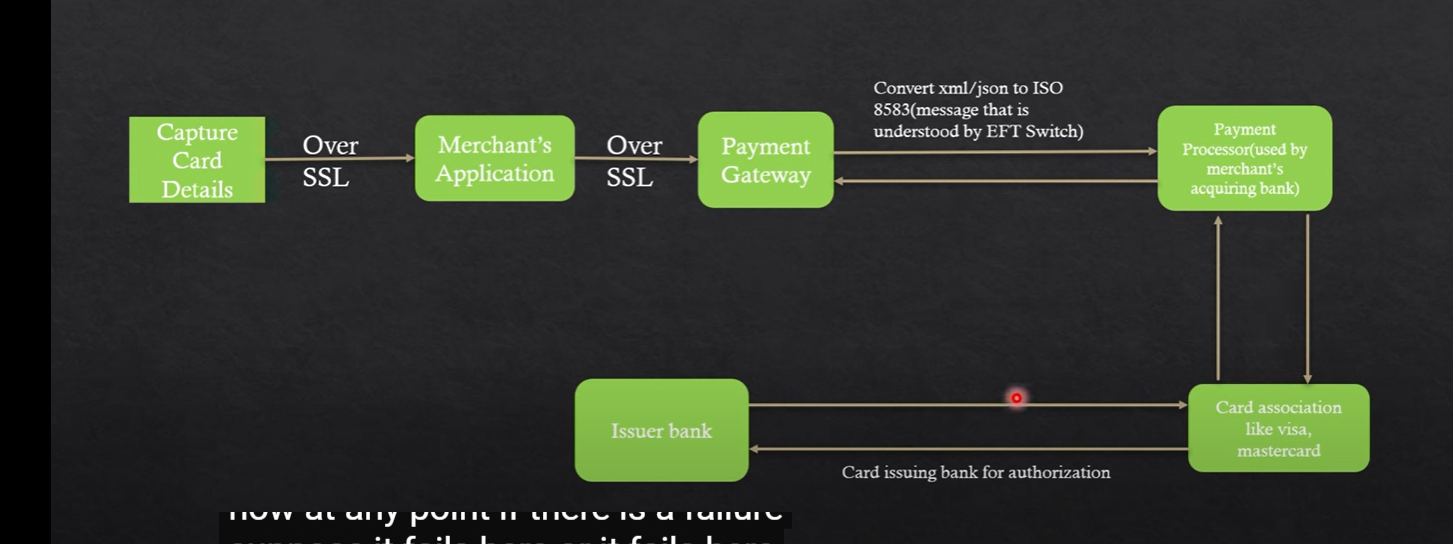
Payment Gateway:-   
A service which allows us to make payment onlone while making purchace from e-commerce website. It supports paying through car, internet banking , e-wallet etc.  
  
**Payment Service Provider :-(PSP)**   
It is the service which makes sure that money is transferred from buyers account to merchant’s account.

**Issuer Bank :-**   
This is the bank which Buyer is related to.

**Payment Card Industry Data Security Standard(PCI DSS)**   
It is a security standard, if a seller is complaint with this standard then payment page can be generated on seller’s page else they need to redirect the request to payment gateway’s payment page, which is complaint with this standard.   
  
ISO-8583 :-  
EFT switch message format for card payment processing.  
  
 

There are several methods of API authentication, such as Basic Auth (username and password) and OAuth (a standard for accessing user permissions without a password).

**In our case, Alrajhi Bank Gateway APIs generates OAuth. client access token using client credentials grant type following FAPI standards.**

**PSU: - Payment Service User  
   
FAPI: - Financial-grade API Security Profile**

**AISP :- Account information service providers**

**PISP :- Payment initiation service providers**

**TPP:- Third Party Provider  
  
  
  
1:- for the Client insertion we need**

To properly send the authorization request using the Push Authorization Request (PAR) endpoint provided, you will need to follow these steps:

1. **Generate a JWT for Client Assertion**: This JWT (JSON Web Token) will be used as a client assertion to authenticate your client application.
2. **Create the Authorization Request**: This will be a JWT that includes all necessary authorization parameters, such as response type, scope, redirect URI, state, etc.
3. **Push the Authorization Request to the PAR Endpoint**: You will send this authorization request to the PAR endpoint to receive a request\_uri.

### Step-by-Step Instructions

#### Step 1: Generate JWT for Client Assertion

First, you need to create a JWT for client authentication. This JWT should be signed using the client’s private key. Here's an example of the JWT header, payload, and signature.

Authentication Request Jwt or for request : -eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNwb25zZV90eXBlIjoiY29kZSIsImNsaWVudF9pZCI6IjBhOTk1M2FlNWExMDZiNDdkZDkxNTc5ODFjNmNkMGFmIiwicmVkaXJlY3RfdXJpIjoiaHR0cHM6Ly9lc2NhcGVzcGFvcGVyYXRpb25hbHBhbmVsLndlYi5hcHAvZGFzaGJvYXJkIiwic2NvcGUiOiJvcGVuaWQgYWNjb3VudHMiLCJzdGF0ZSI6Inh5eiIsIm5vbmNlIjoiYWJjIn0.EoXN2SjJOlqg1Dg3eYAt9IZKWkpiSzr6EEFCLo0jF9M  
  
  
JWT for client insertion :- eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiIwYTk5NTNhZTVhMTA2YjQ3ZGQ5MTU3OTgxYzZjZDBhZiIsInN1YiI6IjBhOTk1M2FlNWExMDZiNDdkZDkxNTc5ODFjNmNkMGFmIiwiYXVkIjoiaHR0cHM6Ly9hdXRoLmFscmFqaGliYW5rLmNvbS5zYS9hdXRoL3JlYWxtcy9zYW5kYm94IiwiZXhwIjoxNjE0Nzg4NDAwLCJqdGkiOiJ1bmlxdWUtand0LWlkIn0.uumwCVJ9W5drCC5AGyzvQKEmHNIDd8MWi5xs2iwq27g

Checkout.com Payment Gateway

Sand box : - for testing environment

Authentication :-  
When you sign up for an account, you can authenticate with either Access keys (OAuth 2.0), or Secret API keys. Unless explicitly stated, all endpoints require authentication using either your Access or Secret API Keys. Public keys should only be used in JavaScript or native applications.

* **Client-side authentication**. Use your [**public key**](https://api-reference.checkout.com/#section/Authentication/ApiPublicKey) for client-side authentication. It only has access to a limited set of our APIs – mostly those called as part of your payment environment.
* **Server-to-server authentication**. Use your [**secret key**](https://api-reference.checkout.com/#section/Authentication/ApiSecretKey) or [**OAuth**](https://api-reference.checkout.com/#section/Authentication/OAuth) for server-to-server communication. Support for API keys depends on the endpoint. Never share your OAuth credentials, API keys, or access tokens. Keep them guarded and secure.

Public key and Secret Keys :-   
  
 **Obtaining API Keys:** API keys are typically issued by the service provider when you register your application. Here’s how you usually get them:

* **Account Registration:** Sign up for an account with the API provider.
* **Application Creation:** Create a new application within the provider’s developer console.
* **API Key Generation:** The provider generates and displays the API key for your application.

 **Storing and Using API Keys:** API keys should be stored securely and used in your application to authenticate requests to the API. It’s important to keep them confidential and avoid exposing them in client-side code or version control systems.   
  
Access key id:-

ack\_lhmfakpwnr3uzhkqgek3plerxa  
Access key secret :-

jivqOd0yEfg8BNWjfcXuDYcr48cn\_Qy3Fm4p8R5yD63pzIi4tOQ\_vfZue0P8KbcA0A7LlkNKpgofifu\_KwtA

curl --location --request POST 'https://access.sandbox.checkout.com/connect/token' --header 'Content-Type: application/x-www-form-urlencoded' --header 'Authorization: Basic base64(<ack\_fnupsghafidetipxapyvj4hykm>:<jivqOd0yEfg8BNWjfcXuDYcr48cn\_Qy3Fm4p8R5yD63pzIi4tOQ\_vfZue0P8KbcA0A7LlkNKpgofifu\_KwtA >)' --data-urlencode 'grant\_type=client\_credentials'