Agrawal Kanan

1716210105o

Q.1
>

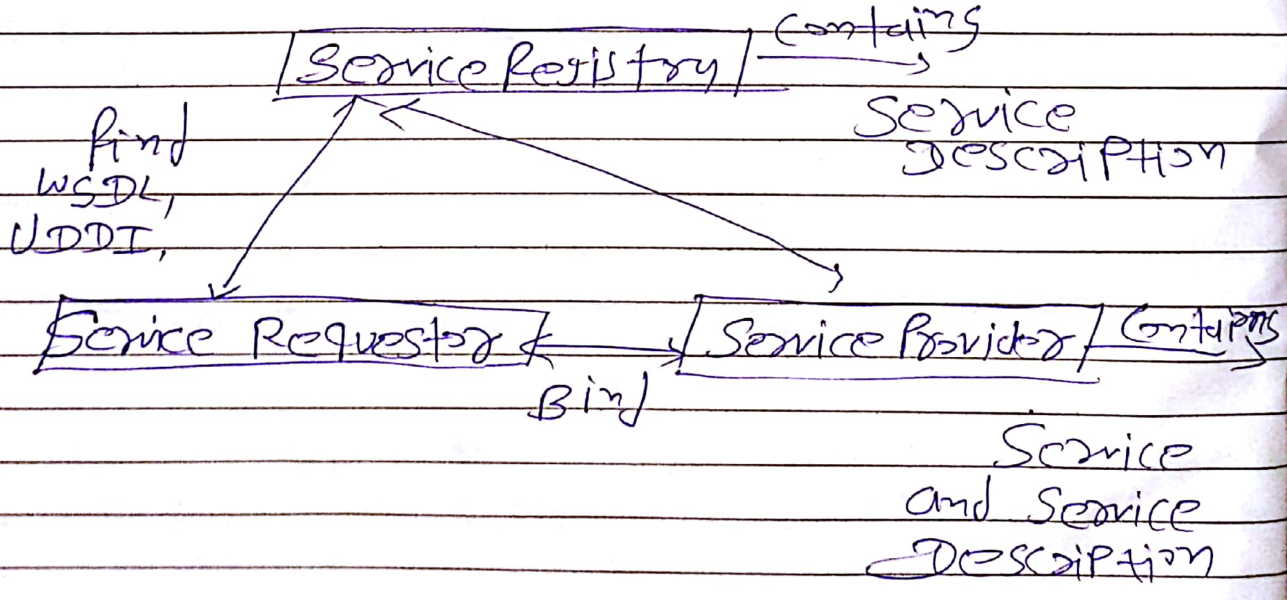
The web services architecture describes how to instanticate the elements and implement the operations in an interoperable manner.

-> The architecture of web service interacts among three roles:
-> service Provider
-> Service requestor

-> Service registry

-> interaction involves the three operations: publish, find and bind.

Architecture:-



Examples:-

->Service Registry:-
Provides support for publishing and locating services
->like telephone Yellow Pages

->Service Provider:-

->Provides e-business Services
->Publishes these Services through a registry

-> Service requestor:-

->finds required Services via the service ~~Br~~ Broker
-> Binds to Services Via Service Provider.

Agrawal Karan
1716210050

[illegible]

Q. 2
Compare SoA with client-Server architecture

Service oriented Architecture	client Server Architecture
-> Presentation layer is deta-ched from services themselves	Application logic, resides in client software client needs to control ~~[illegible]~~ User experience, back-end resour ces.
-> Complex Security model but Provides Flexibility	for simple scenario, im-plements a very easy Security model.
-> highly distributed processing services themselves may be distributed over many server	Client is responsible for most of the processing. 80/20 rule was used as a rule of thumb
-> Also has high maintenance costs and requires Powerful administration tools	Very high maintenance costs of maintenance : cli-ent server applications
-> object oriented	message oriented

Scanned by CamScanner

Asswall Kadam

1716210050

[illegible]

Q.3

= Service modeling

-> Candidate - The Primary goal of the Service-oriented analysis stule is to figure out what it is we need to letter design and build in subsequent Project phases. we are Producing abstract candidates that may or may not be realized as part of the eventual concrete design.

-> Service Candidates :- Service candidates are derived from original candidates. The we ~~Process~~ propose service operation candidates. finally, Service candidates and service operation candidates are the end-result of a Process called Service modeling

-> Steps:-

Step 1:- Decompose the business Process
-> Take the documented business Process and break it down into a series of granular Process steps.

Step 2:- Identify business service operation candidates
-> Some steps withing a business Process can be easily identified as not belonging to the Potential logic that should be encapsulated by a service candidate

Steps3:-
->Identify the Parts of the Processing logic that this layer would potentially abstract
-> potential types of logic suitable for layer include:
Business logic, exception logic

Step4:- Create business Service Candidates
-> Review the Processing Steps that remain and determine one or more logical contexts with which these steps can be grouped, Each context represents a Service candidate

Steps 5:- Refine and apply Principles of service-orientation
-> look at the Underlying logic of each Proposed Service operation ~~[illegible]~~ Candidate

-> steps 6:- Identify Candidate Service compositions

-> Steps 7:- Revise business Service operation grouping

-> Steps 8:- Analyze application Processing requirements
-> steps 9:- Identify application Service operation Candidates

-> steps 10:- Create application Service Candidates
-> Step 11:- Revise Candidate service Compositions

-> step 12:- Revise application Service operation grouping

Scanned by CamScanner

Agarwal Pradum
17162190050

Q.4

Schema File:-

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/
                    XML/Schema"
elementFormDefault="qualified">

<xs:element name="order">
  <xs:complexType> (complex)
  <xs:element name="firstname" type="xs:string"
                                          />

<xs:element name="lastname" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="mobile" ~~∅~~ type="xs:string"
                                          />

<xs:element name="Address" type="xs:string"/>
<xs:element name="card no" type="xs:string"/>
<xs:element name="cvv" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

XML File:-

```
~~xml ver~~
  <?xml version="1.0"?>
<order xmlns:xsi="http://www.w3.org/2001/XML
 Schema-instance" xsi:schemaLocation=~~"order~~
 "order.xsd">
<firstname> xyz </firstname>
<lastname> abc </lastname>
<email> abc@gmail.com </email>
```

Scanned by CamScanner

```
<mobile> 992345678 </mobile>
<address> Mansa </address>
<card no> 9245 6784 3323 </card no>
<CVV> 233 </CVV>

</order>
```