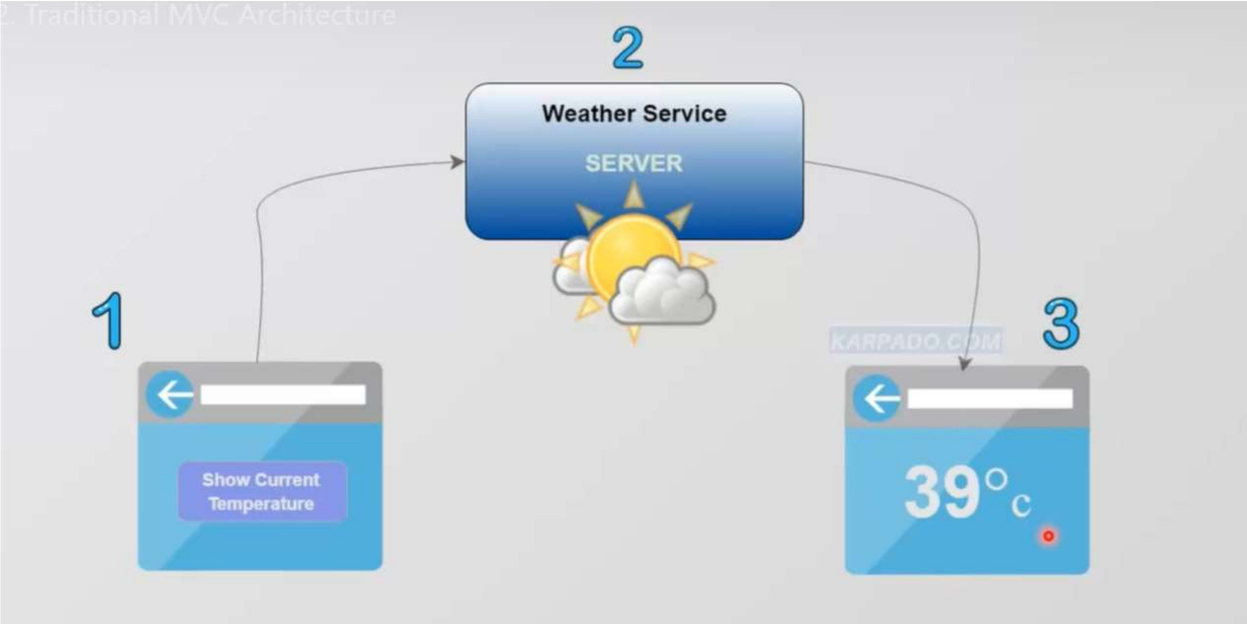
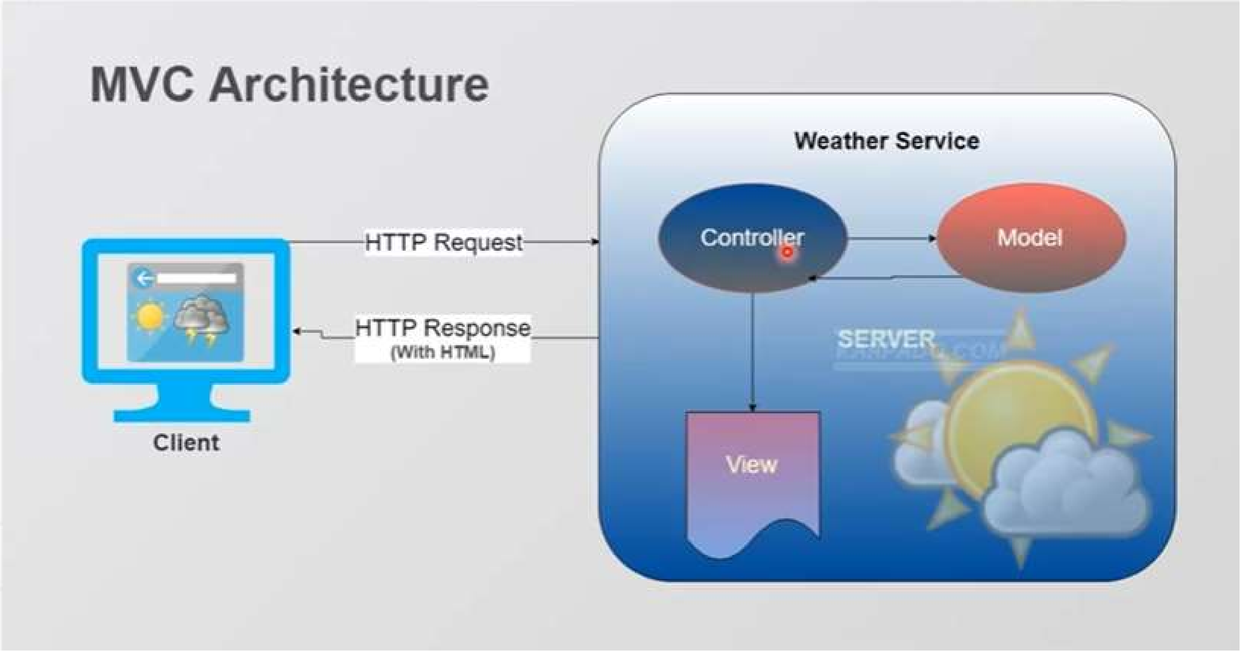
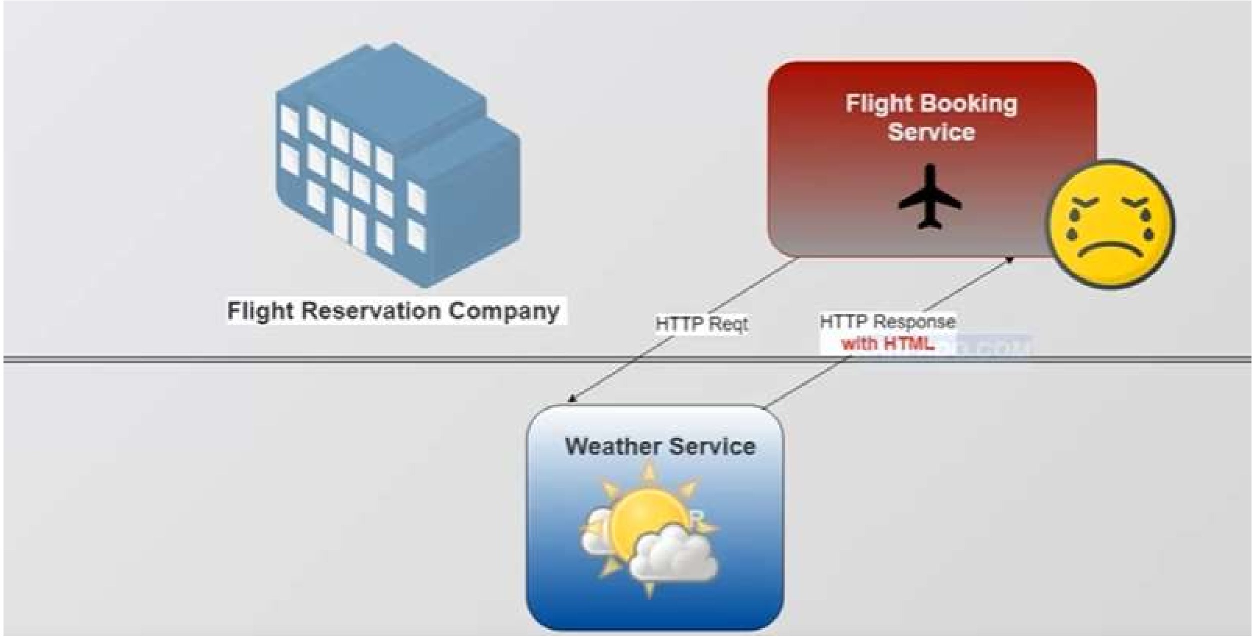
Monolithic MVC Architecture

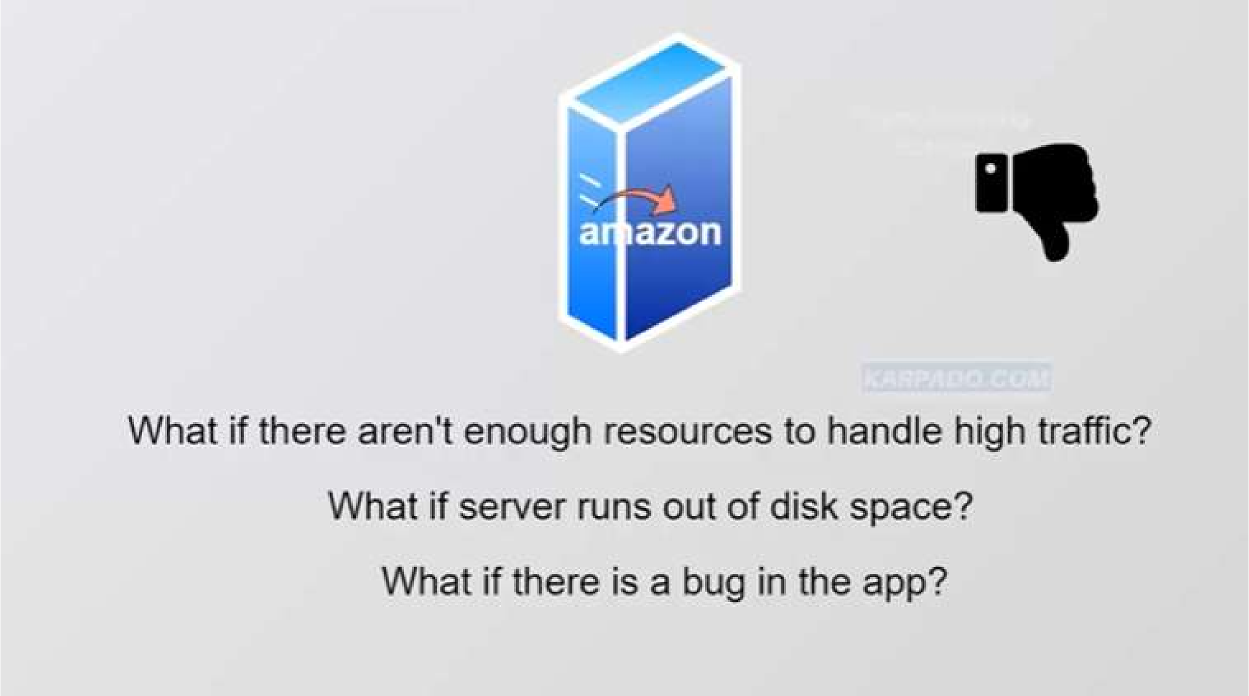




MVC Architecture

Challenges with Monolithic Applications





KARPADO.COM

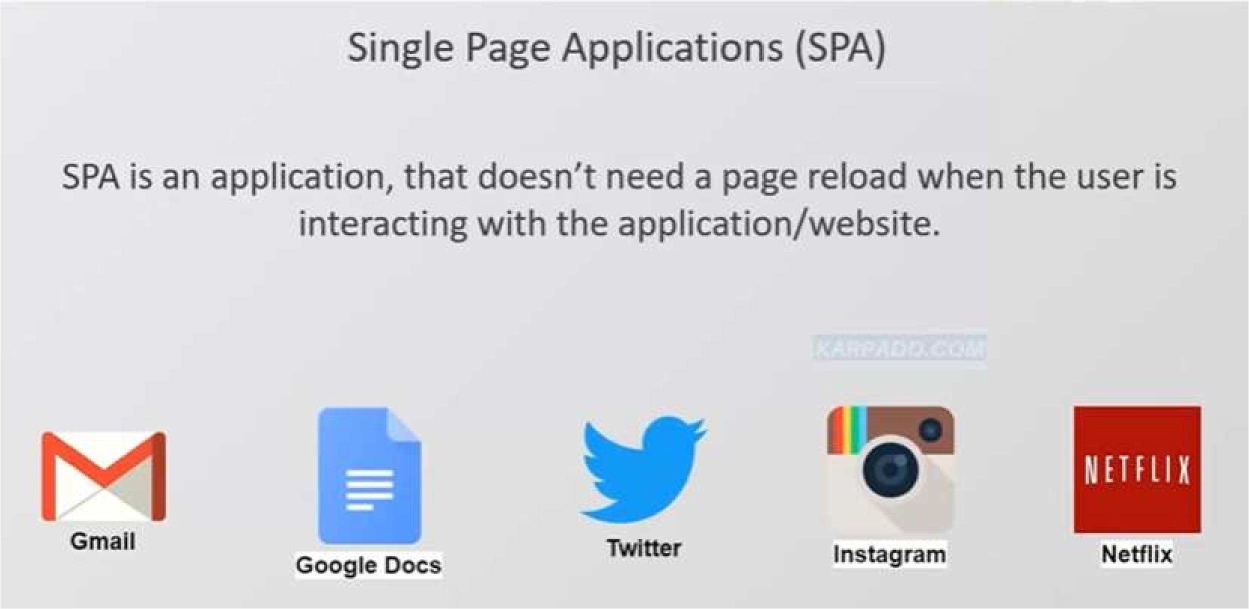
What if there aren't enough resources to handle high traffic?

What if server runs out of disk space?

What if there is a bug in the app?

Order Service
amazon
Email Service
amazon
Product Service
amazon

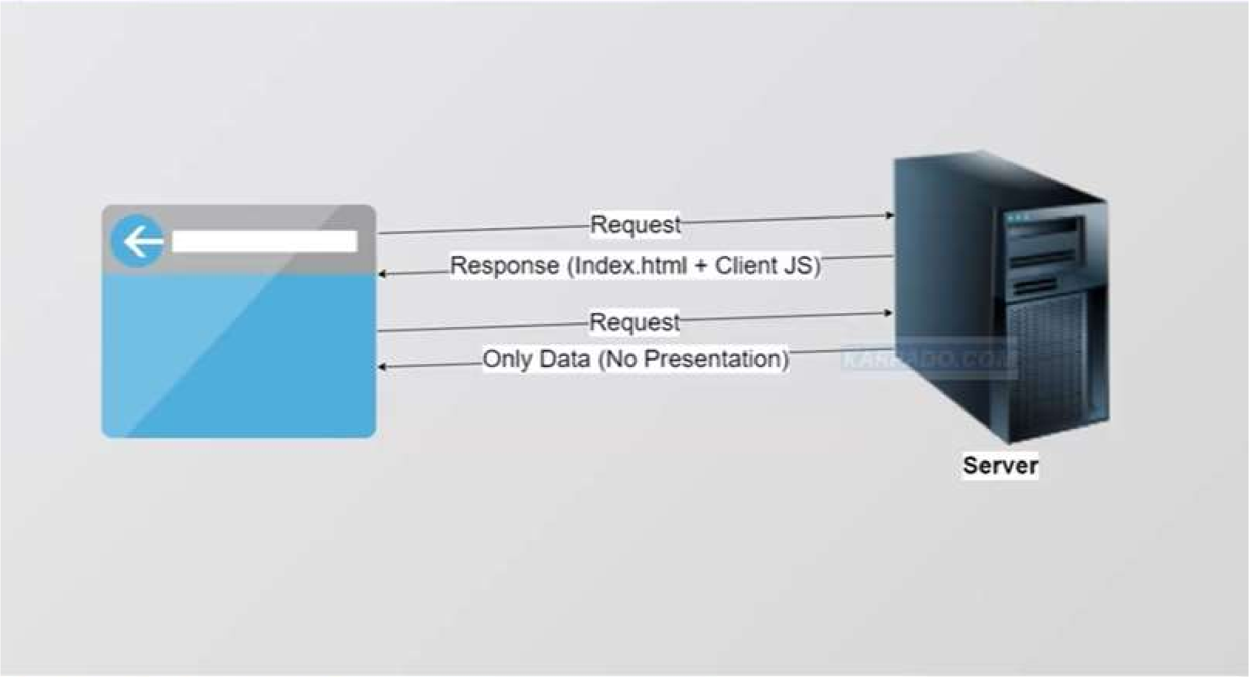
KARPADO.COM

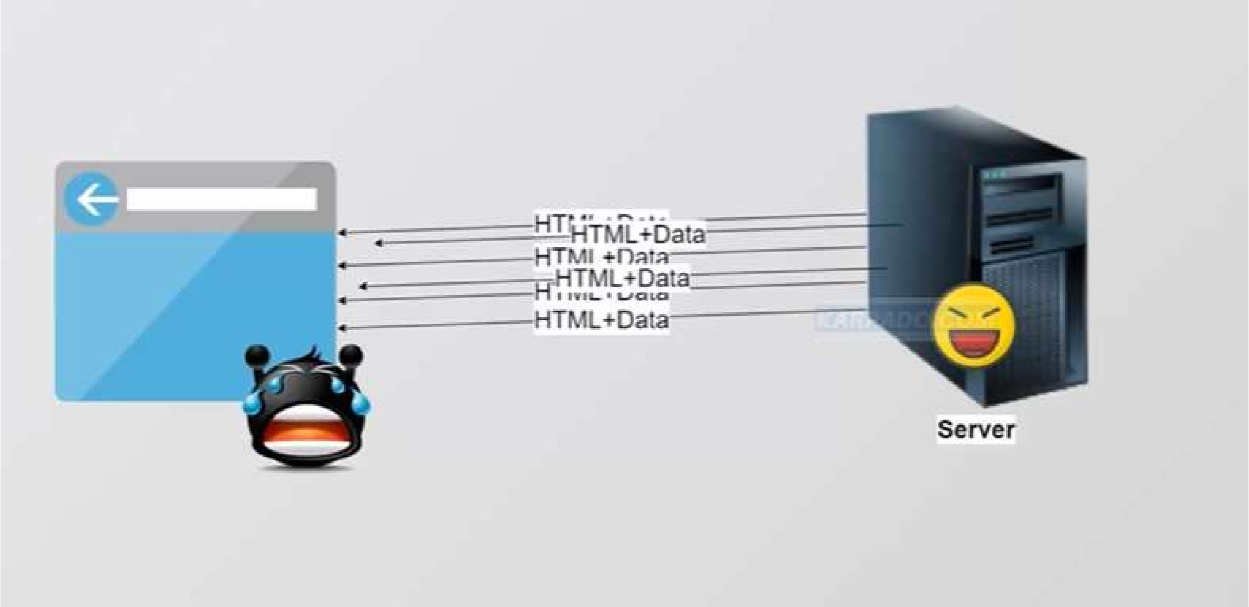


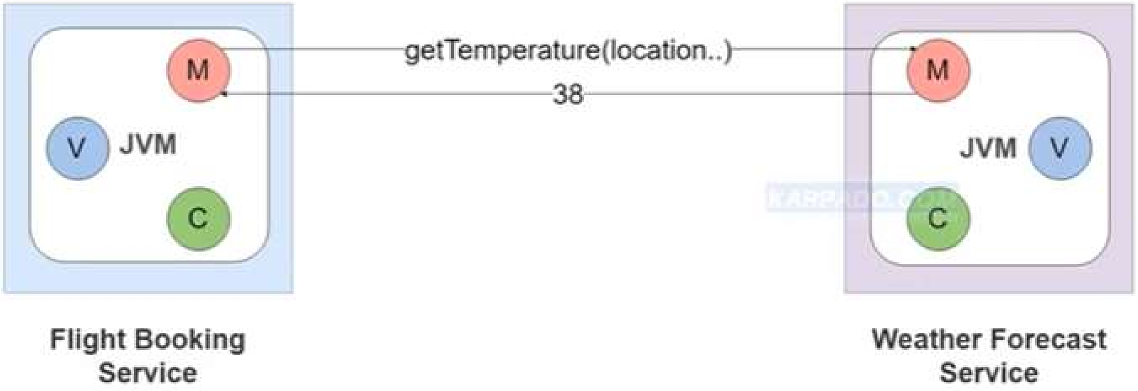
Single Page Applications (SPA)

SPA is an application, that doesn’t need a page reload when the user is interacting with the application/website.

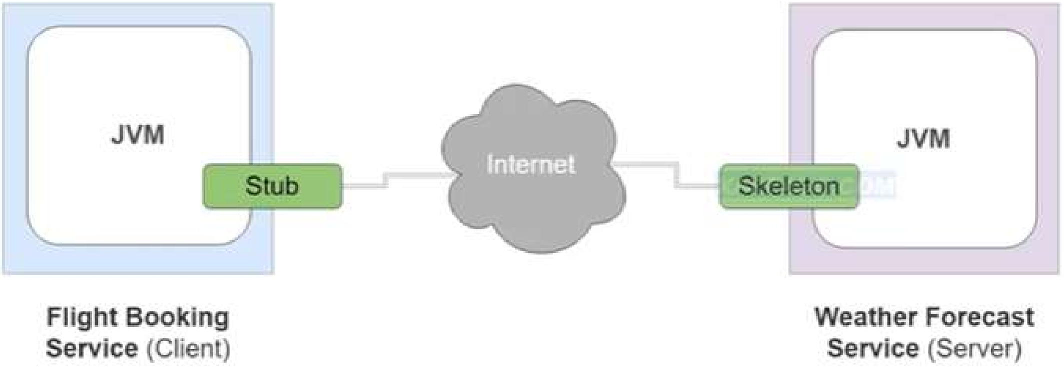
KARPADO.COM

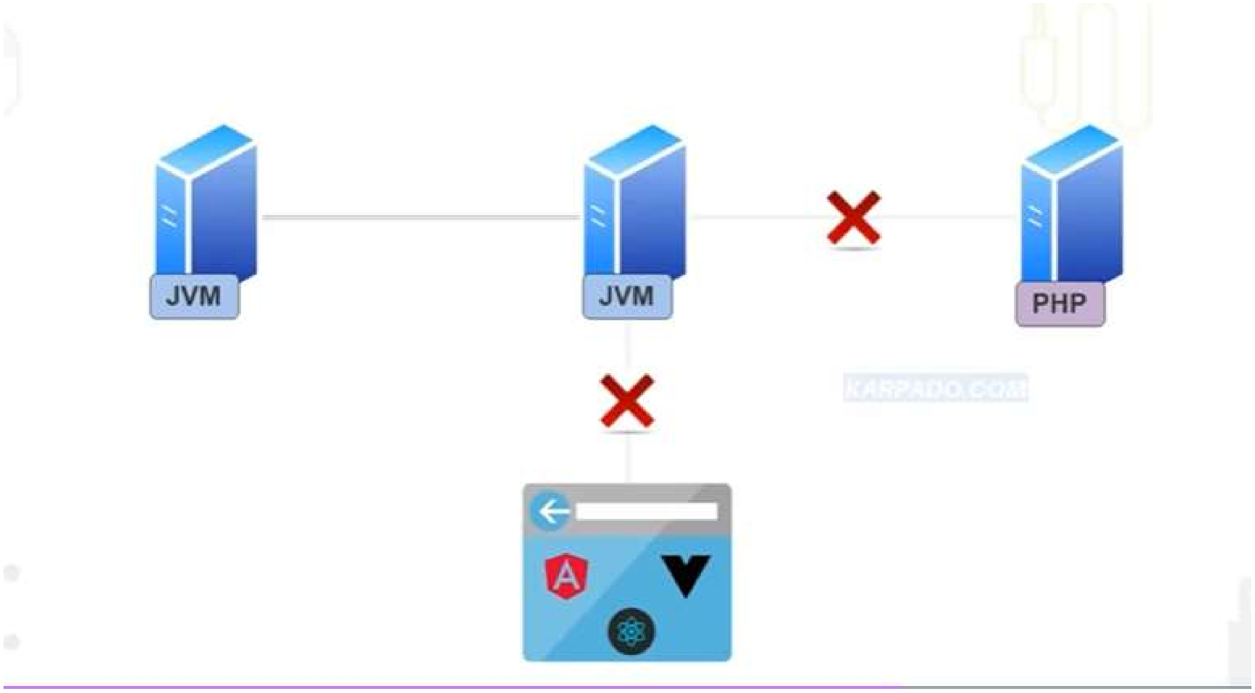

Request
Response (Index.html + Client JS)
Request
Only Data (No Presentation)
Server

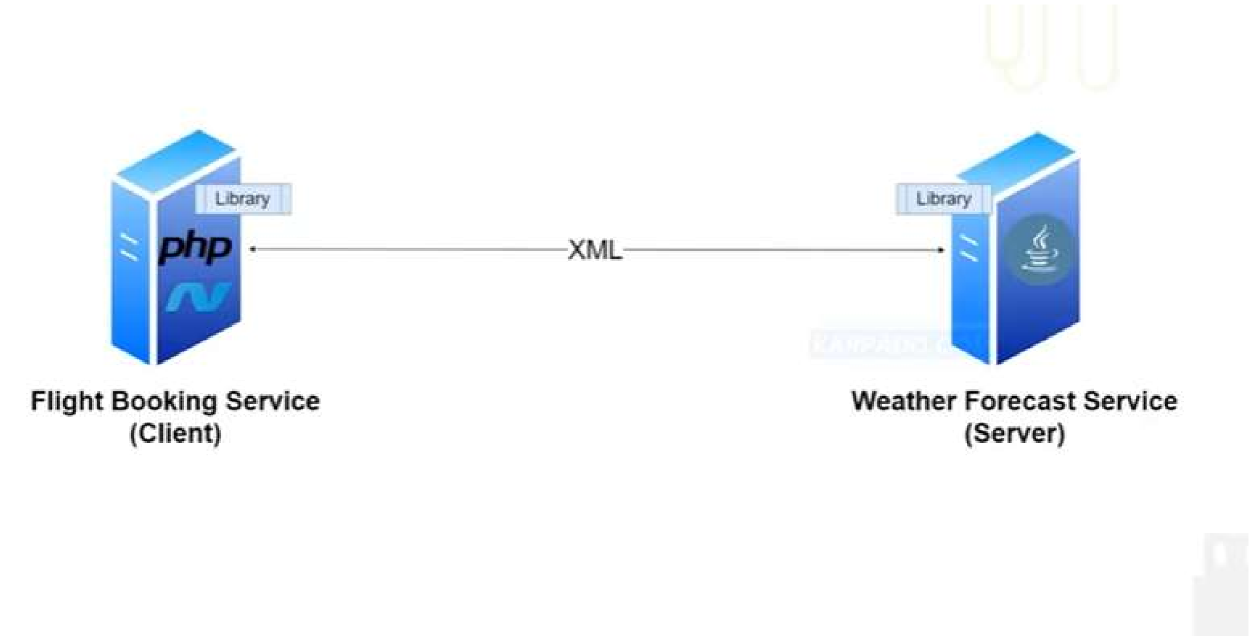

HTML+Data
HTML+Data
HTML+Data
HTML+Data
HTML+Data
HTML+Data
Server



Remote Method Invocation (RMI)


JVM
Stub
Internet
Skeleton
JVM
Flight Booking Service (Client)
Weather Forecast Service (Server)


JVM
JVM
PHP


Library
php
XML
Library
Flight Booking Service
(Client)
Weather Forecast Service
(Server)

SOAP Message Format

```
<env:Envelope>
        <env:Header>
        </env:Header>

        <env:Body>
                <location>
                        <name>MyLocation</name>
                        <temperature>35</temperature>
                </ns2:location>
        </env:Body>
</env:Envelope>
```

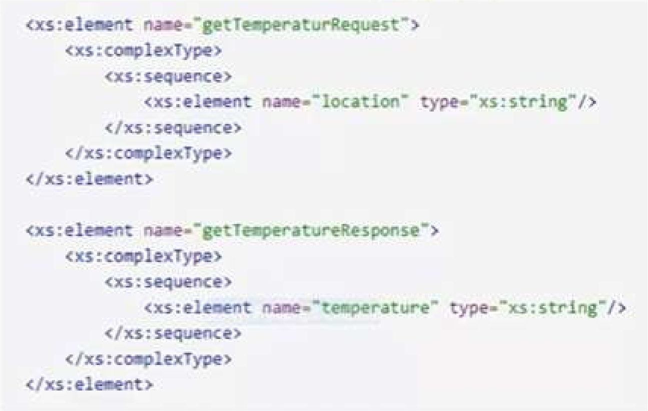
Spring-WS

Exposing Weather Forecast Service

- Create XSD File
- Use JAXB to create Domain Objects using XSD File
- Create an Endpoint and send response using Domain Objects
- Spring-WS will create and expose WSDL (Accessible over internet)

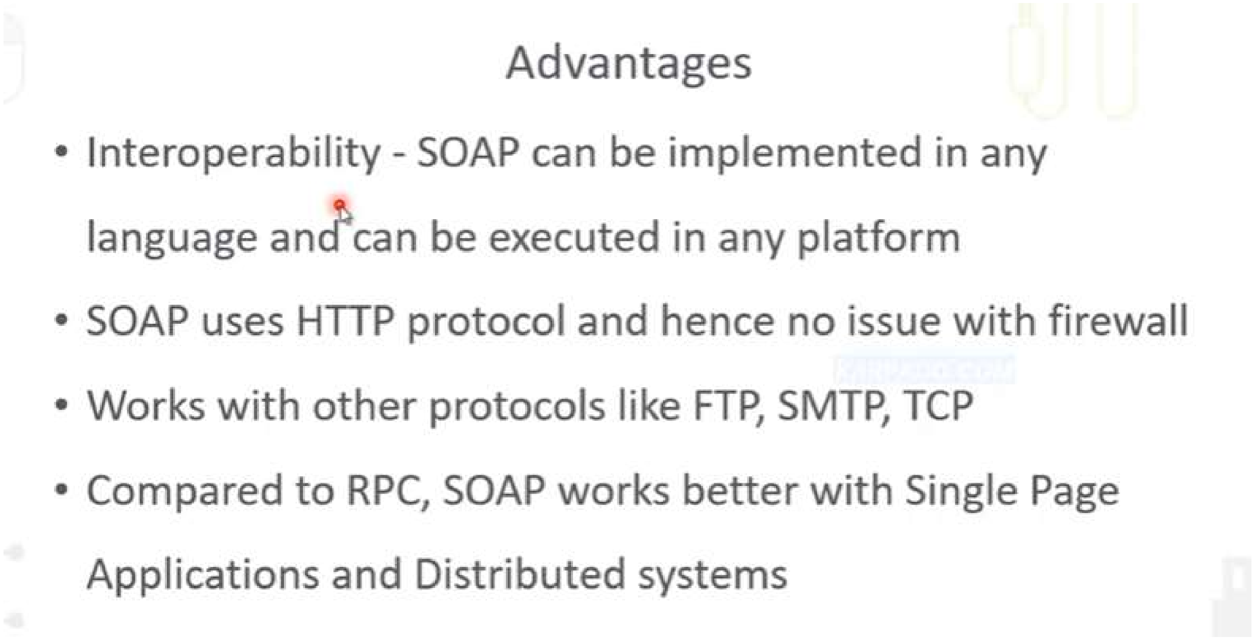
Consuming Weather Forecast Service

- Use JAXB to create Java classes using WSDL file
- Use request/response domain classes to send the request and receive response



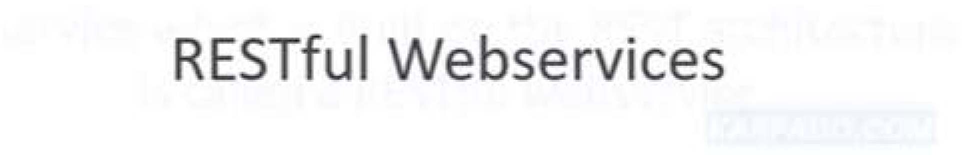
```
<xs:element name="getTemperaturRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="location" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="getTemperatureResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="temperature" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

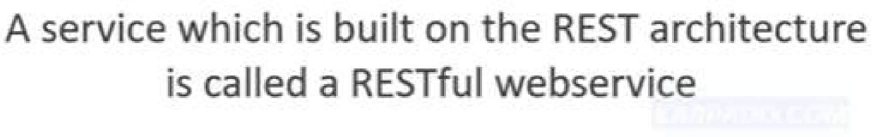


Advantages

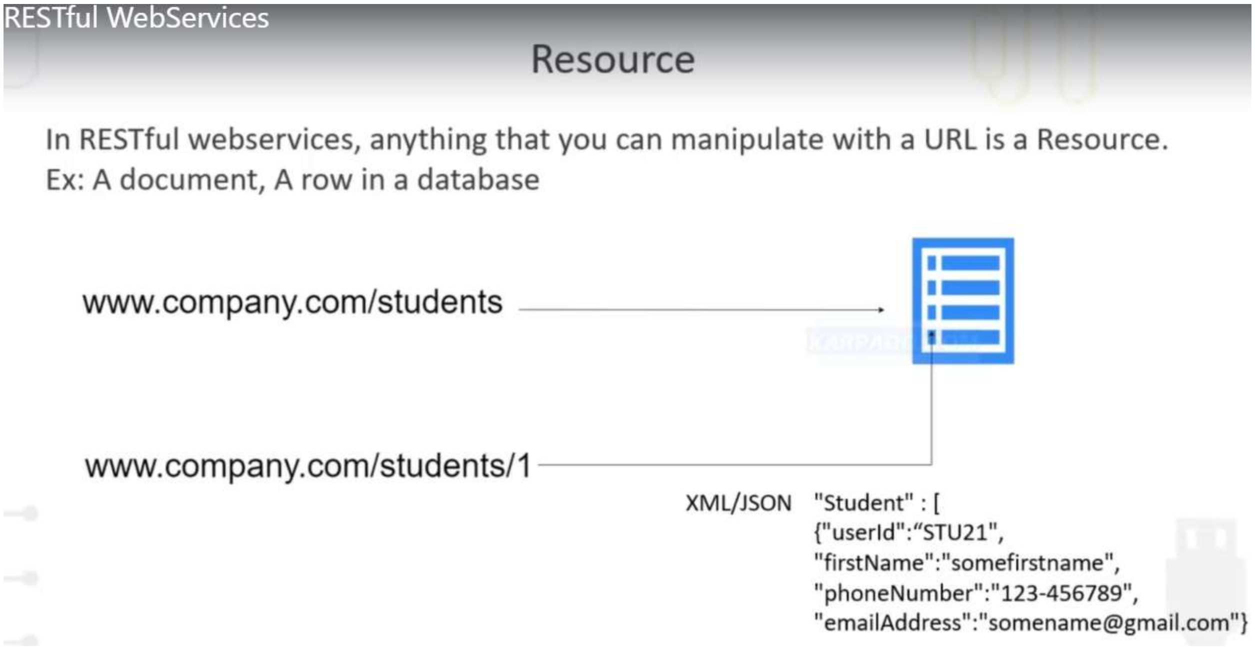
- Interoperability - SOAP can be implemented in any language and can be executed in any platform
- SOAP uses HTTP protocol and hence no issue with firewall
- Works with other protocols like FTP, SMTP, TCP
- Compared to RPC, SOAP works better with Single Page Applications and Distributed systems



RESTful Webservices



A service which is built on the REST architecture is called a RESTful webservice



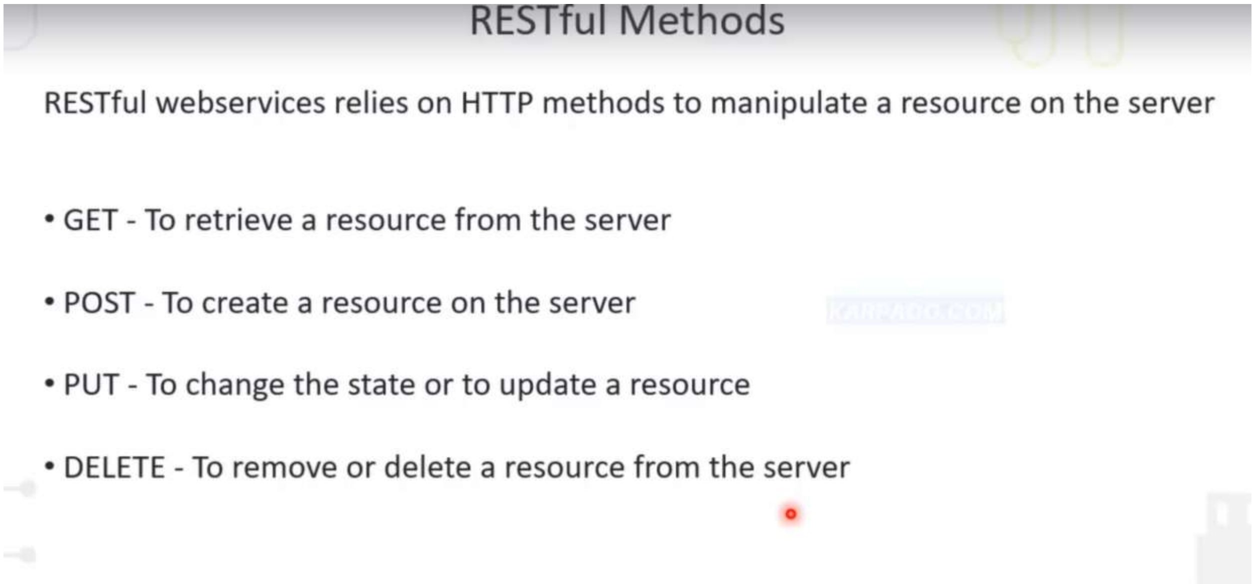
RESTful WebServices

Resource

In RESTful webservices, anything that you can manipulate with a URL is a Resource.
Ex: A document, A row in a database

XML/JSON

```
"Student" : [
{"userId":“STU21",
"firstName":"somefirstname",
"phoneNumber":"123-456789",
"emailAddress":"somename@gmail.com"}
```



RESTful Methods

RESTful webservices relies on HTTP methods to manipulate a resource on the server

- GET - To retrieve a resource from the server
- POST - To create a resource on the server
- PUT - To change the state or to update a resource
- DELETE - To remove or delete a resource from the server

KARPAGG.COM