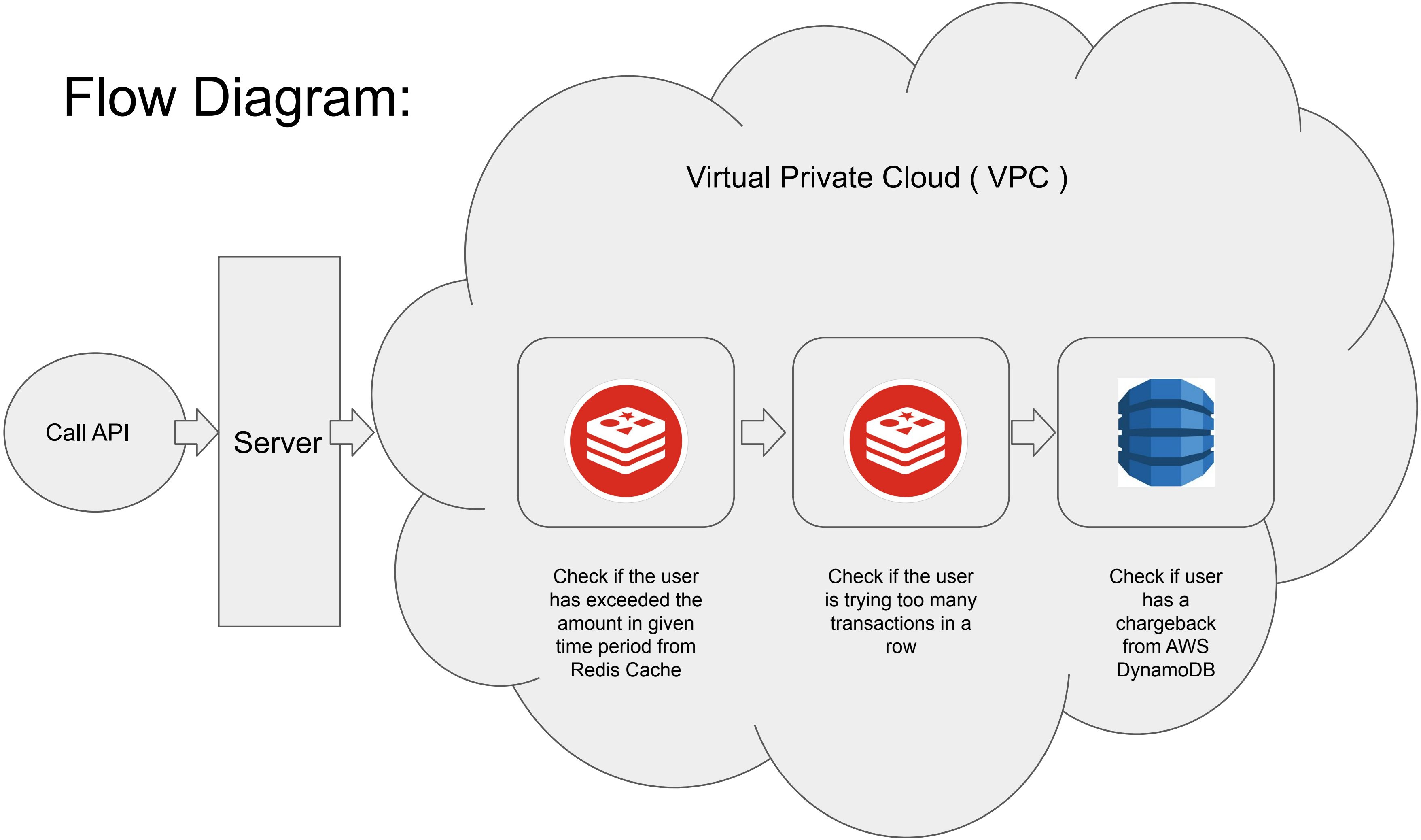
# Anti Fraud System.

# Technology stack used:

(1) NodeJs - 16.13.0 LTS
(2) AWS DynamoDB ( An encrypted key-value pair database. )
(3) Redis Cache ( An in memory cached database. )
(4) Json Web Tokens for encryption of content.
(5) Docker.
(6) Github and Git.
(7) Python libraries like Pandas, Numpy, Matplotlib for statistics.

# Flow Diagram:



Call API

Server

Virtual Private Cloud ( VPC )

Check if the user has exceeded the amount in given time period from Redis Cache

Check if the user is trying too many transactions in a row

Check if user has a chargeback from AWS DynamoDB

# Flow:

(1) The API endpoint is called.

(2) The server connects to Redis Cache to check if the user has exceeded the amount for transaction, if yes, send deny transaction response. else

(3) Check further in cache if the user is transacting too much in a row, if yes, send deny transaction response, else

(4) Connect to AWS Dynamo DB in the ‘transactions_with_chargeback’ table to check if the user has any chargeback, if yes, send deny transaction response, else.

(5) Send an approve transaction response.

# Reasons for using the stack.

(1) Node Js is a javascript runtime that uses Chrome v8 engine.

(2) Redis Cache for reducing the latency, a traditional database call can take upto 2 ms where as a redis call only take upto 0.2 ms.

(3) Encrypted AWS DynamoDB for security of the data.

(4) The whole of production environment can be in a VPC over a VPN connected by a gateway and will be hosted over a VPS ( Virtual Private Server ).

(5) Json Web Tokens for authorizing and authenticating every API call.

(6) Docker for dockerizing the application and making it easy to deploy on any machine.