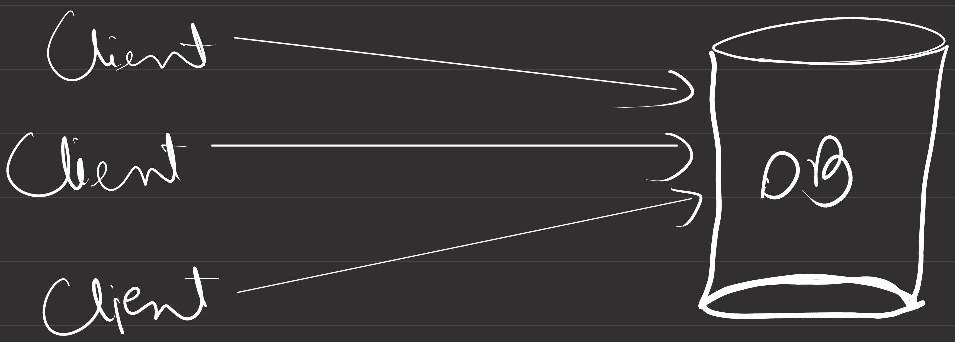
Proxy Design Pattern

A proxy is a structural design pattern, which lets you provide a substitute or a placeholder for another object.

A proxy controls access to other objects, allowing some operation to perform in between.

What problem does it solve?

let's say we have an app, where client request comes in and asks for data. Now the problem is the data querying is very slow from DB and expensive

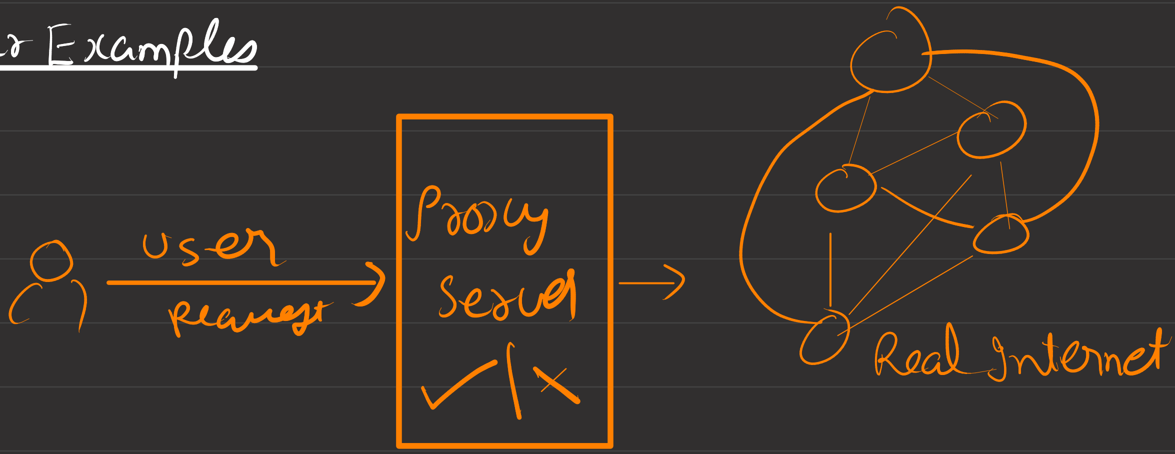


Now the problem arises if these client calls are recurring in a small span of time, it may overwhelm the sytem.

So here we can use the proxy design pattern to have a cache in place and return the data if it has already.

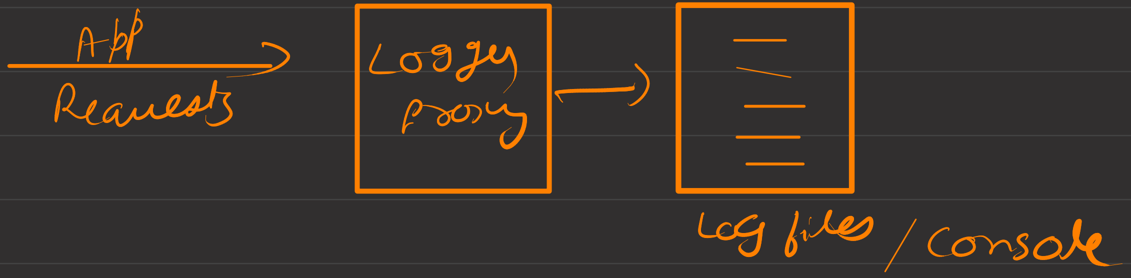
Other Examples

①

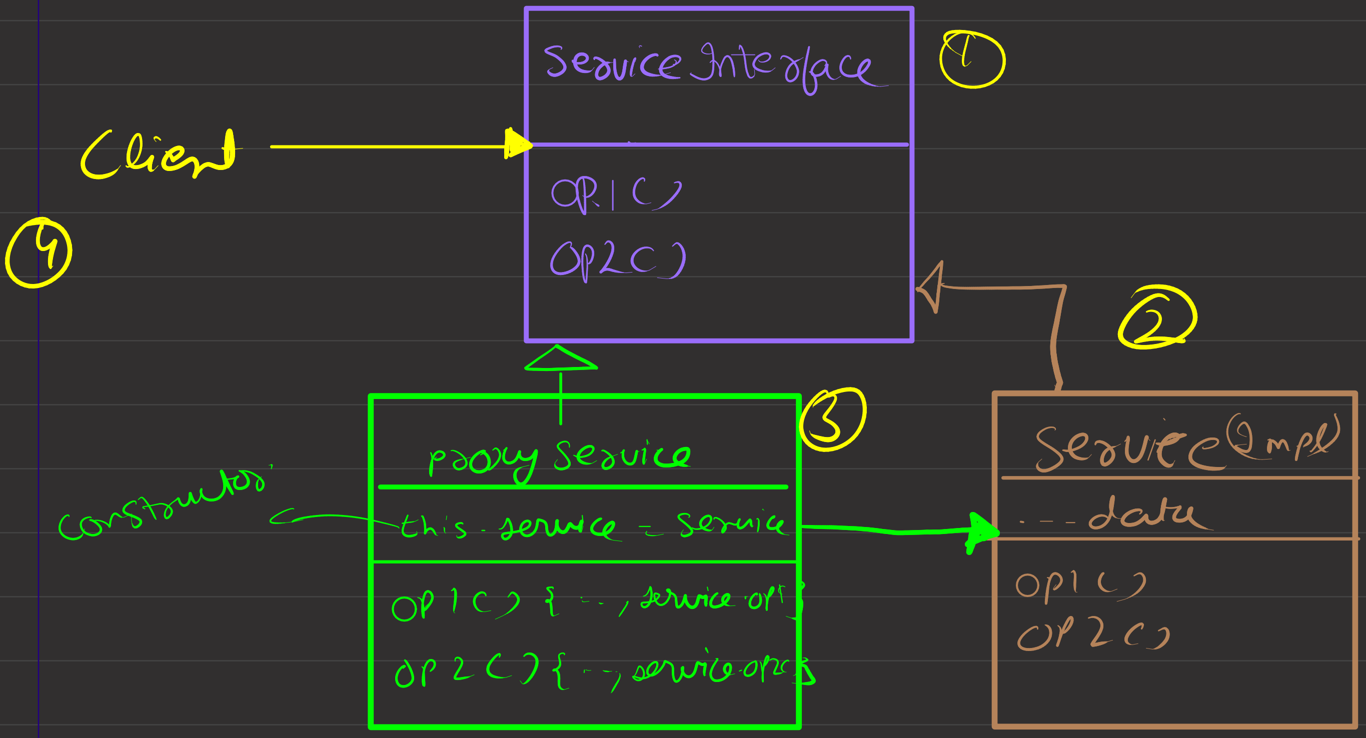


We can build a proxy server based on some whitelisting logic to filter access to internet

②



General Structure



1 Service Interface :- We make an interface for the service we wish to use

2 service Class :- concrete class of service interface.

3 Proxy Service :- This is a concrete class of the interface which has instance of the concrete service class and uses this to perform actions

4 Client :- It uses the service interface to interact with both proxy & concrete objects.