

## Unit : 1 : Introduction To Distributed Computing and SOA.

\* Write Define of following term :

### 1 Distributed Computing :

A Distributed Computing is a System whose components are located on different networked Computer, which communicate by passing the message to achieve a common goal.

### 2 The Web :

Design of Web is used to get information regarding any topic. The Web is Processing asynchronous client-server.

### 3 Web Service :

Web Service is a client-server application component for communication. It is a method of communication between two devices over the network.

\* Explain Different types of System Architectures.

=> This are the basic type of System Architectures.

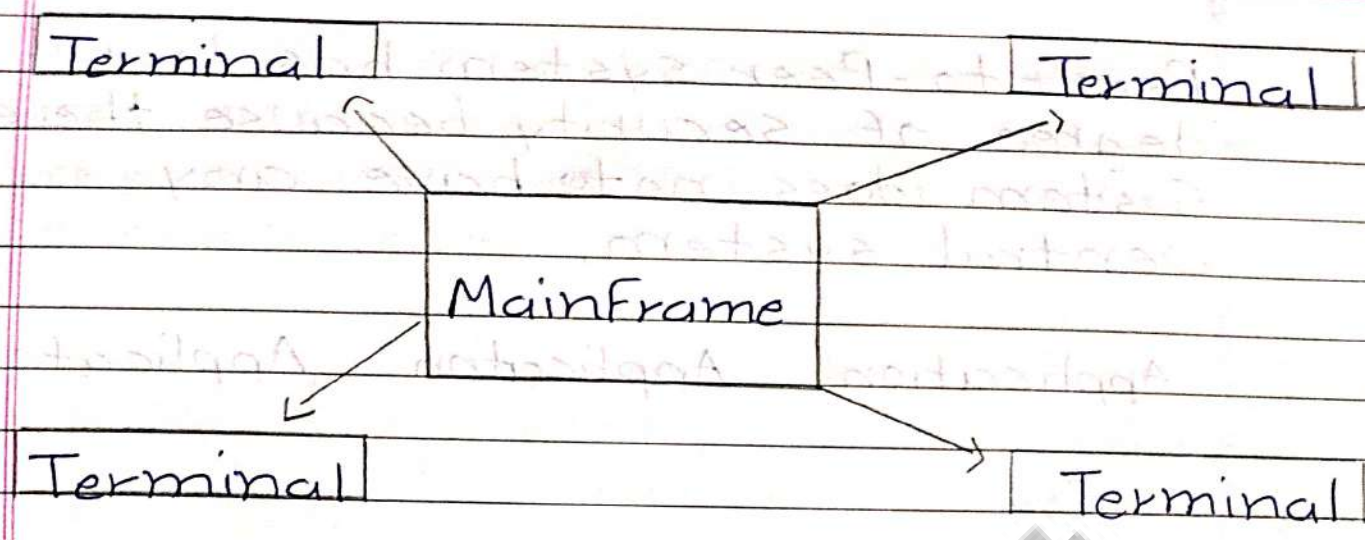
- a) Centralized Architecture
- b) Peer-to-Peer Architecture
- c) Client-Server Architecture
- d) Cooperative Architecture.

a Centralized Architecture :

In Centralized Architecture, every node of a system is connected with one centralized system.

Centralized System has high degree of security because there they have only one kind of access of the system.

The system performance of this system may be decreases when many users try to work simultaneously.



Here, Mainframe is a Central System, which given to access other Terminals in a system.

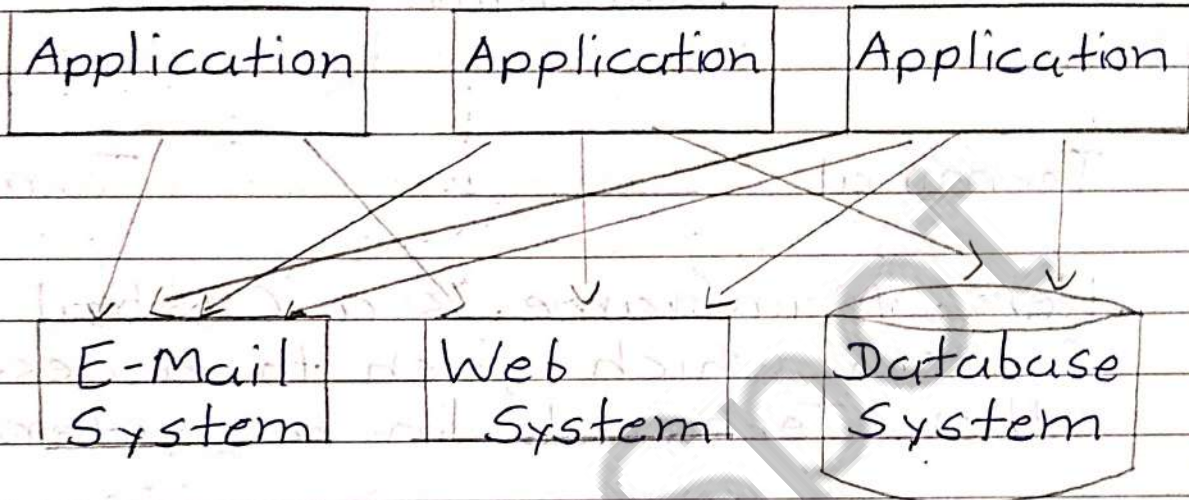
### b Peer-to-Peer Architectures:

In Peer-to-Peer Architecture, the computers connects with each other in a work group to share information.

In this system, each computer has ability to share the information in a computer network.

Peer-to-Peer System does not need for any dedicated server in a system.

Peer-to-Peer system has less degree of security because ~~there~~ System does not have any central system.



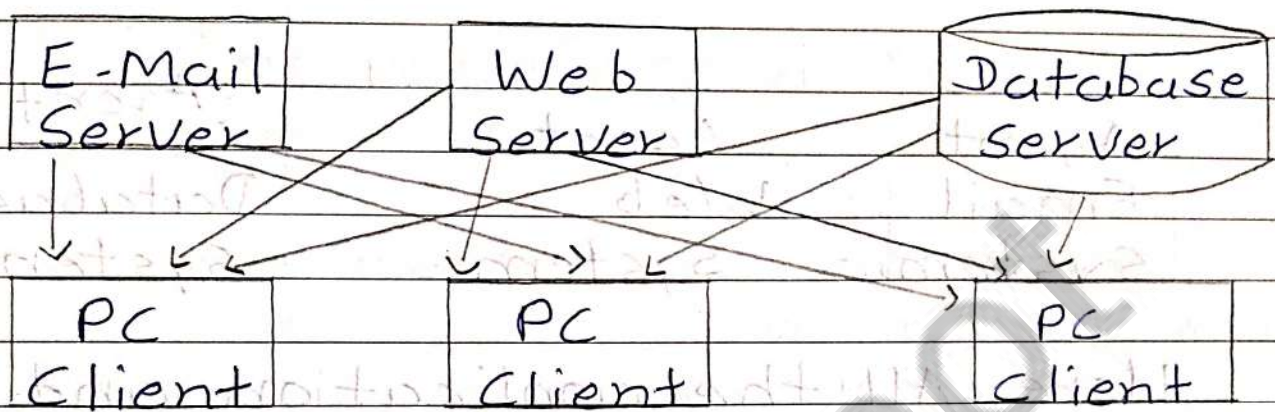
In this system, all the application and system can access all the network system.

### c. Client-Server Architecture :

In Client-Server Architecture, system is divided into the two part which is providers of a resource and service requesters.

Providers of a Resource is called Server and Service requesters is called Client.

When a Client sends a request for data access using internet, the server accepts the requested process and deliver the information

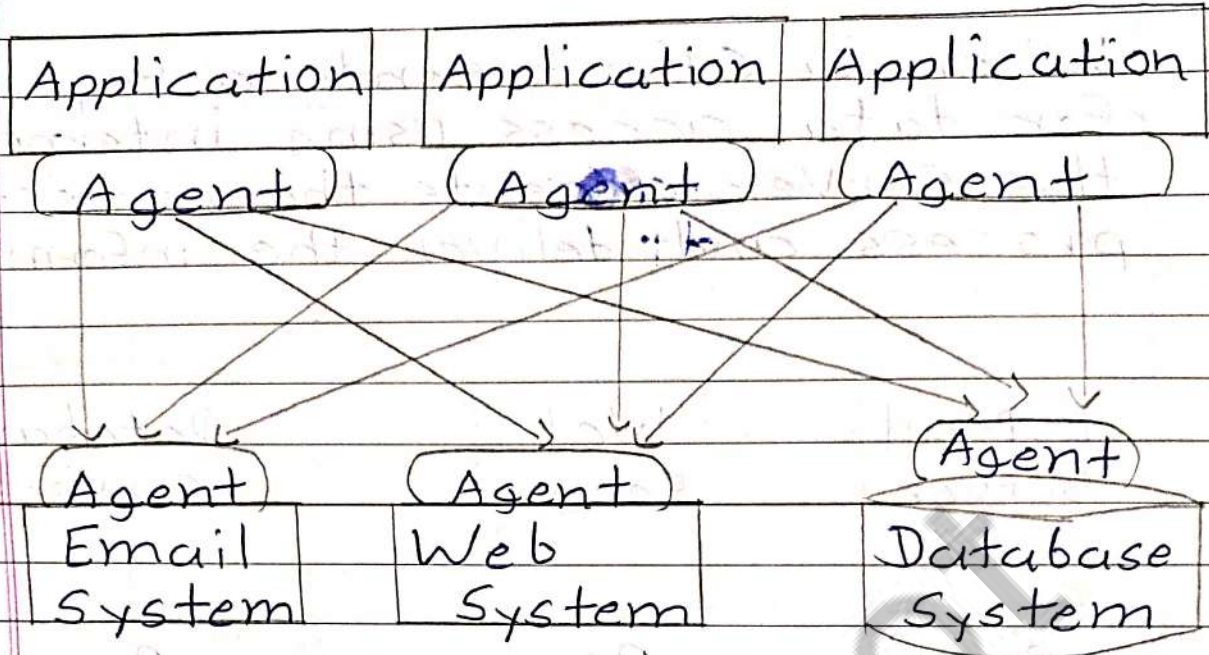


Here, All the PC Client can request to access e-mail, web and database server and All the server can deliver the information

#### d Co-operative Architecture:

In Co-operative Architecture, the computer connect with each other in workgroup with the Agent.

All the Computer and All the System has agent to access the system.



Here, All the application and system has Agent.

Using Agent, application and system can access the system.

\* Explain Web Service operations and roles.

=> Web Service Roles:

There are three roles in web service architecture.

- a) Service Registry
- b) Service Provider
- c) Service Consumer

a Service Registry :

Service Registry contains all the service information.  
Service Provider can register the service in the Service Registry.

Service Requestor can find service in the service registry.

b Service Provider :

Service Provider provides service for service requestor.

Service Provider can register the service in the service Registry.

Service Requestor have to request the Provider to provide any type of services.

c Service Requestor :

Service Requestor can consume the service which is provide by

Service provider.

Service Requester can find the service in the service registry.

=> Web Services Operation:

There are three type of Operation in web services.

- 1) Publish
- 2) Find
- 3) Bind

1. Publish:

In Publish operation, service description is publish in the Service Registry.

After the Publish the service in service registry, service requestor can find the service.

2 Find:

After the Publish operation is complete than find operation.



is perform to find the services.

Service Requestor, using find operation can find the services.

### 3 Bind:

After the find the service, service requestor can perform the bind operation.

In bind operation, service requestor invoke the services for use the services.

Page \_\_\_\_\_

\* What is XML? Write its Features and Advantages. Create DTD File for student information and create valid well-formed XML document to store student information with DTD File.

=> XML :

XML stands for extensible Markup Language much like HTML.

XML is designed to store and transport data and to be self-descriptive.

XML language does not have any predefined tags.

XML can be save using .xml extension.

XML simplifies Data Sharing, Data Transport and Data availability.

Syntax :

<root >

<child >

<subchild > . . . . </subchild >

<child >

<root >

All the User defined tags are case sensitive.

→ XML Features :

- 1 XML Language is Human and computer friendly format.
- 2 XML has tree structure for Data Handle like root, child etc.
- 3 XML can handle complex structure data for a long time.
- 4 XML has a User defined structure to create any XML document.

5 In XML, Data is described using markup Language.

6 XML uses a Document Type Definition or XML Schema for describe the data.

7 XML is Platform independent.

→ XML Advantages

1 XML Document is plain text and human readable.

2 XML Document has a tree structure to handle complex data.

3 XML Files are Operating System independent.

4 XML Document is allows content reuse.

5 XML is allows Users to defines their own tags.

6 XML is extendable.

-> XML File :

```
<?xml version = "1.0" ?>  
<!DOCTYPE Student-Info SYSTEM "  
C:/Users/Lenovo/Desktop">
```

```
<student-Info>
```

```
<Student>
```

```
<name> Khushi Gandhi </name>
```

```
<id> 21BETT30026 </id>
```

```
<age> 19 </age>
```

```
<address> Godhra </address>
```

```
</Student>
```

```
</Student-Info>
```

-> DTD File :

```
<?xml version = "1.0" ?>
```

```
<!ELEMENT Student-Info (Student*)>
```

```
<!ELEMENT Student-Info (name, id,  
age, address)>
```

```
<!ELEMENT name (#PCDATA)>
```

```
<!ELEMENT id (#PCDATA)>
```

<!ELEMENT age(#PCDATA)>

<!ELEMENT address(#PCDATA)>

\* What is XML Schema. Why we need XML Schema. Create XML Schema for student information and create a valid well-formed XML.

=> XML Schema:

XML Schema is used to define structure of an XML document.

XML Schema is also called XML Schema Definition.

XML Schema is used to validate the XML Document data.

XML Schema is defined whole XML document in the form of tree structure.

XML Schema alternative to DTD use.

XML Schema can be save using .xsd extension.

-> Why we need XML Schema:

The main purpose of an XML Schema is to define the legal building blocks of an XML Document.

XML Schema contains element and attributes that can appear in a document.

-> Student Information XML File:

```
<?xml version="1.0"?>
```

```
<Student Info
```

```
  xmlns:xsi="http://www.w3.org/2001
```

```
  XMLSchema-instance"
```

```
  xsi:noNamespaceSchemaLocation  
  = "C:\Users\Lenovo\Desktop">
```

```
<student>
```

```
<name>Khushi Gandhi</name>
```

```
<Id>26</Id>
```

```
<age>19</age>
```

```
</student>
```

```
</Student Info>
```

-> XML Schema:

```
<?xml version="1.0"?>
```

```
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema">
```

```
<xs:element="StudentInfo"  
  type="xs:string">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element="student"  
  type="xs:string">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element="name"  
  type="xs:string">
```

```
<xs:element="ID"
```

```
  type="xs:decimal">
```

```
<xs:element="age"
```

```
  type="xs:decimal">
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:schema>
```



\* What is XSL in XML. Using XSL display student information in tabular format.

=> XSL:

XSL stands for Extensible Stylesheet Language.

XSL is used to style the XML document like CSS styling.

XSL file can be saved using .xsl extension.

XSL provides an easy way to merge XML data into the presentation.

-> Student information XML File:

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl"  
  href="C:\Users\Lenovo\Desktop"?>  
<Student-Info>
```

```
  <Student>
```

```
    <name>Gandhi Khushi</name>  
    <Id>26</Id>  
    <age>19</age>
```

</Student>

<Student>

<name> Gandhi khush </name>

<Id> 26 </Id>

<age> 19 </age>

</Student>

</Student-Info>

-> Student Information XSL File:

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h1 align="center">Student  
Details </h1>

<table border="3" align="center">

<tr>

<th> Name </th>

<th> Id </th>

<th> Age </th>

</tr>

```
<xsl:for-each select = "Student-Inf  
/Student">  
<xsl:sort select = "age" />  
  
<tr>  
  
<td><xsl:value of select = "name">  
</td>  
<td><xsl:value of select = "age"></td>  
<td><xsl:value of select = "Id">  
</td>  
</tr>  
  
</xsl:for-each>  
</table>  
  
</body>  
</html>  
</xsl:template>  
</xsl:stylesheet>
```

\* What is SoA. Explain the Fundamental of SoA.

=> SoA:

SOA is stands for Service-Oriented Architecture which defines a way to make

software components.

SOA is define way, how services communicate with each other and provides Architectures of Services.

SOA is define how Services Provider and Services consumer communicate with each other using interface.

SOA is Provides all the Services specification in Services registry.

Using SOA, Service Provider can implements any Services specification.

Using SOA, Service consumer can request to consume the Services.

The SOA services are easily available to any service requester or consumer.

The SOA services is easier to test and deployed any where.

## Fundamental of SOA :

SOA allows users to combine a large number of facilities in a one services.

SOA Provides loosely couples of services.

SOA can reduced cost of application development and deployment.

SOA can Provides Antonomy in which whole services have control over the logic.

SOA Services can be easily available to anyone on request.

SOA applications are more reliable because it is easy to maintain large service.