This section illustrates implementation of the new-implemented distributed programming solution. For that experiments there will be demonstration of 2 consumers (one developed with C# and another with Java) and 2 providers(one developed with C# and another with Java).

Properties of the consumers can be seen in Table1.

|  |  |  |
| --- | --- | --- |
| Mobile 1 (Windows Mobile) | Consumer | C# |
| Mobile 2 (Android Mobile) | Consumer | Java |

Properties of providers can be found in Table 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cloud Server 1 | Provider | .Net | http://csharpserverthesis.azurewebsites.net | IIS 7.0 Server |
| Cloud Server 1 | Provider | Java | http://javatomcatthesis.azurewebsites.net | Tomcat Server |

Windows Mobile application (Listing x) will send 2 different query messages (Query1 and Query2) to Cloud1 and Cloud2 providers then if the receiver conforms to what the sender expects then there is compliance, the message received is partially assigned to that argument and the operation invoked.

Android Mobile application (Listing x) will send a query messages (Query3) to Cloud1 and Cloud2 providers as seen in Table XX

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Cloud Server 1 | Cloud Server 2 |
| Mobile 1 | Query 1 | No | Yes |
| Query 2 | Yes | Yes |
| Mobile 2 | Query 3 | No | No |
|  |  |  |  |

Providers have operations with an input parameter of each of its operations. Matching will be done with message from consumer and input parameter of operations. Input parameters of operations in each server can be seen in Table XX

|  |  |
| --- | --- |
| Cloud Server 1 | Cloud Server 2 |
| Weather1(in Listing) | Weather 4 |
| Weather 2 | Weather 5 |
| Weather 3 | Weather6 |

Lets start with Windows mobile application (Mobile1); first thing to do is creating a message with query1 (in Listing) and sending it to .Net cloud provider(Cloud1).

Weather1 object in Cloud1 provider will not be mapped to query1 because …

Weather2 object in Cloud1 provider will not be mapped to query1 because…

Weather3 object in Cloud1 provider will not be mapped to query1 because…

As explained above, there is no compliance for query1 in the Cloud1 provider.

Next step is, sending query1 message to Java server in Cloud2.

Weather4 object in Cloud2 provider will be mapped to query1 and complies. Because …

Weather5 object in Cloud2 provider will not be mapped to query1, because…

Weather6 object in Cloud2 provider will not be mapped to query1, because…

Now, what about when sending query2(in Listing) with Windows mobile application to Cloud1 and Cloud2 providers.

Weather1 object in Cloud1 provider will not be mapped to query2 because …

Weather4 object in Cloud2 provider will be mapped to query1 and complies. Because …

Weather3 object in Cloud1 provider will not be mapped to query2 because…

And again, sending query2 message to Java server in Cloud2 to see if there is any compliance.

As explained above, sending query1 message with Windows phone application will be mapped in Java weather provider and return result message after compliance.

The consumer does not need to be written in C#. it can be written in Java or other language. Lets show implementation of Android phone application (Mobile2). This application has query3 and it will send that query message to Cloud1 and Cloud2 providers.

The results that are explained above about partial assignment of Mobile1 and Mobile2 applications can be seen in Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Weather1 | Weather2 | Weather3 | Weather4 | Weather5 | Weather6 |
| Query 1 |  |  |  | ++ |  |  |
| Query 2 |  | ++ |  |  |  | ++ |
| Query 3 |  |  |  |  |  |  |