Recommendation Report

For

Technology Engineers

By: Ravikumar Chimmalgi

December, 2012

Table of Contents

Problem Statement

The IT Company Technology Engineers has planned to venture into the mobile application market. Their requirement is to be able to reuse their existing .NET 3.5 code to create applications for all major mobile platforms, thus reducing application deployment time, manpower and cost.

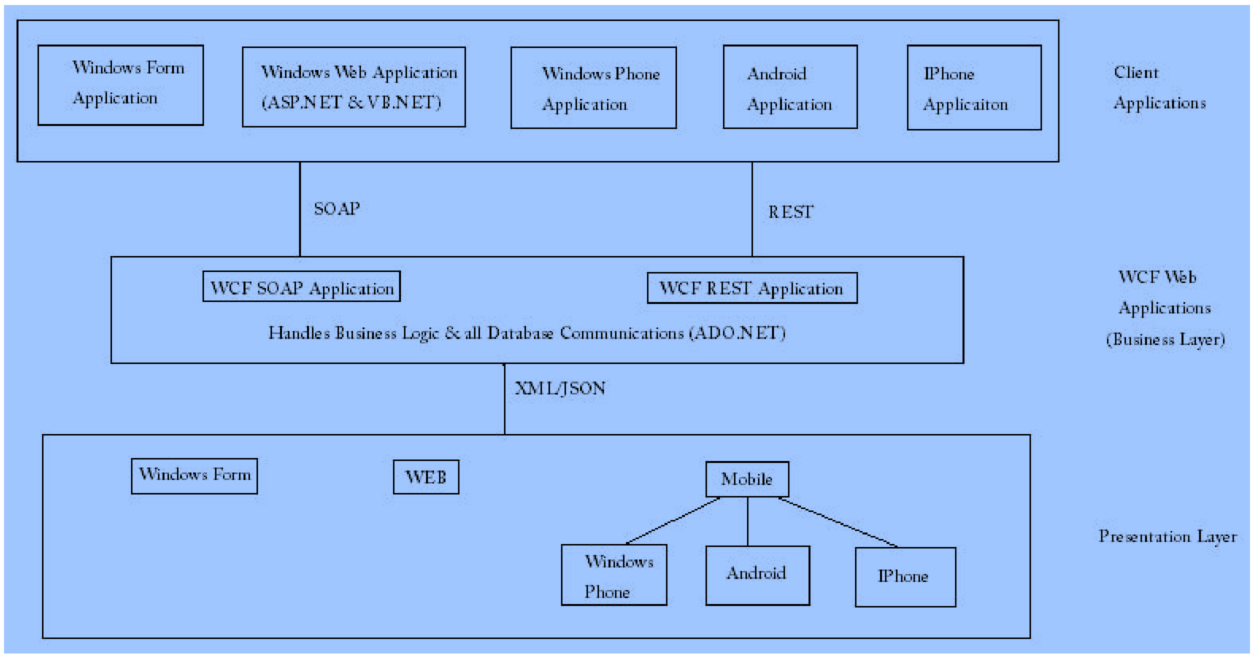
Objectives:

- Research existing mobile development technologies
- Research compatibility of .NET with all major mobile platforms
- Create and test mobile applications based on WCF SOAP
- Create and test mobile applications based on WCF REST
- Recommend a strategy on cross platform mobile development

PART I: Concept

1. Overview

The purpose of this report is to recommend on a strategy of mobile development using Windows Communication Foundation (WCF) to reuse existing .NET code.



Even though there are tools in the market for cross platform mobile development, like as Mono for Android and IPhone, they are expensive and have not yet reached the full compatibility level. These tools don't give you the full freedom as when compared to developing on native platforms

The proposed strategy is implementing a service oriented architecture based on WCF applications as the services and the mobile applications and .NET applications are the clients. The architecture can be based on SOAP/REST where the clients communicate to the WCF applications to send data and get back results.

The client applications act as the controller and as well as the view of the MVC model. The results obtained from the WCF service are displayed on the client devices, hence the developer has the benefits of being in a native environment. The WCF web applications act as the model and handle all the business logic of the system.

2. Windows Communication Foundation (WCF)

WCF is a set of APIs in the .NET framework which allows the creation of service oriented web applications.

WCF provides a single unified solution, rather than requiring different technologies for different communication styles like ASMX, .NET Remoting, System.Net etc.

WCF supports interoperability with WCF applications running on the same Windows machine or WCF running on a different Windows machines or standard Web services built on platforms such as java running on Windows or other operating systems. WCF lets us create Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) based web-services application. WCF creates a Web Services Description Language (WSDL) for the services which describe the services themselves. A WSDL file provides a machine-readable description in an XML format, of how the service can be called, what parameters it expects, and what data structures it returns.

3. WCF SOAP & REST

SOAP is a protocol specification for sending/receiving structured SOAP messages from point A to point B in the implementation of web services. It relies on XML for the message format. A SOAP service exposes the named operation of the service for use by the service clients.

A REST service exposes objects/nouns and instead of exposing a set of operations to clients as SOAP does, it allows its client to access everything through well defined universal HTTP methods/verbs such as GET, POST, PUT, DELETE etc. In RESTful services, instead of specifying the data for the RPC operations via parameters, as done with SOAP, everything is assigned a URL. To deal with the number of URLSs created, WCF provides URI templates which make it easier for developers to express URI patterns and work with URIs that match those patterns.

4. SOAP vs. REST

For SOAP

- For a traditional OOP programmer, SOAP is easy to understand. REST on the other hand has a learning curve.
- Along with SSL, SOAP supports WS-Security which adds some enterprise security features like identity through intermediaries, data integrity and data privacy. REST doesn't provide such added security.
- SOAP supports WS-AtomicTransaction for ACID (Atomicity Consistency Isolation Durability) transactions over a service. REST is not ACID compliant.
- SOAP supports WS-ReliableMessaging for messaging system which provides successful/retry logic built in and provides end-to-end reliability even through SOAP intermediaries. Rest doesn't have this support and expects clients to handle communication failures by retrying.

- SOAP services provide WSDL (Web Service Definition Language) makes generating a proxy for a SOAP-based service easier than writing the code to call a RESTful service.
- SOAP can be used over multiple transport protocols like TCP, HTTP, SMTP, and MSMQ. Rest is based on HTTP.

For REST

- SOAP permits only XML, where as REST supports XML, JSON and other formats. It's easier and faster to parse data using JSON.
- As REST relies on semantics of HTTP, GET requests (requests for data) can be cached. On the other hand, SOAP even with using HTTP does not allow this as it uses HTTP POST verb.
- Compared to REST, SOAP requests have additional headers and use more bandwidth.
- In SOAP, you have to deal with myriad WS-* specifications and figure out which to use (or not to use) and how that affects interoperability. In REST only thing you need is HTTP, and most devices today have internet connectivity. So, it is comparatively easier to implement REST based service oriented architecture.

SOAP and REST are both important and useful depending on the task you need to implement in the service. For example, in banking and other confidential information based services SOAP would be a better fit. For other services, as per the current trends, more and more developers and companies are shifting towards REST.

5. Graphics for this architecture

In this architecture, the graphics is expected to be handled in each of the individual platforms individually.

Currently there are some third party tools in the market which allow for handling graphics in cross-platform mobile development. One such promising tool I found was the PhoneGap framework.

5.1 PhoneGap

PhoneGap is an open source solution for building cross-platform mobile apps with standards-based Web technologies like HTML, JavaScript, CSS.

The benfit of PhoneGap is that it doesn't require higher programming skills like JAVA or Objective-C and that there is a single code base for all mobile platforms.

The downside of PhoneGap is that it doesn't allow the use of native pre-built controls of the individual platforms. To get the native look and feel, you can use try using Sencha Touch, JQ Touch, or similar tools with pre-built UI elements, but it will be more intensive and require a longer development time.

To use PhoneGap we would need to first create the web service, and then create the applications for all the platforms using PhoneGap. Then we can call the service in the mobile applications.

PART II: Implementing SOAP based SOA

6. SOAP based WCF Service Application

A WCF service class is like a normal class but with the addition of contracts. It has at least one ServiceContract which defines the operations that are exposed by the service. The service class can either be written by explicitly using an interface type or, by marking the class directly with ServiceContract and the class’s service contract is implicitly defined to consist of all methods in that class that are marked with OperationContract.

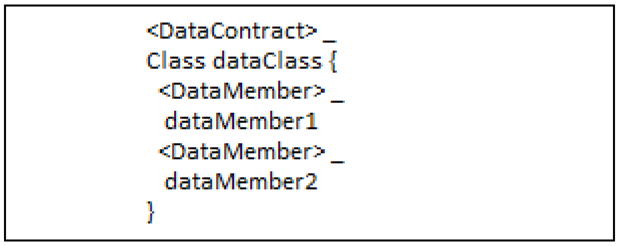
Using interfaces is a better option as it provides flexibility; a class can implement more than one interface, which means that it can also implement more than one service contract.

```
<ServiceContract> _
Interface serviceInterface {
 <OperationContratct>_
   Method1()
<OperationContratct>_
  Method2()
}
```

```
<ServiceBehaviour> _
Class Service1 Implements serviceInterface {
 Method1() Implements serviceInterface.Method1
    { Definition}
Method2() Implements serviceInterface.Method2
    { Definition}
}
```

6.1 DataContract

To use objects of complex types as parameters in Operation Contract methods, an explicit Data Contract is required. Data contracts are a mechanism for controlling how data is serialized.



```
<DataContract> _
Class dataClass {
 <DataMember> _
  dataMember1
 <DataMember> _
  dataMember2
}
```

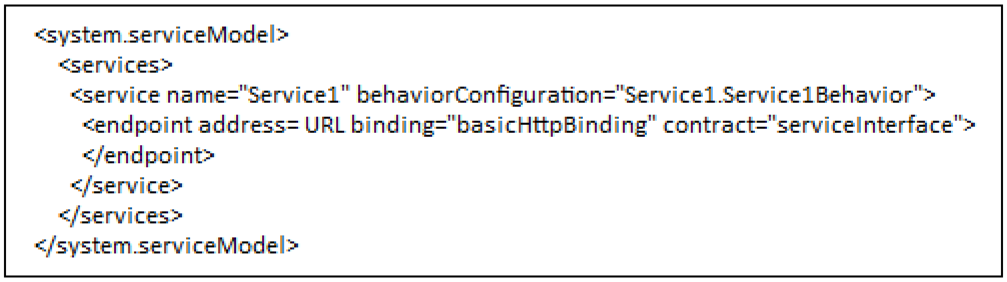
6.2 Endpoints

A WCF service exposes one or more endpoints through which all the communication with a WCF service happens. An endpoint definition consists of 3 components:

-An address indicating where this endpoint can be found. Addresses are URLs that identify a machine and a particular endpoint on that machine.

-A binding determining how this endpoint can be accessed. The most important ones for this project are BasicHttpBinding/WsHttpBinding for SOAP services and WebHttpBinding for REST services.

-The Service Contract



```
<system.serviceModel>
  <services>
   <service name="Service1" behaviorConfiguration="Service1.Service1Behavior">
     <endpoint address= URL binding="basicHttpBinding" contract="serviceInterface">
     </endpoint>
   </service>
  </services>
</system.serviceModel>
```

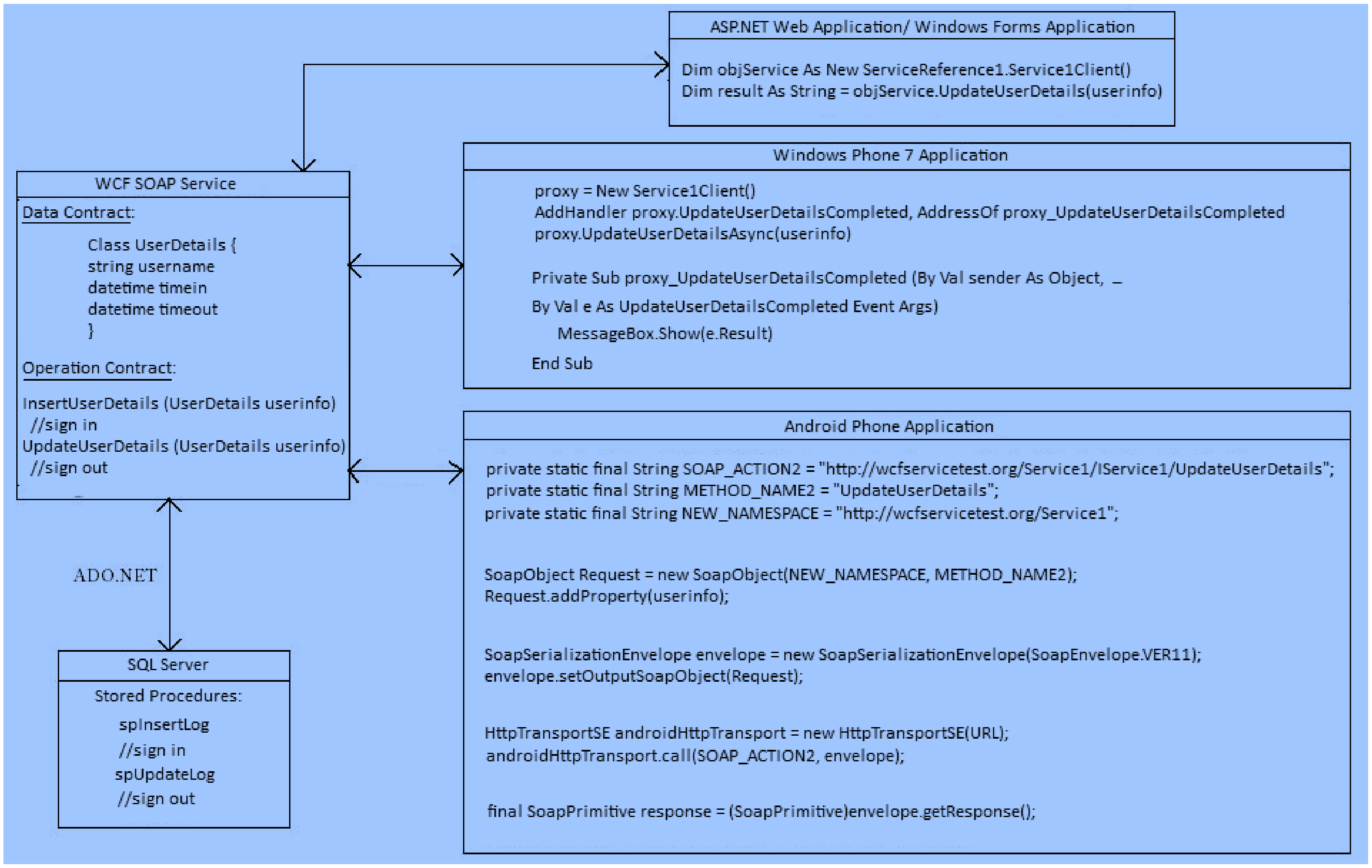
7. TimeClock Application

In order to test and validate the approach of service based architecture for cross platform development, I made a simple TimeClock application which allows users to select their name from a list and sign-in/ sign-out. The name of the user and timestamps of sign-in/sing-out are stored in a database.

The client applications communicate with the WCF service through SOAP and call the exposed methods defined with an operation contract in the service. While executing the methods, the WCF service communicated to the database using ADO.NET to read/write to the SQL database.

The SOAP service has in its data contract, the UserDetails class which has username, time in, time out as the class members. The operation contract consists of the methods InsertUserDetails for adding a record for ‘singing in’ to the database and UpdateUserDetails for ‘signing out’ to update the record and specify the time out field.

InsertUserDetails and UpdateUserDetails both accept an object of class UserDetails and uses ADO.NET to call stored procedures spInsertLog and spUpdateLog, which do the insertion/updating in the database.



```
Dim objService As New ServiceReference1.Service1Client()
Dim result As String = objService.UpdateUserDetails(userinfo)
```

```
Class UserDetails {
string username
datetime timein
datetime timeout
}

InsertUserDetails (UserDetails userinfo)
 //sign in
UpdateUserDetails (UserDetails userinfo)
 //sign out
```

```
proxy = New Service1Client()
AddHandler proxy.UpdateUserDetailsCompleted, AddressOf proxy_UpdateUserDetailsCompleted
proxy.UpdateUserDetailsAsync(userinfo)

Private Sub proxy_UpdateUserDetailsCompleted (By Val sender As Object, _
By Val e As UpdateUserDetailsCompleted Event Args)
    MessageBox.Show(e.Result)
End Sub
```

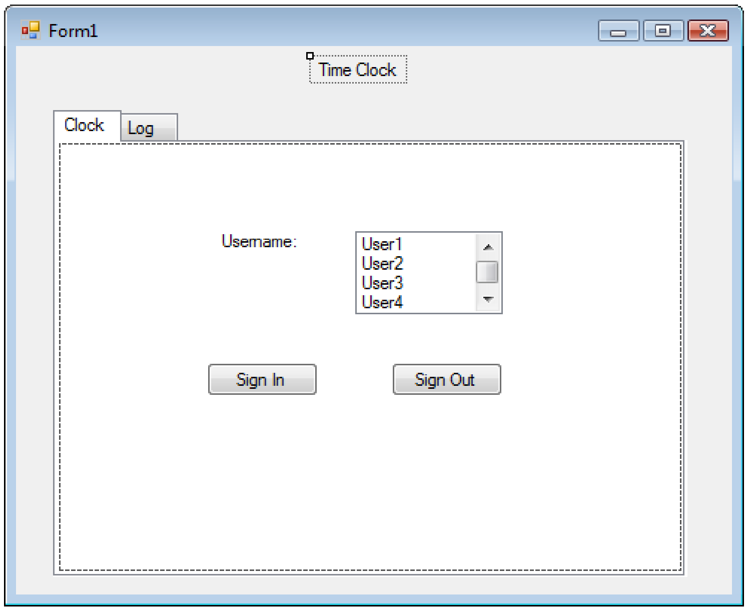
```
private static final String SOAP_ACTION2 = "http://wcfservicetest.org/Service1/IService1/UpdateUserDetails";
private static final String METHOD_NAME2 = "UpdateUserDetails";
private static final String NEW_NAMESPACE = "http://wcfservicetest.org/Service1";

SoapObject Request = new SoapObject(NEW_NAMESPACE, METHOD_NAME2);
Request.addProperty(userinfo);

SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.setOutputSoapObject(Request);

HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
androidHttpTransport.call(SOAP_ACTION2, envelope);

final SoapPrimitive response = (SoapPrimitive)envelope.getResponse();
```

8. ASP.NET Web & Windows Forms Applications

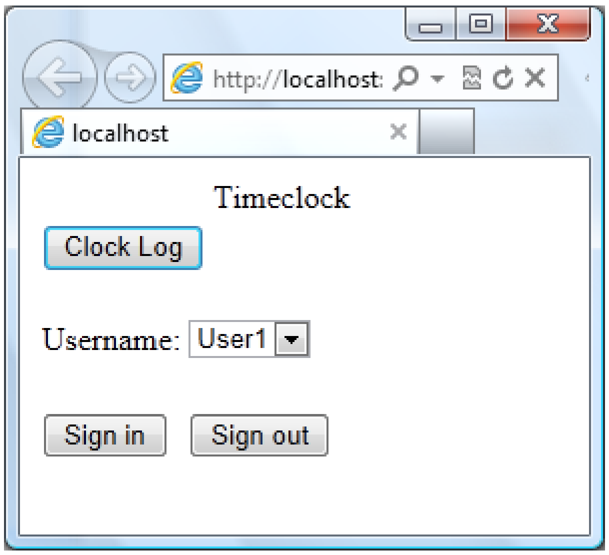
A SOAP based service is utilized in web/windows forms application by adding a service reference through visual studio and then defining a service client referring to the service. The service client is an object of the service class with which you can invoke the exposed methods of the WCF service.

```
Dim objService = New ServiceReference1.Service1Client()
Dim result as String = objService.UpdateUserDetails (userinfo)
```

Here objService is the service client and an object of the WCF service. UpdateUserDetails is invoked by passing the object of UserDetails on the client side.

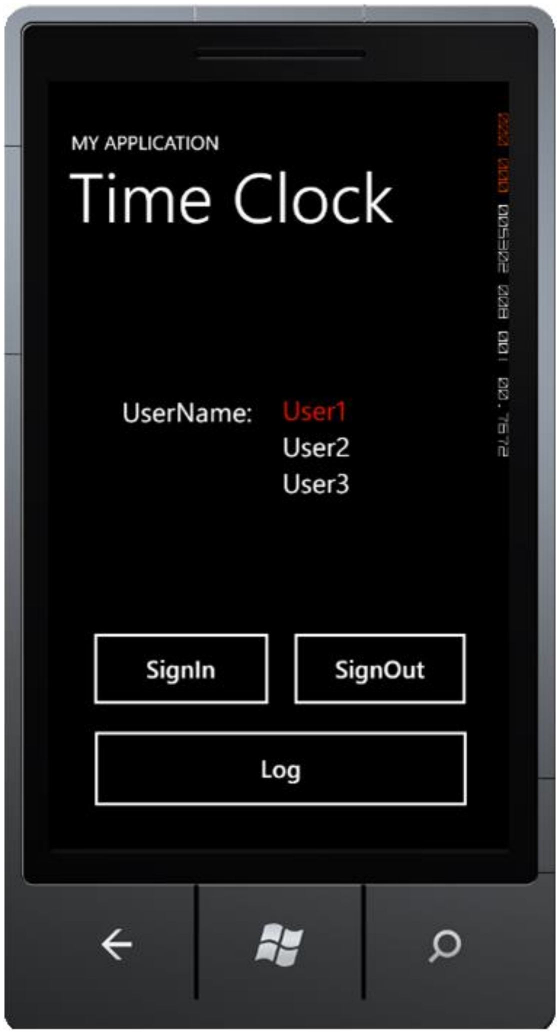


Windows Forms Application



ASP.NET Application

9. Windows Phone 7 Application



A service client is also used in windows phone application to invoke the service methods.

```
Dim proxy As Service1Client
proxy = New Service1Client()
AddHandler proxy.InsertUserDetailsCompleted, AddressOf proxy_InsertUserDetailsCompleted

proxy.InsertUserDetailsAsync(user1)
```

Here, proxy is the name of the service client . In windows phone, the asynchronous version of the methods are used, so that the phone doesn't wait for the response and can carry on with other operations. When the method is done executing and gets a response, InsertUserDetailsCompleted event is fired. To handle the completed event we add a handler to the InsertuserDetailsCompleted at the address of a new method proxy_InsertedUserDetailsCompleted.

```
Private Sub proxy_UpdateUserDetailsCompleted(ByVal sender As Object, ByVal e As
UpdateUserDetailsCompletedEventArgs)
    MessageBox.Show(e.Result)
End Sub
```

proxy_UpdateUserDetailsCompleted is passed with the parameter UpdateUserDetailsCompletedEventArgs so that the response from the InsertUserDetailsAsync is passed on. When the sign in / sign out buttons are pressed, a MessageBox is displayed with either confirmation of the action or an error.

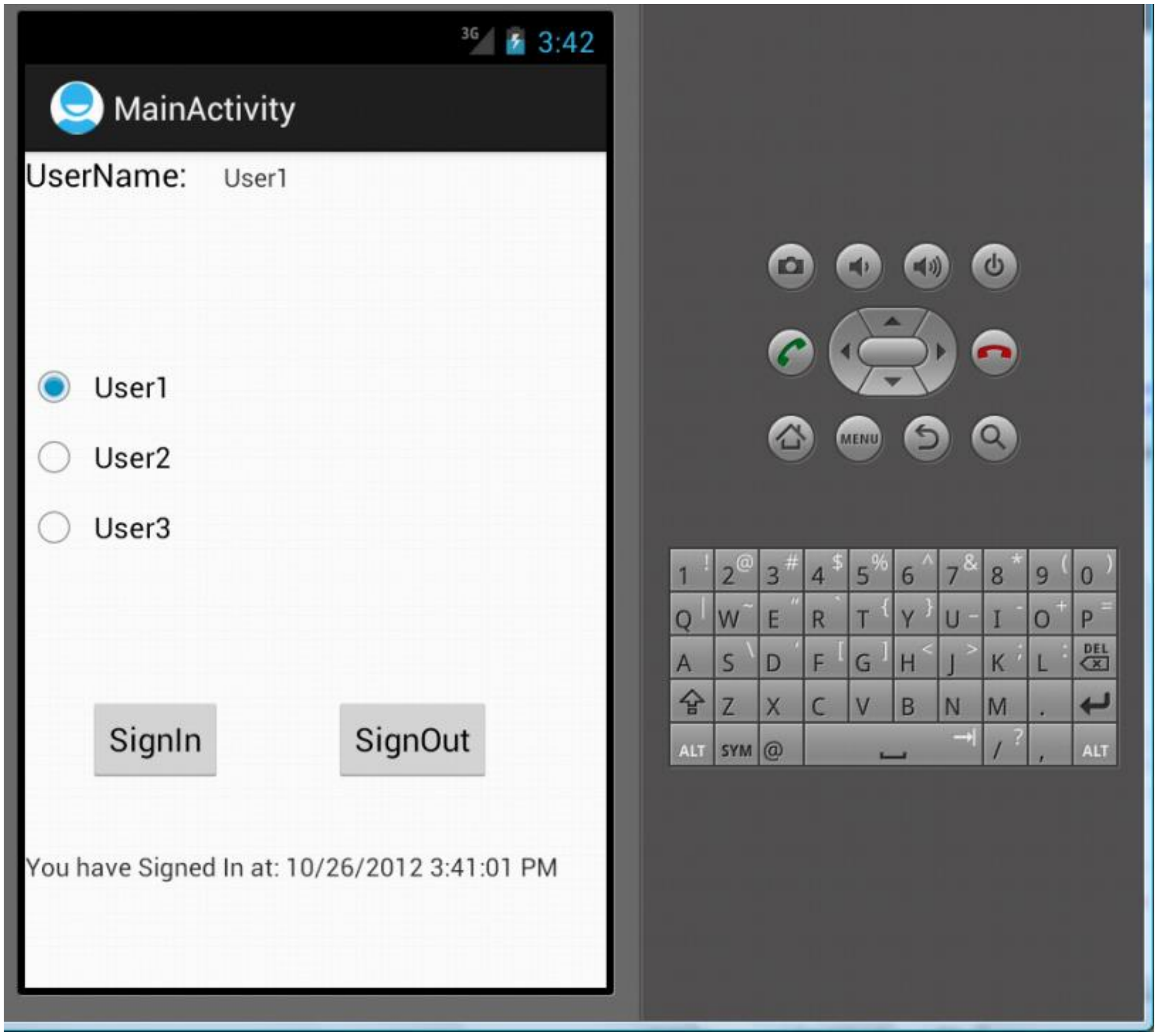
9.1 Problems Faced

The main problem encountered was that the return events from the async function were being called multiple times which increased with every call. This was because I was using a single service client and a new event handler for the completed event was added every time the service operation was called.

This problem was fixed with reinitializing the service client before the async calls and in the page load method. [1]

```
proxy = New Service1Client()
```

10. Android Application



You have Signed In at: 10/26/2012 3:41:01 PM

To call WCF SOAP service on android I used a third party SOAP client library for the android platform called Ksoap2-android.

10.1 Ksoap2-android

Ksoap2-andoid is part of the kSOAP2 SOAP web service client library and is mostly specific to the android platform.

10.2 Complex objects

The classes of the complex objects to be sent the web service using Ksoap2 implement kvmserializable. Kvmserializable requires you to override the following additional methods in the classes:

```
Object getProperty(int arg0)
int getPropertyCount()
void get PropertyInfo(int arg0, Hastable arg1, PropertyInfoinfo)
void setProperty(int arg0, Object arg2)
```

10.3 Referencing SOAP Service

For referencing a SOAP service using Ksoap2, unlike in visual studio, you have to explicitly define the URL's of the service and name of the methods. Visual studio automatically takes care of this when you add a service reference to the project.

In my application I used 3 types of strings; SOAP_ACTION, METHOD_NAME and NAMESPACE.

```
private static final String METHOD_NAME2 = "UpdateUserDetails";

private static final String SOAP_ACTION2 =
"http://wcfservicetest.org/Service1/IService1/UpdateUserDetails";

private static final String NAMESPACE = "http://wcfservicetest.org/Service1";
```

Namespace and method names can be found on the WSDL of the WCF service whereas, soapaction is namespace + method name.

10.4 New Thread

In android 3.0 and above, web requests are not allowed on the main thread. Hence, they must be run on a separate/background thread. So you can either use AsyncTask or create a new thread. In my program I used a new thread.

```
Thread networkThread = new Thread() {
        @Override
        public void run() {
                try {

                //web request call

                } catch (Exception e) {
                e.printStackTrace();
        }
        networkThread.start();

}
```

10.5 SoapOject

The whole web request is an object of SoapObject class, and the complex objects you are sending to the web service for the parameters are added to the soap object as a PropertyInfo. The name of the property info is set to the name of the parameter in the webservice method definition.

```
SoapObject  Request = new SoapObject( New_NAMESPACE, METHOD_NAME);
UserDetails usr = new UserDetails(username, date, date);

PropertyInfo usr = new PropertyInfo();
usr.setNamespace(NAMESPACE);
usr.setName("userInfo");
usr.setType(user.getClass());
usr.setValue(user);
Request.addProperty(usr);
```

10.6 Soap Envelope

Soap Envelope is the container that holds data for both web requests and responses.

```
SoapSerializationEnvelope envelope = new SoapSerializationEnvelop(SoapEnvelope.VER11);

Envelope.setOutPutSoapObject(Request);
```

Other properties of the envelope set true are, dotnet for calling a dotnet service, and implicittypes for automatic mapping of the types.

10.7 MarshalDate

Ksoap2 does not automatically serialize data parameters. But it provides a MarshalDate class that marshals the date for serialization. If sending a date parameter in the request MarshalDate must be used.

```
MarshalDate md = new MarshalDate();
md.register(envelope);
```

10.8 Web Request

To make a call to the web service, you need a connection to the service. The following code creates a http connection to the webservice according to the URL provided. To create a https connection the URL must be changed to https.

```
HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
```

Then web request is called by:

```
androidHttpTransport.call(SOAP_ACTION1, envelope);
```

10.9 Web Response

The response of the SOAP request is handled by:

```
Final SoapPrimitive response = (SoapPrimitive) envelope.getResponse();
```

If a complex object is being received then SoapObject is used to catch it.

10.10 Debugging

For debugging and to see the request and responses you can set androidHttpTransport.debug=true; and then you can use the Log to print androidHttpTransport.requestdumps and androidHttpTransport.responsedumps

To test the web service and compare the SOAP request, you can use a tool called SOAPUI which creates a XML of the SOAP request that the web service is expecting.

10.11 Problems Faced

10.11.1 Namespace Problem

During the implementation of ksoap2 library in the android application, one of the prominent problems faced was related to namespace and the SOAP request envelop. In the early experimenting phase, I created two methods in the WCF SOAP service to add two numbers passed. One took in two numbers as parameters (ksoapAdd) the other (addParam) accepted an object (class testadd) with two numbers as elements. Both these methods returned the sum of the two numbers.

The addParam method worked well and the correct response was received. But ksoapAdd always got a response of 0. On further debugging it was noted that even though the ksoapAdd request message sent the right numbers, the method received 0 for all the elements on the service side.

```
SOAP Request Message:
<v:Envelope xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns:d="http://www.w3.org/2001/XMLSchema"
xmlns:c="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:v="http://schemas.xmlsoap.org/soap/envelope/">
<v:Header />
<v:Body>
<ksoapAdd xmlns="http://tempuri.org/" id="o0" c:root="1">
<n0:num1 xmlns:n0="http://tempuri.org/">
<number_1>25</number_1>
<number_2>25</number_2>
</n0:num1>
</ksoapAdd>
</v:Body>
</v:Envelope>


// num1 is an object of class testadd which has two elements; number_1 &
//number_2
```

```
SOAP Response Message:
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Body>
<ksoapAddResponse xmlns="http://tempuri.org/">
<ksoapAddResult>0</ksoapAddResult>
</ksoapAddResponse>
</s:Body>
</s:Envelope>
```

The problem turned out that the ksoap2 code had an issue with the default namespace of the service set by WCF (http://tempuri.org) even though WP7 didn’t have any problems with it.

This problem was solved by getting rid of the default namespace in the WCF SOAP service by explicitly setting a namespace.

My conclusion was that with the default namespace, the request envelop was unable to map the passed testadd object to the DataContract in which the testadd class is defined in the service. By defining a namespace explicitly to the ServiceContract and DataContract in the WCF service, this changed/rearranged the WSDL and made the DataContract accessible for the incoming SOAP request to map the values of num1 client side to num1 server side .

I changed the namespace to “http://wcfservicetest.org/Service1" by adding the namespace attribute in the following places in the WCF SOAP service.

```
<ServiceContract(Namespace:="http://wcfservicetest.org/Service1")> _
<DataContract(Namespace:="http://wcfservicetest.org/Service1")> _
<ServiceBehavior(Namespace:="http://wcfservicetest.org/Service1")> _
```

<u>Request/Response With New Namespace:</u>

```
SOAP Request Message:
<v:Envelope xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns:d="http://www.w3.org/2001/XMLSchema"
xmlns:c="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:v="http://schemas.xmlsoap.org/soap/envelope/">
<v:Header />
<v:Body>
<ksoapAdd xmlns="http://wcfservicetest.org/Service1" id="o0" c:root="1">
<n0:num1 xmlns:n0="http://wcfservicetest.org/Service1">
<number_1>25</number_1>
<number_2>25</number_2>
</n0:num1>
</ksoapAdd>
</v:Body>
</v:Envelope>
```

```
SOAP Response Message:
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Body>
<ksoapAddResponse xmlns="http://wcfservicetest.org/Service1">
<ksoapAddResult>50</ksoapAddResult>
</ksoapAddResponse>
</s:Body>
</s:Envelope>
```

10.11.2 DateTime Problem

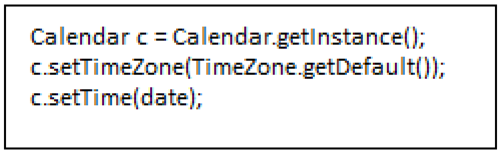
Since Java Date and .Net DateTime different format, the Java Date has to be reformatted to the DateTime standard. Also Ksoap2 requires marshalling of date to so that the XML parser would know how to serialize and deserialize which is provided by the MarshalDate class of the Ksoap2 library.

The problem after using MarshalDate was that time fields sent through the SOAP request was 2 hours in the future. Further debugging revealed that the Marshaldate class implements the dateToString(Date) method of the ISODate class which sets the timezone to GMT.

This turned out to be a feature because if the clients are located across different time zones, its best to save the timestamps in a single standard timezone(GMT).

But if wanted this can be changed by editing the code in ISODate.dateToString(date) to set the timezone.

This sets the timezone to the timezone of the location the client is present:



```
Calendar c = Calendar.getInstance();
c.setTimeZone(TimeZone.getDefault());
c.setTime(date);
```

Part III: Implementing REST based SOA

11. WCF REST Service Application

Implementing a WCF REST based service application is similar to SOAP service application but with some change in settings.

Each methods is either defined as a WebInvoke (PUT, POST) or WebGet(GET) depending on what kind of operation they do.

```
<OperationContract()> _
<WebInvoke (UriTemplate:=”/login”, Method:”PUT”, RequestFormat
:=WebMessageFormat.Xml)> _
Function InsertUserDetails(ByVal userinfo As UserDetails) As String
```

Here, the method is defined as WebInvoke as it performs a PUT operation and puts data in a database. RequestFormat is set to XML; this will return the results in XML.

12. ASP.NET Web, Windows Forms, Windows Phone Application

All the .NET running application invoke the REST methods by using the WebCleint.

```
Dim client As New WebClient
Dim stream As IO.Stream = client.OpenRead(uri)
        Dim reader As New StreamReader(stream)
        Dim s As String = reader.ReadToEnd
Return s
```

Here, a REST method is being invoked with the URI saved to the variable uri. The response of the method is returned as a string.

13. Android Application

In Android, the REST service methods is invoked by assigning the URI of the method to the usi variable and using HttpClient to send the requests.

```
HttpClient httpclient = new DefaultHttpClient();
HttpResponse response;
 String responseString = null;
        try {
            response = httpclient.execute(new HttpGet(uri));

            StatusLine statusLine = response.getStatusLine();

            if(statusLine.getStatusCode() == HttpStatus.SC_OK){

                ByteArrayOutputStream out = new
ByteArrayOutputStream();

                response.getEntity().writeTo(out);

                out.close();

                responseString = out.toString();

            }
```

14. Conclusion

In this research, I explored different technologies for cross-platform mobile development and compatibility of .NET 3.5 with different mobile platforms. I demonstrated that a service oriented architecture using WCF SOAP and REST technologies can be used for cross platform mobile development.

From my experience, the concept of SOAP was easy to understand, but there were many problems and implementing SOAP took the longest time. But REST on the hand, is more difficult to understand but implementing it is much simpler and easier.

In conclusion I recommend using the WCF REST based architecture for most cases. WCF SOAP can be used in cases where high security and custom settings are needed.

15. Resourses

Introducing Windows Communication Foundation in .NET Framework 4
http://msdn.microsoft.com/library/ee958158.aspx

http://en.wikipedia.org/wiki/Windows_Communication_Foundation

(1)http://forums.silverlight.net/t/20252.aspx/1

https://devcentral.f5.com/tutorials/tech-tips/the-minimum-steps-to-utilize-ksoap2

http://javatutorialspoint.blogspot.com/2012/02/android-web-service-access-using-ksoap2.html

http://code.google.com/p/ksoap2-android/

Introduction to WCF REST http://msdn.microsoft.com/en-us/magazine/dd315413.aspx#id0070034