

Lecture 23: Build Payment Gateway System

Page No. _____
Date _____

To ideas/
objects

Requirements

- should support multiple providers (paytm, Razorpay)
- We can easily add new gateways in future
- There should be a standard payment flow, with required validation.
- Have error Handling and Retries Mechanism

H/W - for additional features

→ There can be suff different ways /strategies of retry mechanism. (Handle them in code)

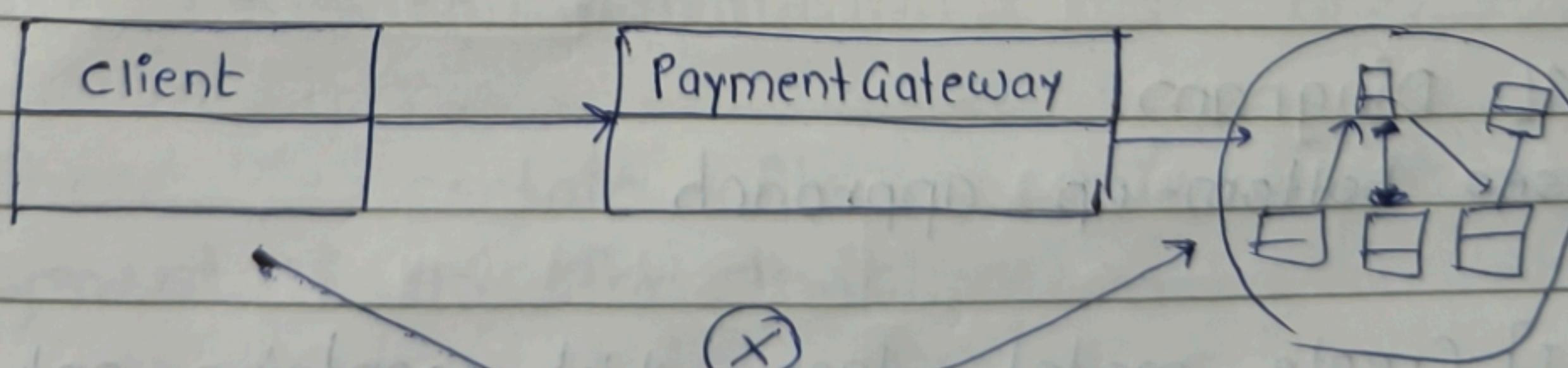
→ Linear Retry

→ Exponential Back-off

→ Try adding recurring payment flow /Subscription

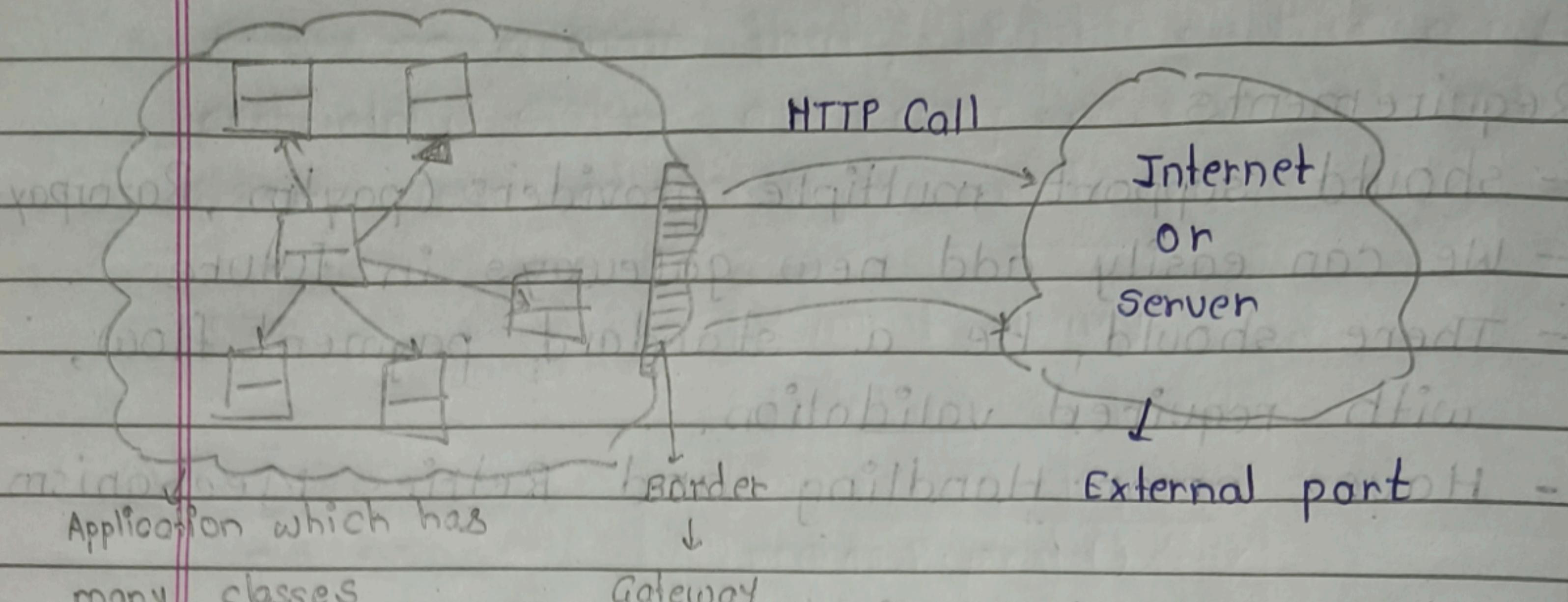
Happy Flow

- Our application should be based on plug and play model
- Client sirf payment gateway se interact karna chahiye usse internal details nahi maalum chahiye.



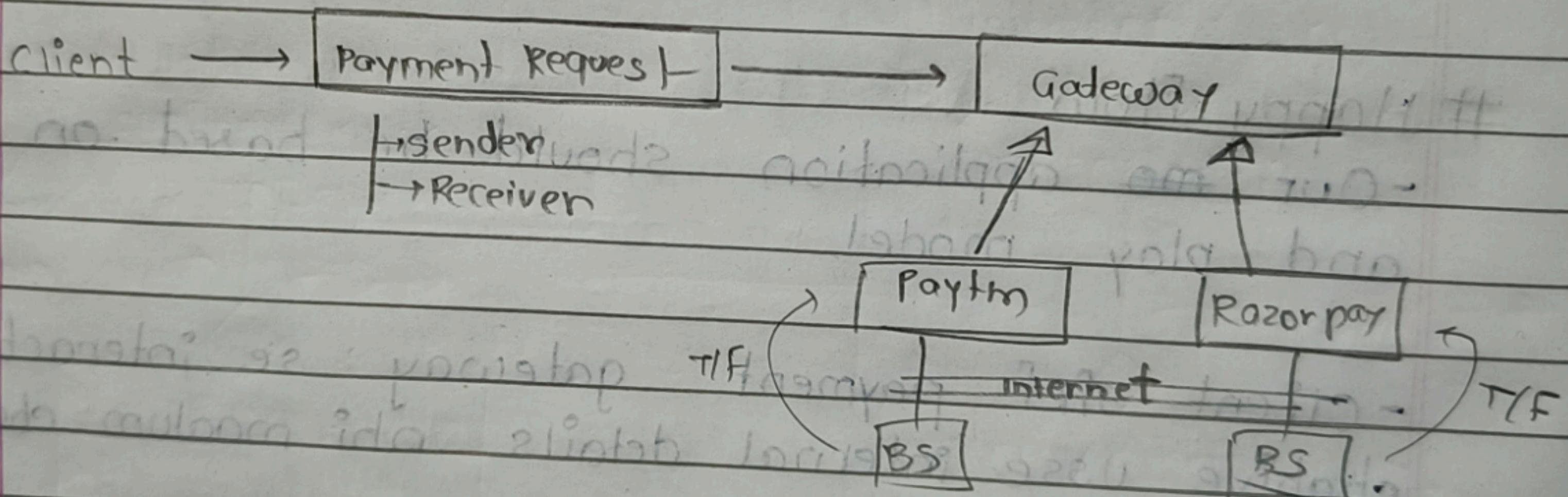
Let's understand what Gateway means

- Humare application mein ek border hai jo responsible honga http calls karna to external world and that border is known as gateway.



→ Internal classes interact with gateway. Further gateway make calls to bank. The bank gives response to gateway & gateway gives to classes.

Flow



UML Diagram

- uses bottom-up approach

S-I] Create modal class which contains only basic information

Payment Requests

String sender;

String receiver;

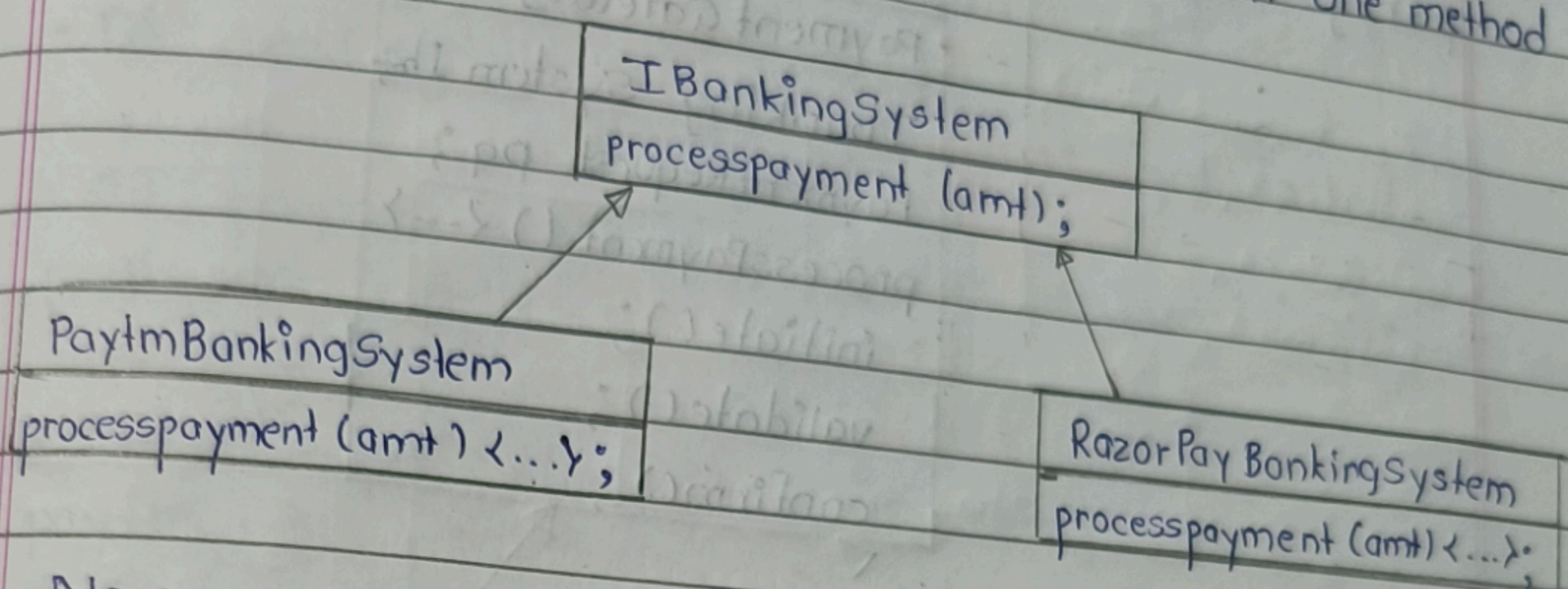
double amt;

String currency;

|| Getter & Setter |

→ will add in code

- Ab hum banking system through internet so humme banking system banana padega ab hum us mein one method - processing payment.



- Now we will create gateway which work btw banking system and payment request.

- Now why need proxy for having interaction with banking system

→ The object of `IBankingsystem` can't be stored as reference as it exists on external servers, as it is not part of our application.

∴ We will need to take it a proxy ie.

Remote proxy which we take will talk / interact with Banking system.

- There is certain flow while making ~~pattern~~ payment & so for that we use template method in which we define the steps -

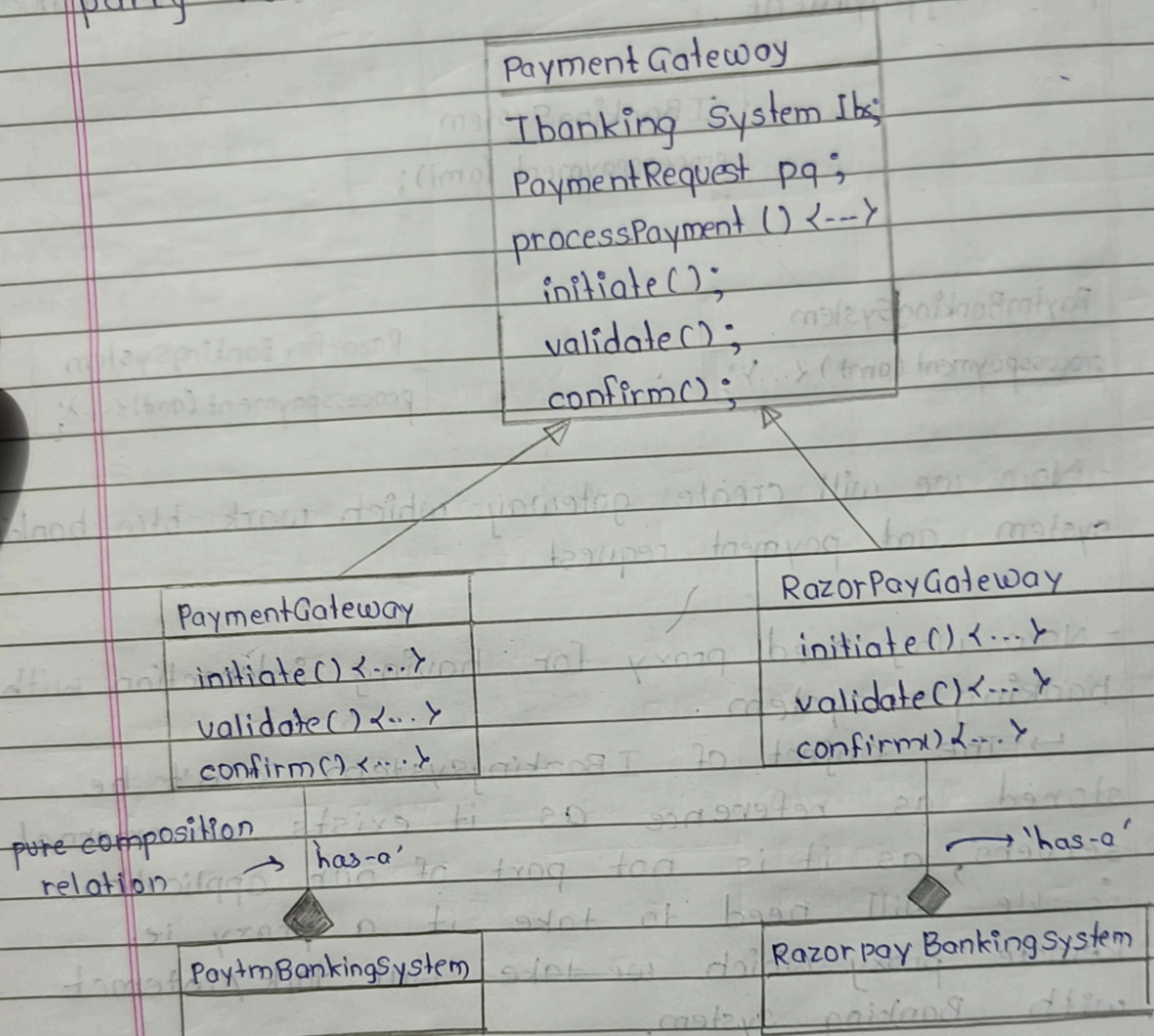
`initiate()`

`validate()`

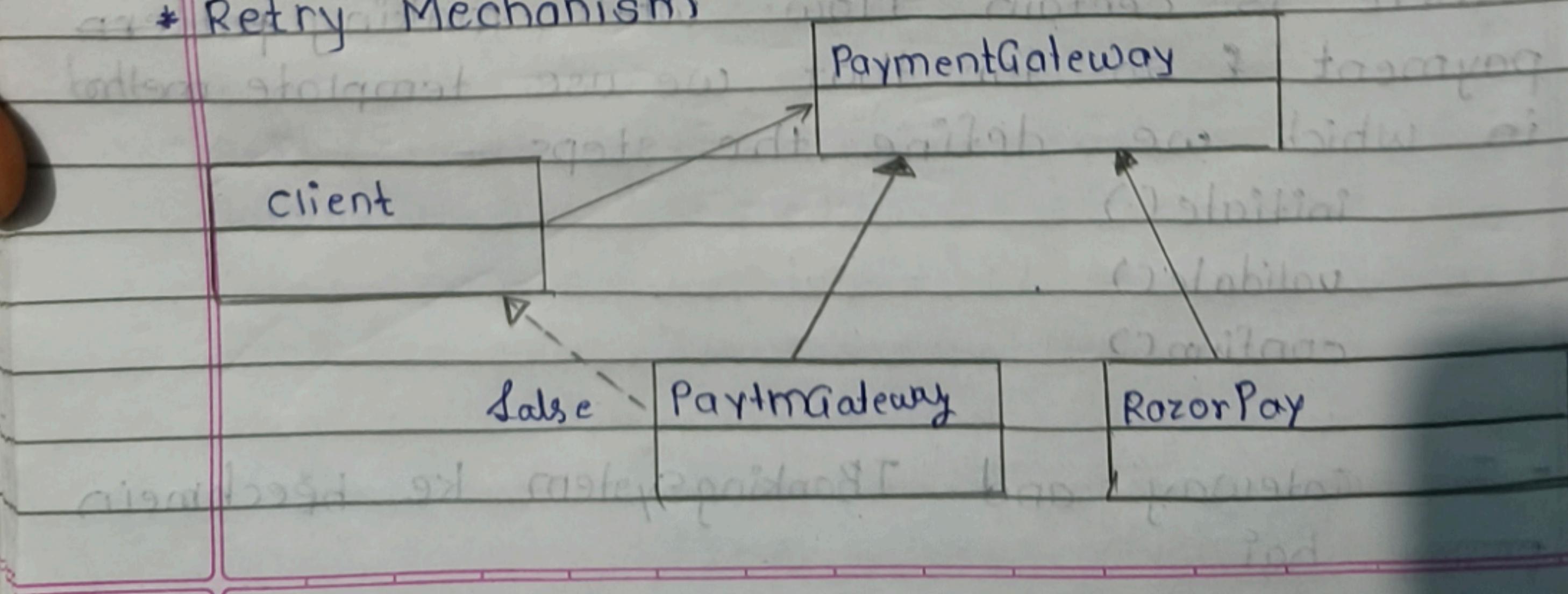
`confirm()`

- Ab Gateway and `IBankingSystem` ke beech mein proxy hai

- We can assume that, Banking hierarchy completely works as proxy means it calls to 3rd party vendors through Internet.



* Retry Mechanism



- Let's assume a flow
→ client interacts with paymentGateway. And now paymentGateway goes to suppose paytmGateway to initiate(), validate(), confirm().
- At confirm() stage payment gets failed.
- So, paytm gateway replies false to client as payment failed.
- Now, it gets dependent on retry mechanism and paytm or razorpay or any other class has different mechanism/methods to retry. For that, we have to make changes in client.

2nd approach is to provide validation retry mechanism at validation step

→ But this too doesn't work as we have to change the retry mechanism every time we come to child or other concrete classes

- So, to avoid this we apply Proxy Design Pattern in Payment Gateway.
But which type of proxy?

- Virtual X → becz we are not creating any expensive object
- Protection ✓
 - ↳ used becoz we are protecting the paytm Gateway, RazorPay Gateway by making validation
- Remote X

Payment
~~PaymentGatewayProxy~~

PaymentGateway
real;

processPayment()
initiate();
validate();
confirm();

Payment Gateway

PaytmGateway

Razorpay Gateway

→ Before calling the real concrete class, proxy gateway
make sure that it should validate retry mechanism.

→ Create Gateway Factory. It will return a proxy object
and internally it will pass paytm ~~or~~ or razorpay
«singletons»

Gateway Factory | 'has-a' | Gateway TYPE

createGateway
(Gateway type g)

PayTM

RazorPay

Final UML Diagram

