EX.NO: 1 PROCEDURE TO DEVELOP A NEW WEB SERVICE FOR CALCULATOR

DATE:

AIM:

To write a procedure to develop a new Web Service for Calculator.

PROCEDURE:

Step 1: Create a Java Web Project

Open NetBeans IDE

Click on New Project and choose Java Web ->Web Application

Enter the Project Name: CalculationWS, using the default settings and then click on “Finish”.

Step 2: Create a Web Service

Now go to the Project Tree Structure on the left side of the window.

Right click on the project and select “New” and then choose “Web Service”

Specify web service name “CalWS” and package name “CalculationWS”. Click on “Finish”.

In Open CalWS.java file, replace the existing code with the following code:

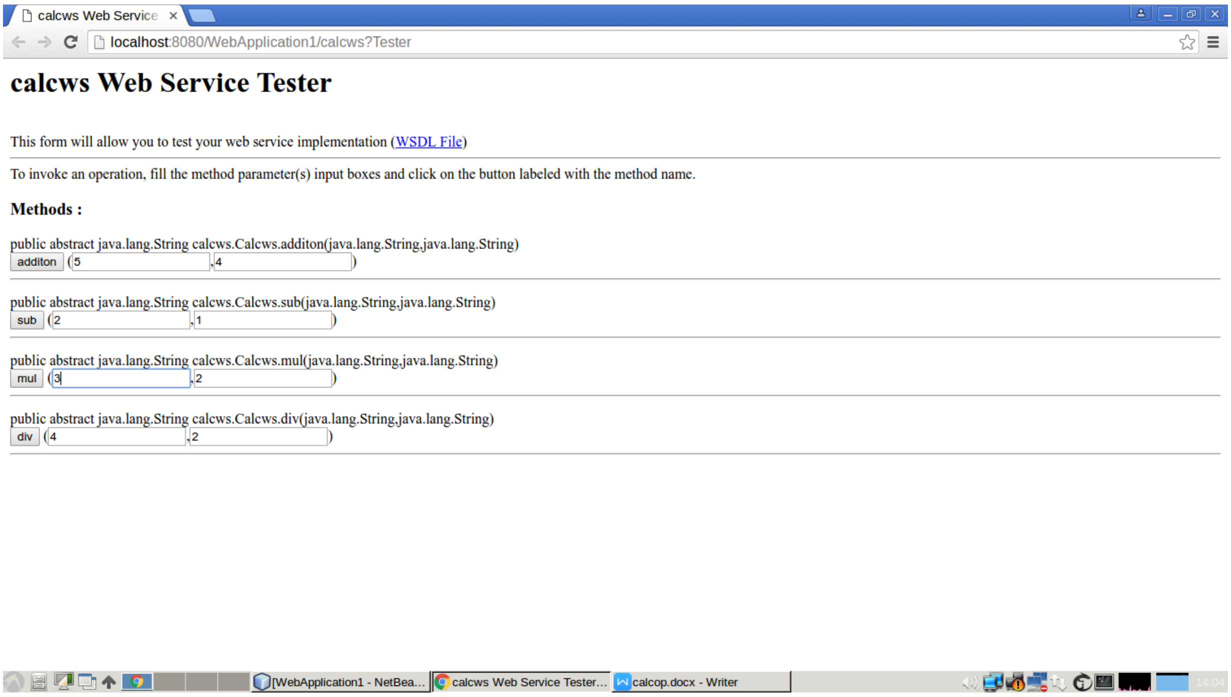
```
@WebMethod(operationName = "Addition")

public String Addition(@WebParam(name = "value1") String value1,@WebParam(name = "value2") String value2 ) {

float value=Float.valueOf(value1)+Float.valueOf(value2);
return (Float.toString(value));

}

@WebMethod(operationName = “Subtraction")

public String Addition(@WebParam(name = "value1") String value1,@WebParam(name = "value2")
String value2 ) {

float value=Float.valueOf(value1)-Float.valueOf(value2);

return (Float.toString(value));

}

@WebMethod(operationName = "Multiplication")

public String Addition(@WebParam(name = "value1") String value1,@WebParam(name = "value2")
String value2 ) {

float value=Float.valueOf(value1)*Float.valueOf(value2);

return (Float.toString(value));

}

@WebMethod(operationName = "Division")

public String Addition(@WebParam(name = "value1") String value1,@WebParam(name = "value2")
String value2 ) {

float value=Float.valueOf(value1)/Float.valueOf(value2);

return (Float.toString(value));

}
```

Step 3: Deploy and Test Web Service

Right click on the project and select “Deploy”

To test the web service, right click on the service and select “Test Web Service”.

OUTPUT:


calcws Web Service
localhost:8080/WebApplication1/calcws?Tester
[WebApplication1 - NetBea...
calcws Web Service Tester...
calcop.docx - Writer

calcws Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

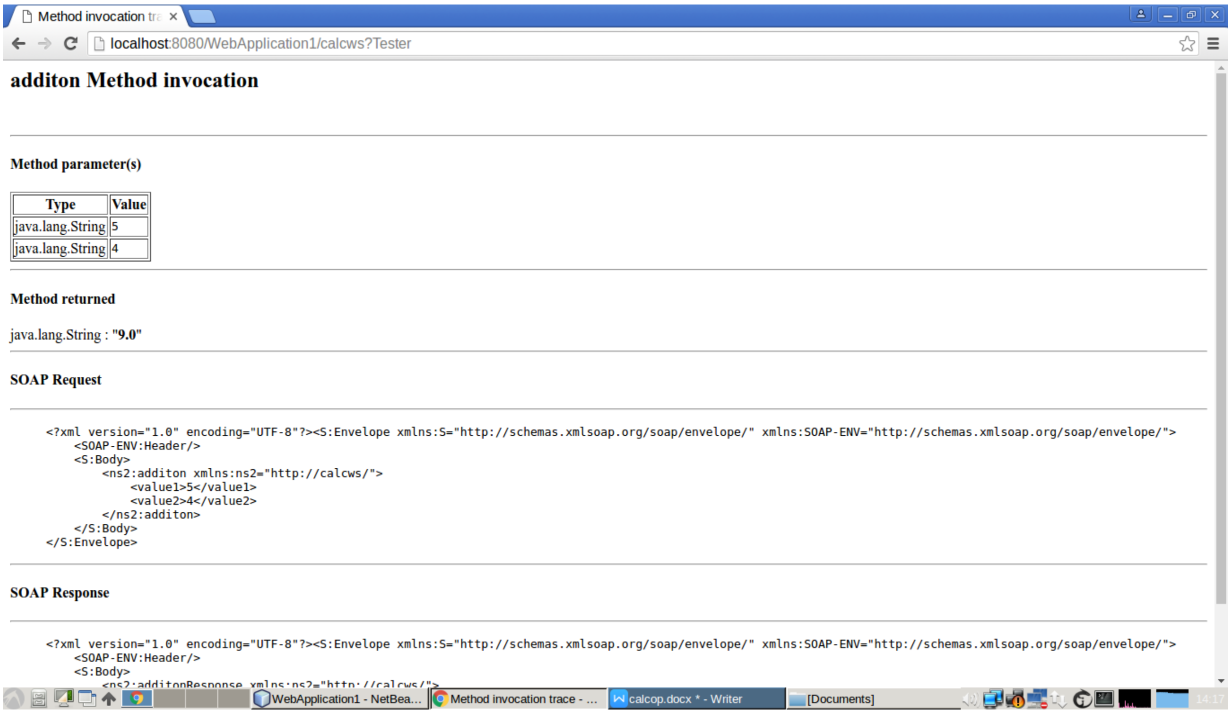
Methods :

public abstract java.lang.String calcws.Calcws.additon(java.lang.String,java.lang.String)
additon (5 ,4)

public abstract java.lang.String calcws.Calcws.sub(java.lang.String,java.lang.String)
sub (2 ,1)

public abstract java.lang.String calcws.Calcws.mul(java.lang.String,java.lang.String)
mul (3 ,2)

public abstract java.lang.String calcws.Calcws.div(java.lang.String,java.lang.String)
div (4 ,2)


Method invocation tr
localhost:8080/WebApplication1/calcws?Tester
WebApplication1 - NetBea...
Method invocation trace - ...
calcop.docx * - Writer
[Documents]

additon Method invocation

Method parameter(s)

Type	Value
java.lang.String	5
java.lang.String	4

Method returned

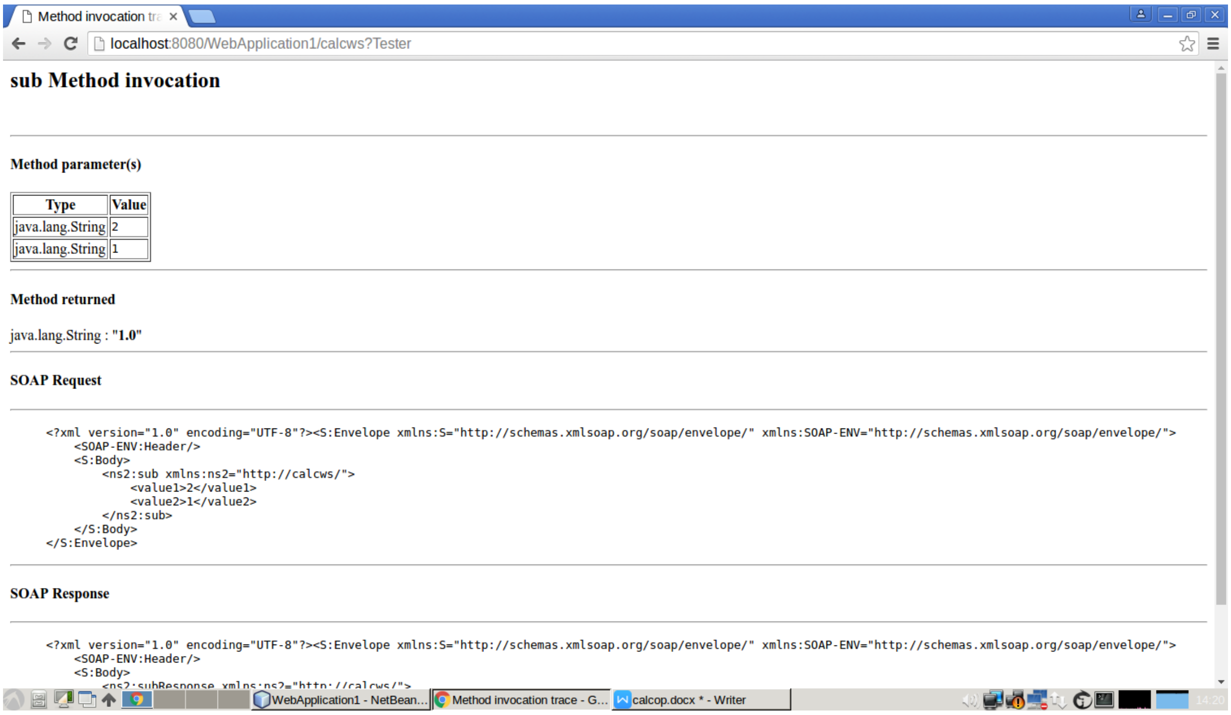
java.lang.String : **"9.0"**

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:additon xmlns:ns2="http://calcws/">
            <value1>5</value1>
            <value2>4</value2>
        </ns2:additon>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:additonResponse xmlns:ns2="http://calcws/">
```



Method invocation tr
localhost:8080/WebApplication1/calcws?Tester

sub Method invocation

Method parameter(s)

Type	Value
java.lang.String	2
java.lang.String	1

Method returned

java.lang.String : **"1.0"**

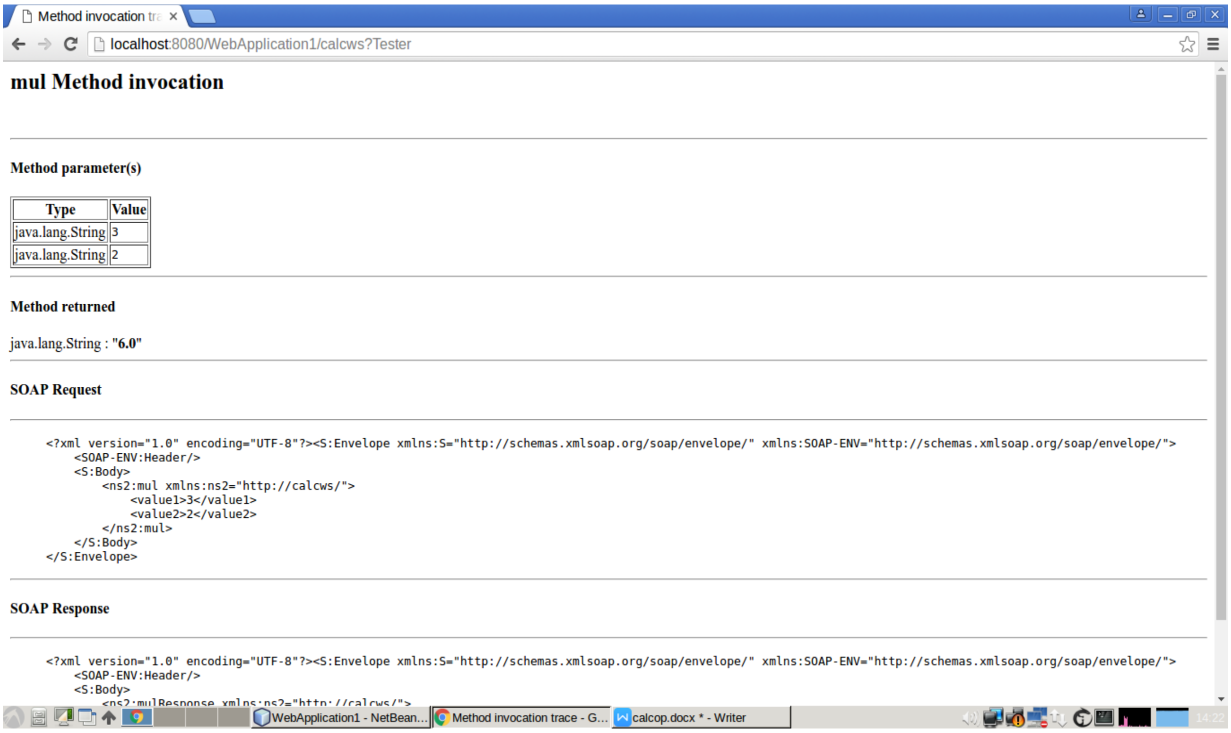
SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:sub xmlns:ns2="http://calcws/">
            <value1>2</value1>
            <value2>1</value2>
        </ns2:sub>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:subResponse xmlns:ns2="http://calcws/">
```

WebApplication1 - NetBean...
Method invocation trace - G...
calcop.docx * - Writer
14:20



Method invocation tr
localhost:8080/WebApplication1/calcws?Tester

mul Method invocation

Method parameter(s)

Type	Value
java.lang.String	3
java.lang.String	2

Method returned

java.lang.String : **"6.0"**

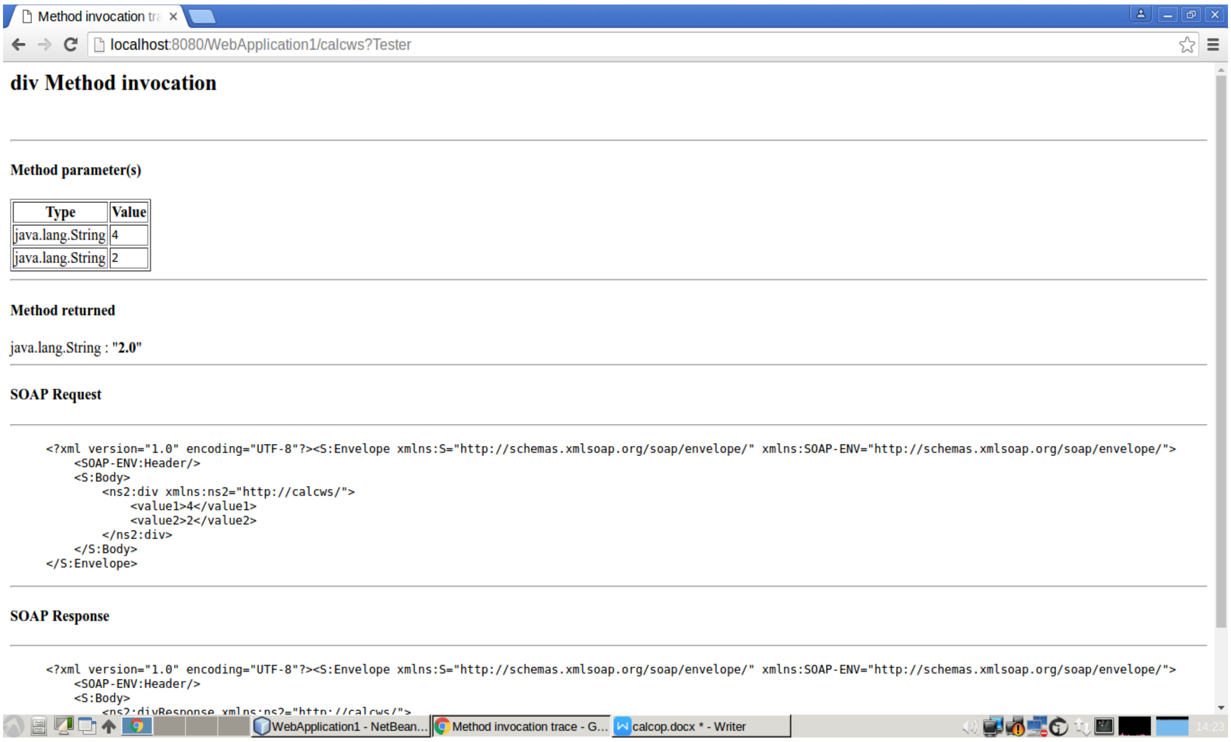
SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:mul xmlns:ns2="http://calcws/">
            <value1>3</value1>
            <value2>2</value2>
        </ns2:mul>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:mulResponse xmlns:ns2="http://calcws/">
```

WebApplication1 - NetBean...
Method invocation trace - G...
calcop.docx * - Writer
14:22



Method invocation tr
localhost:8080/WebApplication1/calcws?Tester
WebApplication1 - NetBean...
Method invocation trace - G...
calcop.docx * - Writer
14:23

div Method invocation

Method parameter(s)

Type	Value
java.lang.String	4
java.lang.String	2

Method returned

java.lang.String : **"2.0"**

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:div xmlns:ns2="http://calcws/">
            <value1>4</value1>
            <value2>2</value2>
        </ns2:div>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:divResponse xmlns:ns2="http://calcws/">
```

RESULT:

Thus the procedure to develop a new Web Service for Calculator is executed and output is verified.

EX.NO: 2 **DEVELOP NEW OGSA-COMPLIANT WEB SERVICE**
DATE:

AIM:

To study the development of a new OGSA-Compliant Web service in Grid Service using .NET language.

PROCEDURE:

The implementation of Grid Services with .NET technologies, management of the dynamic nature of the information describing the re-sources (e.g. CPU usage, Memory usage, etc.), extension of the MS directory service functionalities (e.g. Active Directory) in order to implement the OGSA Index Service functionalities.

To perform the collection and provisioning of the performance data to an index service, we leverageon Windows Management Instrumentation (WMI) architecture. WMI is based on the Common Information Model (CIM) schema, which is an industry standard specification.

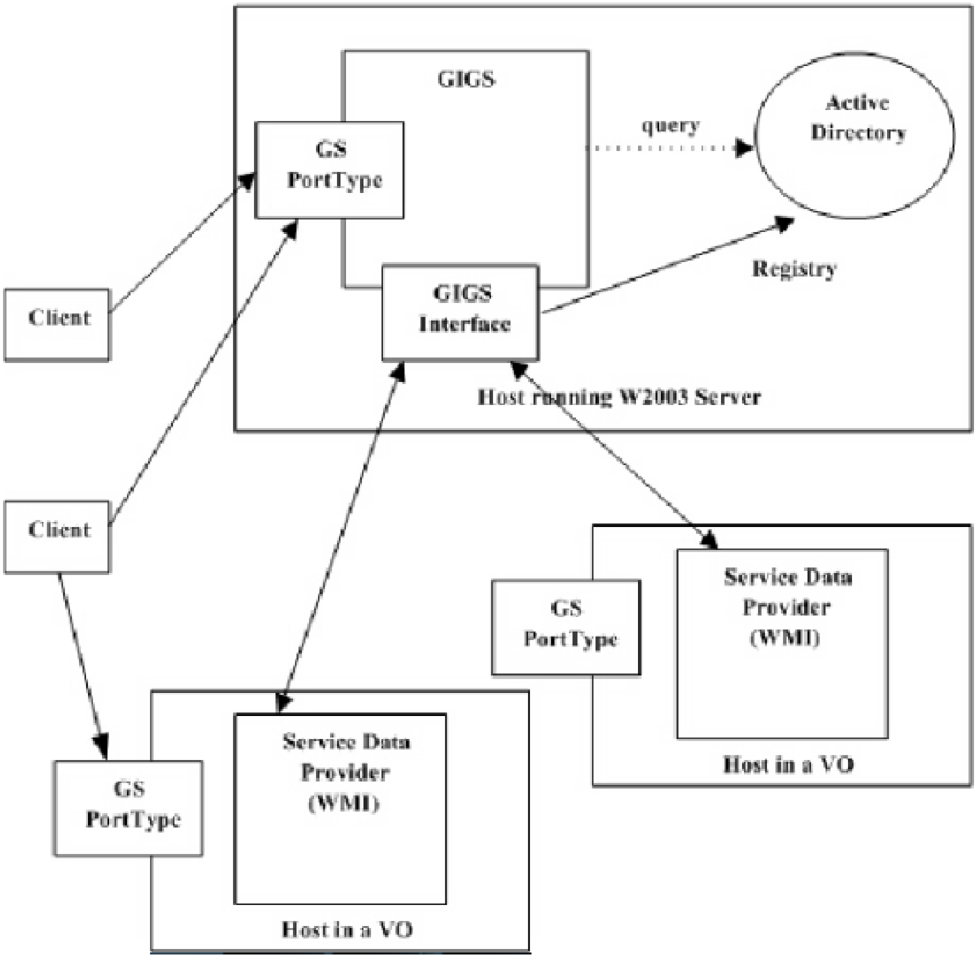
To implement an OGSA-compliant Index Service, we exploit some Active Directory (AD) features . AD is a directory service designed for distributed networking environments providing secure, structured, hierarchical storage of information about interesting objects, such as users, computers, services, inside an enterprise network.

Our goal is to implement a Grid Service that, taking the role of a consumer, queries at regular intervals the Service Data Providers of a VO to obtain resources information, collect and aggregate these information, and allows to perform searches, among the resources of an organization, matching a specified criteria (e.g. to search for a machine with a specified number of CPUs). In our environment this Grid Service is called Global Information Grid Service (GIGS).

The hosts that run the GIGS have to be Domain Controllers (DC). A DC is a server computer, running on Microsoft WindowsNT, Windows2000, or Windows Server2003 family operating systems, that manages security for a domain.

GIGS has to implement an interface in order to obtain, using a publish/subscribe method, a set of data from Service Data Providers describing an active directory object. Such data are then recorded in the AD by using Active Directory Service Interface (ADSI), a COM based interface to perform common tasks, such as adding new objects.

After having stored those data in AD, the GIGS should be able to query AD for retrieving such data. This is obtained exploiting the Directory Services.



EXAMPLE OF GIGS

RESULT:

Thus the development of a new OGSA-Compliant Web service in Grid Service using .NET language has been studied.

EX.NO: 3 **DEVELOP GRID SERVICE USING APACHE AXIS**
DATE:

AIM:

To write a program using Apache Axis to develop a Grid service.

PROCEDURE:

Step 1: Create the interface using WSDL. Specify the portTypes, messages and data encoding.
Step 2: Generate Stubs.
Step 3: Add functionality.
Step 4: Compile and Build the code using Globus libraries.
Step 5: Create a GAR (Grid Archive).
Step 6: Deploy it.

Listing 1

```
package org.globus.ogsa.guide.impl.guide;
public interface Counter {
    public int add(int value);
    public int subtract(int value);
    public int getValue();
}
```

Listing 2

```
<types>
<xsd:element name="add">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
  </xsd:element>
</types>
<message name="AddInputMessage">
  <part name="parameters" element="tns:add"/>
</message>
<gwsdl:portType name="CounterPortType" extends="ogsi:GridService">
  <operation name="add">
    <input message="tns:AddInputMessage"/>
    <output message="tns:AddOutputMessage"/>
    <fault name="Fault" message="ogsi:FaultMessage"/>
  </operation>
</gwsdl:portType>
```

Listing 3

```
<ant antfile="${build.services}" target="generateWSDL">
  <property name="interface.package" value="org.globus.ogsa.guide.impl"/>
  <property name="interface.name" value="Counter"/>
  <property name="generated.dir" value="guide"/>
</ant>
```

Listing 4

```
<ant antfile="${build.services}" target="generateStubs">
  <property name="schema.file.dir" value="guide/Counter"/>
  <property name="schema.file" value="counter_service.wsdl"/>
</ant>
```

Listing 5

```
public class CounterImpl extends GridServiceImpl implements CounterPortType {
    private int val = 0;
    public CounterImpl() {
        super("Guide Counter");
    }
    public int add(int val) throws RemoteException {
        this.val = this.val + val;
        return this.val;
    }
    public int subtract(int val) throws RemoteException {
        this.val = this.val - val;
        return this.val;
    }
    public int getValue() throws RemoteException {
        return this.val;
    }
}
```

Listing 6

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment name="defaultServerConfig"
       xmlns="http://xml.apache.org/axis/wsdd/"
       xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
  <service name="guide/counter/CounterProviderFactoryService"
               provider="Handler" style="wrapped">
  <parameter name="name" value="Guide Counter Provider Factory"/>
  <parameter name="instance-name" value="Guide Counter Proivider Counter"/>
  <parameter name="instance-schemaPath"
               value="schema/guide/Counter/counter_service.wsdl"/>
  <parameter name="instance-className"
               value="org.globus.ogsa.guide.Counter.wsdl.CounterPortType"/>
  <parameter name="instance-baseClassName"
               value="org.globus.ogsa.impl.ogsi.GridServiceImpl"/>
  <parameter name="instance-operationProviders"
               value="org.globus.ogsa.guide.impl.CounterProvider"/>
  <parameter name="persistent" value="true"/>
  <parameter name="schemaPath"
               value="schema/ogsi/ogsi_notification_factory_service.wsdl"/>
  <parameter name="baseClassName"
```

```
               value="org.globus.ogsa.impl.ogsi.PersistentGridServiceImpl"/>
  <parameter name="handlerClass"
               value="org.globus.ogsa.handlers.RPCURIProvider"/>
  <parameter name="className"
               value="org.gridforum.ogsi.NotificationFactory"/>
  <parameter name="allowedMethods" value="*"/>
  <parameter name="factoryCallback"
              value="org.globus.ogsa.impl.ogsi.DynamicFactoryCallbackImpl"/>
  <parameter name="operationProviders"
              value="org.globus.ogsa.impl.ogsi.FactoryProvider org.
              globus.ogsa.impl.ogsi.NotificationSourceProvider"/>
 </service
</deployment>
```

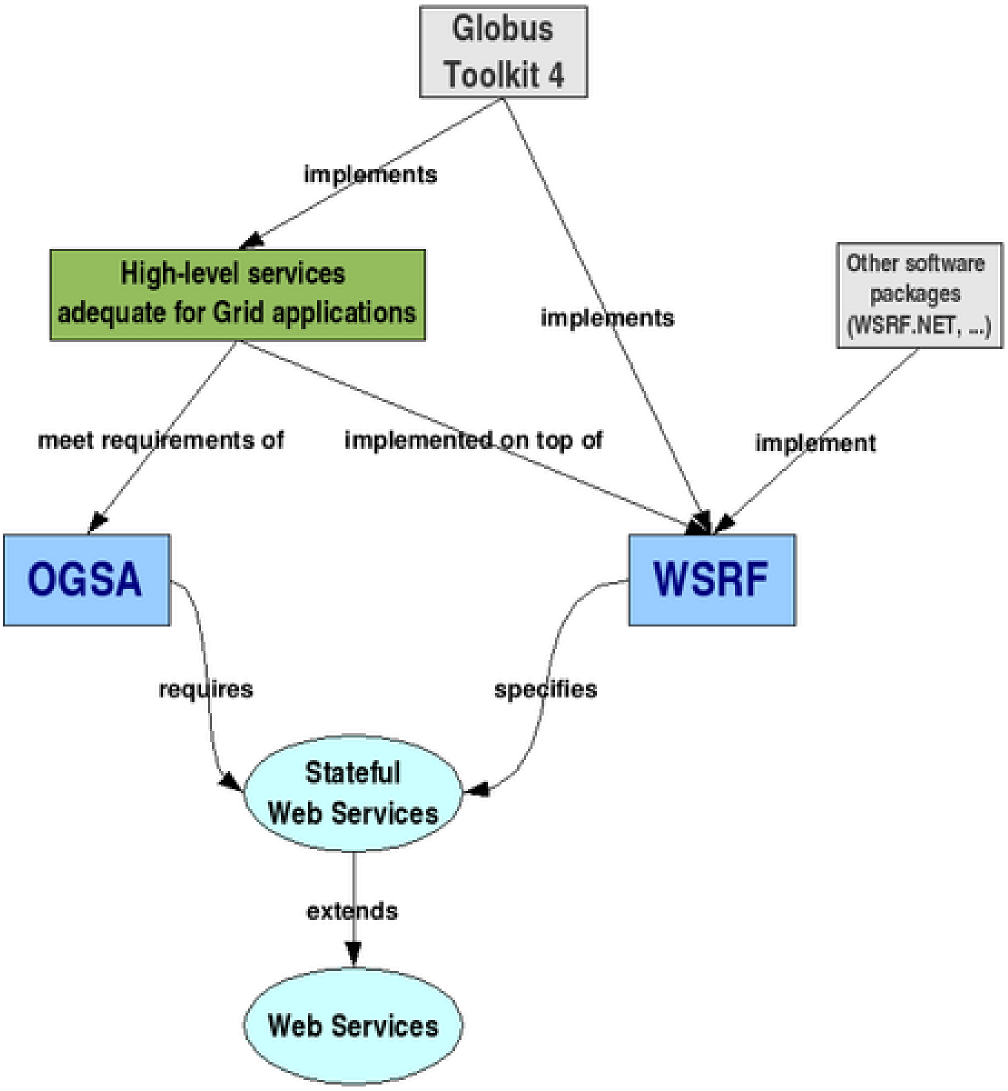
Listing 7

```
<ant antfile="${build.packages}" target="makeGar">
  <property name="gar.name" value="${build.lib}/guide.gar"/>
  <property name="garlib.dir" value="${build.lib}"/>
  <property name="garserverdeployment.file" value="guide-config.wsdd"/>
  <property name="garschema.origin" value="${build.schema}/guide"/>
  <property name="garschema.path" value="guide"/>
</ant>
```

Listing 8

```
OGSIServiceGridLocator gridLocator = new OGSIServiceGridLocator();
Factory factory = gridLocator.getFactoryPort(handle);
GridServiceFactory gridFactory = new GridServiceFactory(factory);
LocatorType locator = gridFactory.createService();
CounterServiceGridLocator counterLocator = new CounterServiceGridLocator();
CounterPortType counter = counterLocator.getCounterPort(locator);
int val = counter.add(2);
```

SAMPLE OUTPUT:



RESULT:

Thus the program using Apache Axis to develop a Grid service has been demonstrated successfully.