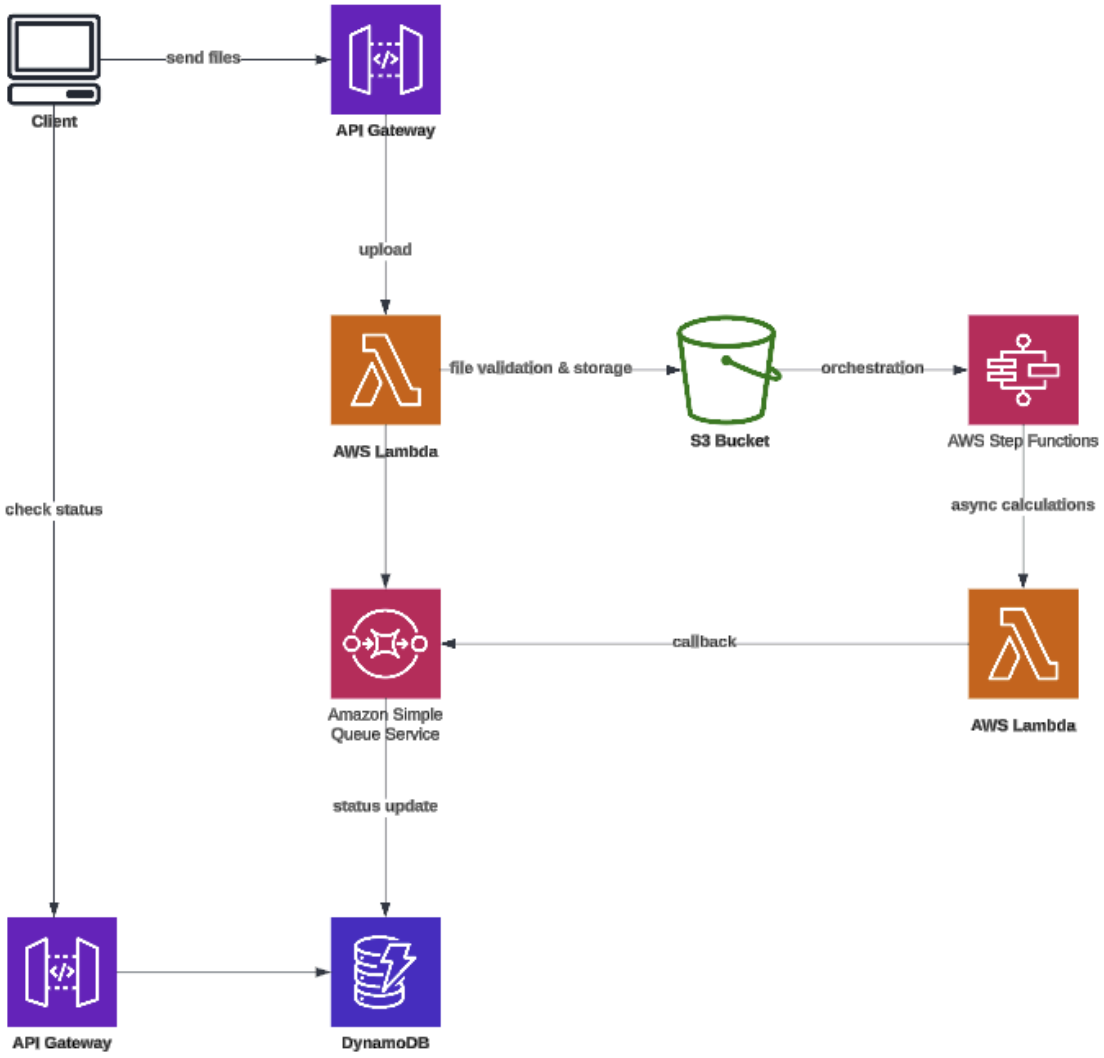
AWS System Design



Explanation of each AWS Component

1- Client API Gateway
API Gateway acts as an entry point, exposing an API for the client to send requests. This API is responsible for receiving excel/csv documents uploaded directly from the client's infrastructure.

2- AWS Lambda
A Lambda function is triggered when the client uploads a document through the API Gateway. The Lambda function validates the document format (Excel or CSV), stores it in an S3 bucket, and generates a unique processing ID for the document.

3- Amazon S3
Once the document is received, it is stored in an S3 Bucket. The document serves as input for further processing.

4- AWS Step Functions
Step Functions orchestrate the workflow for processing the document. Each row in the document triggers a separate calculation of carbon emissions using a Lambda function.
Step Functions allow asynchronous, parallel and scalable processing.

5- AWS Lambda
A dedicated Lambda function performs the calculations for each row in the document. These calculations may take time, but Lambda's scalability ensures multiple rows can be processed concurrently.

6- Amazon SQS
After processing each row, the results are pushed into an SQS queue, which is responsible for delivering the completed document processing results. This ensures decoupling between processing and result delivery.
Using SQS ensures that if the client-side integration is unavailable or there is a high number of requests, the processed results are not lost but queued for later retrieval.

7- DynamoDB
A DynamoDB table tracks the processing status of each document (e.g. 'processing', 'completed'). The client can check this status via API and only retrieve the results once processing is fully complete. The system can use DynamoDB to throttle client queries.

8- Client Status API
The client's infrastructure can poll this API to check the status of the document processing. Once the document is fully processed, the client retrieves the results in bulk, preventing premature queries.
This API ensures that clients only retrieve their results once all processing is complete, saving AWS resource costs by reducing the number of unnecessary executions.

Code Flow and Execution

1- Client Upload
- The client sends a document to the API Gateway, which forwards it to the Lambda function.
- The document is validated and stored in S3. A processing ID is generated, saved in the database and returned to the client.

2- Processing Orchestration
- Step functions are triggered to start processing the document.
- Each row in the document is processed asynchronously using Lambda, calculating carbon emissions row by row.
- The processed data for each row is added to the SQS queue.

3- Status Update and Query
- DynamoDB is updated with the processing status of the document.
- The client can query the status using the Client Status API. The system only allows result retrieval once the document is fully processed.

4- Final Retrieval
- When processing is complete, the client is notified (e.g. via callback or polling the Status API) and can retrieve the full processed document in one API call.