**Introduction**

This document outlines the architecture and implementation of a credit card transaction and claim processing system using AWS services. The system includes AWS API Gateway, AWS Step Functions, and AWS Lambda. AWS Lambda functions interacting with Amazon DynamoDB. The system processes claim by validating credit card transactions, processing the claims, and performing role-based validation.

**Components**

1. **API Gateway**: Handles incoming HTTP requests and triggers the credit card processor directly and also trigger Step Functions state machine for claim processor lambda
2. **AWS Step Functions**: Orchestrates the workflow involving multiple Lambda functions.
3. **Lambda Functions**:
   * Credit Card Processing : validates credit card transactions and logs results to DynamoDB.
   * Claim Processing : Processes claims and logs results to DynamoDB.
   * Role Engine Validation : Validates claims and response to claim processing lambda function logs results to DynamoDB.
4. **Amazon DynamoDB**: Stores transaction and claim processing records.
5. **AWS Serverless Framework / AWS CloudFormation**: Automates the deployment of AWS resources

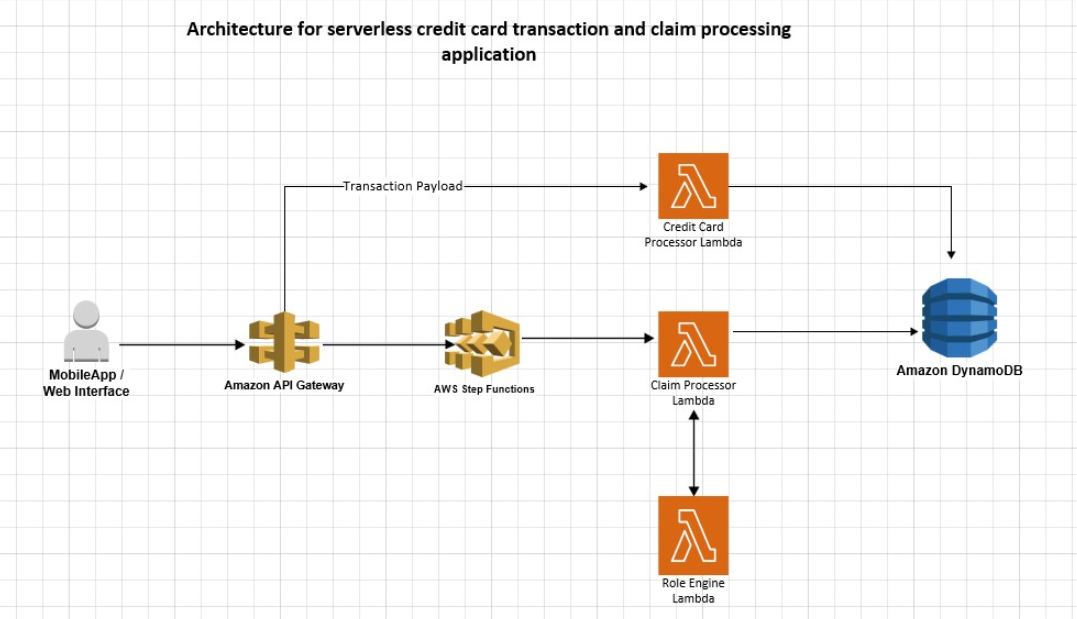
**Workflow**

1. An HTTP POST request or request from Mobile/Web interface goes through the API Gateway transaction payload to the Credit Card Processor Lambda function, which interacts with DynamoDB.
2. API Gateway triggers :
   * + the Step Functions state machine
     + the Credit card Processing lambda function
3. Step Functions executes the following steps:
   * + Invoke the Credit Card Processing Lambda function.
     + If the credit card is approved, invoke the Claim Processing Lambda function.
     + Validate the claim using the Role Engine Validation Lambda function.
     + A request from Mobile/Web interface goes through the API Gateway claim payload to the Step Functions state machine. The state machine invokes the Claim Processor Lambda function, which then invokes the Role Engine Lambda function for validation and responses back to the Claim Processor Lambda function, interacting with DynamoDB.

**IAM Roles and Permissions**

* + - **Lambda Execution Role**: Allows Lambda functions to write logs to CloudWatch and interact with DynamoDB.
    - **Step Functions Execution Role**: Allows Step Functions to invoke Lambda functions.
    - **API Gateway Execution Role**: Allows API Gateway to start Step Functions executions and credit card processor

**Architectural Design**



**Implementation**

**Step 1: Create Lambda Functions**

1. **Credit Card Processing Lambda Function**
2. **Claim Processing Lambda Function**
3. **Role Engine Validation Lambda Function**

**Step 2: Create Step Functions State Machine**

Define the state machine JSON definition:

**Step 3: Create API Gateway**

Create an API Gateway with a POST method to trigger the Step Functions state machine or the Lambda function.

**Step 4: Configure IAM Roles and Permissions**

1. **Lambda Execution Role**

Create an IAM role for Lambda functions:

Attach policies for CloudWatch logging, basic Lambda execution, and DynamoDB access:

1. **Step Functions Execution Role**

Create an IAM role for Step Functions:

Attach a policy to allow Step Functions to invoke Lambda functions:

1. **API Gateway Execution Role**

Create an IAM role for API Gateway:

Attach a policy to allow API Gateway to start Step Functions executions:

**Deployment and Testing**

**CloudFormation Template (YAML)**