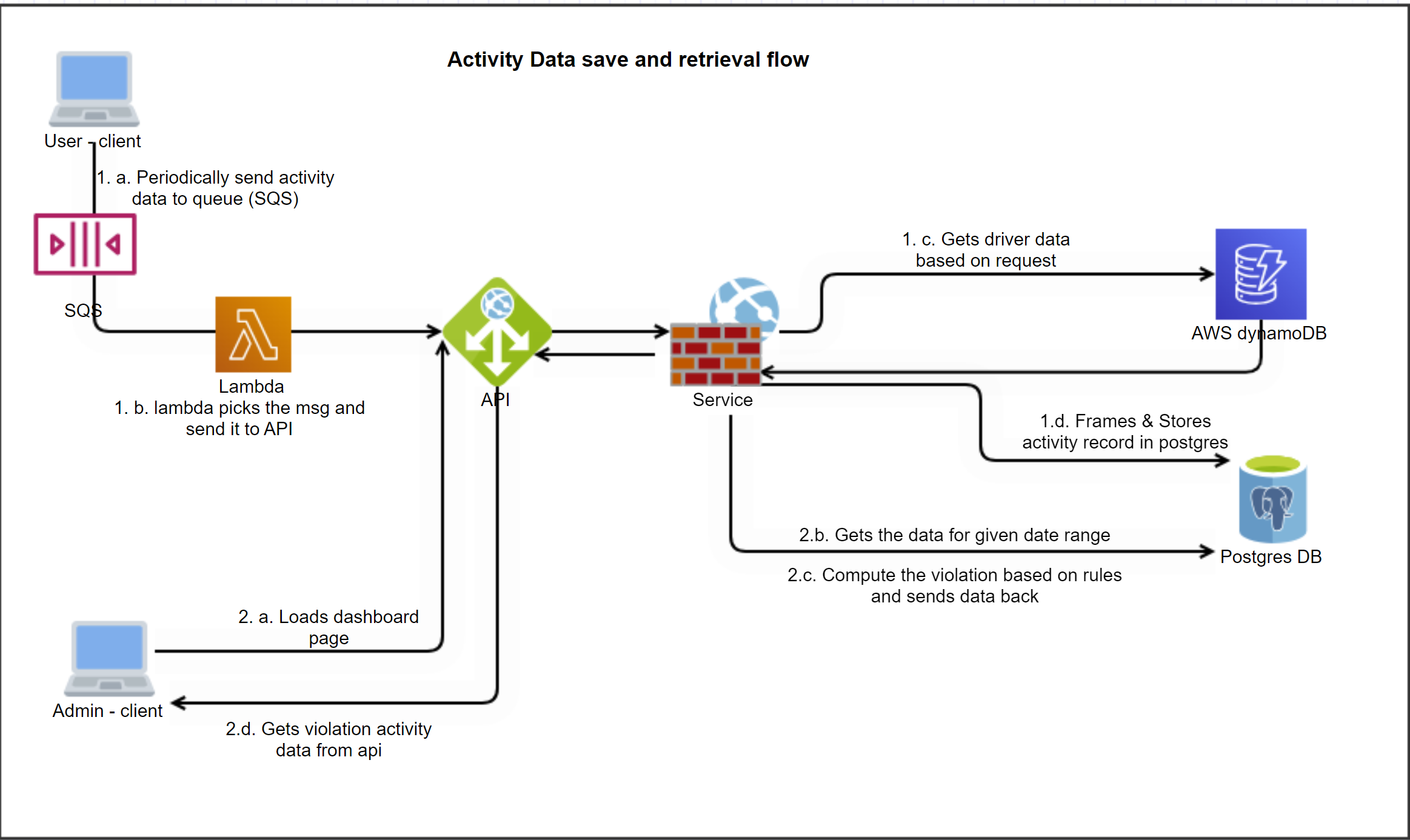
Technical Documentation

Flow diagram:



**Flow explanation:**

1. User activity details are synced to the server periodically
2. The request will be added to a queue (eg: SQS) to avoid flooding at the API
3. There will be a queue processor (eg: Lambda) which picks the msg and does API POST call.
4. The api will cascade the request to a service layer.
5. The service calls DynamoDB to get the driver information and then calls postgres to get/post the Activity details and does a computation and sends it back to client.

**Current Implementation:**

1. The UI (React) has a input to get no of drivers to simulate the data. On click of Start, React will create data for given number of drivers randomly.
2. Once the generation is done, the UI will have an option to validate the simulated data to DB.
3. When validate is clicked, the data is sent to API to validate. This will check various constraints like max no of drivers in the data, max files for driver, etc,. and send the validation response.
4. Once the validation is successful, the UI will have an option to Submit the data to DB.
5. When Save is clicked, the data is sent to API and the service will post the data to PostgresDB.

DynamoDB table structure:

{

"TableName": "DriverDetails",

"KeySchema": [

{

"KeyType": "HASH",

"AttributeName": "DriverId"

},

{

"KeyType": "RANGE",

"AttributeName": "CreationDate"

}

],

"AttributeDefinitions": [

{

"AttributeName": "Id",

"AttributeType": "S"

},

{

"AttributeName": "FirstName",

"AttributeType": "S"

},

{

"AttributeName": "LastName",

"AttributeType": "S"

},

{

"AttributeName": "Address",

"AttributeType": "S"

},

{

"AttributeName": "MobileNumber",

"AttributeType": "S"

},

{

"AttributeName": "Nationality",

"AttributeType": "S"

},

{

"AttributeName": "LicenseNumber",

"AttributeType": "S"

}

],

"BillingMode": "PAY\_PER\_REQUEST"

}

API documentation:

1. Get activities for given date range

HTTP GET **https://{url}/api/driver-activity/get-activity?startDate={}endDate={}**

Parameters:

startDate : DateTime – Required

endDate : DateTime – Required

Sample Response:

{

  "violations": [

    {

      "type": 1,

      "activity": [

        {

          "id": 1,

          "driverId": 123,

          "activityType": "Drive",

          "startTime": "2023-11-29T05:30:40.580Z",

          "endTime": "2023-11-29T05:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

        }

      ]

    }

  ]

}

1. Post activity for given driver

HTTP POST **https://{url}/api/driver-activity/post-file**

Sample Request:

{

          "driverId": 123,

          "activityType": "Drive",

          "startTime": "2023-11-29T05:30:40.580Z",

          "endTime": "2023-11-29T05:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

}

Sample Response:

{

      "Code": 200,

        "Description": "Success",

}

1. Post activities data

HTTP POST **https://{url}/api/driver-activity/post-files**

Sample Request:

[

{

          "driverId": 123,

          "activityType": "Drive",

          "startTime": "2023-11-29T05:30:40.580Z",

          "endTime": "2023-11-29T05:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

},

{

          "driverId": 123,

          "activityType": "Rest",

          "startTime": "2023-11-29T07:30:40.580Z",

          "endTime": "2023-11-29T08:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

},

{

          "driverId": 345,

          "activityType": "Drive",

          "startTime": "2023-11-29T12:30:40.580Z",

          "endTime": "2023-11-29T16:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

}

]

Sample Response:

{ "Code": 200,

“Description": "Success”

}

4. Validate activities data

HTTP POST **https://{url}/api/driver-activity/validate-activities?simulationDate={}**

Parameters:

simulationDate : DateTime – Required

Sample Request:

[

{

          "driverId": 123,

          "activityType": "Drive",

          "startTime": "2023-11-29T05:30:40.580Z",

          "endTime": "2023-11-29T05:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

},

{

          "driverId": 123,

          "activityType": "Rest",

          "startTime": "2023-11-29T07:30:40.580Z",

          "endTime": "2023-11-29T08:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

},

{

          "driverId": 345,

          "activityType": "Drive",

          "startTime": "2023-11-29T12:30:40.580Z",

          "endTime": "2023-11-29T16:30:40.580Z",

          "createTime": "2023-11-29T05:30:40.580Z"

}

]

Sample Response:

{ "Code": 200,

“Description": "Success”,

“Body”:

{

  "isValidSimulation": true,

  "validationFailureType": 0,

  "driverActivityFiles":

[

  {

    "id": 0,

    "driverId": 0,

    "activity": "string",

    "startTime": "2023-11-29T05:50:34.496Z",

    "endTime": "2023-11-29T05:50:34.496Z",

    "createTime": "2023-11-29T05:50:34.496Z"

  }

]

}

}