GETTING SERIOUS ABOUT PROVIDING MOBILE WEB SERVICE

Marc Jansen
Computer Science Institute, University of Applied Sciences Ruhr West, Tannenstr. 43, 46240 Bottrop, Germany

Keywords: Mobile Devices, Web Services, Mobile Web Service Provider.

Abstract: The role of mobile devices as Web Service consumers is widely accepted and a large number of mobile applications already consumes Web Services in order to fullfill their task. Nevertheless, the growing number of powerful mobile devices, e.g. mobile phones, tablets, … even raise the question whether these devices can not only be used as Web Service consumers but at the same time also as Web Service providers. Therefore, this paper presents an approach that allows to deploy Web Services on mobile devices by the usage of the well-known protocols and standards, e.g. SOAP/REST and WSDL.

1 INTRODUCTION

In recent years the number of reasonably powerful mobile devices has increased a lot. According to (IDC, 2011) the number of smartphones worldwide calculates to about 300 million units.

On the other hand, these huge number of smartphones does also provide a large number of heterogeneous devices, with respect to the operating system that smartphones are currently using. According to (Tudor, Pettey, 2010), there were at least five different operating systems for smartphones available on the market in 2010.

Therefore, a platform independent mechanism for the communication with services provided by smartphones seems to be necessary in order not to re-implement each service for each of the mentioned operating systems.

Here, nowadays usually Web Services are used in order to provide a standardized and widely used methodology that is capable of achieving a platform independent way to provide services. Unfortunately, in contrast to consuming Web Services on mobile devices, providing Web Services on mobile devices is not yet standardized due to several problems that occur if a service runs on a mobile device.

This paper presents the description of a framework that allows to provide Web Services on mobile devices.

2 RELATED WORK

Probably the first idea of providing Web Service on mobile devices was presented by IBM (McFaddin, Narayanaswami, Raghunath 2003). This work presents a solution for a specific scenario were Web Service were hosted on mobile devices. More general approaches for providing Web Services on mobile devices are e.g. presented in (Srirama, Jarke, Prinz, 2006) and (AlShahwan , Moessner, 2010).

In (Li, Chou, 2010) another approach focussing on the optimization of the HTTP protocol for mobile Web Service provisioning is presented.

Unfortunately, none of the formerly mentioned approaches manages to overcome certain limitations of mobile devices, as described in the next chapter.

The major difference between the former research and the approach presented in this paper is that, to the best of our knowledge, the former research focuses very much on bringing Web Services to mobile devices by implementing server side functionality to the mobile device in question. The approach presented in this paper follows a different approach: from a technical and communication point of view, the mobile Web Service provider communicates as a Web Service client with a dynamically generated Web Service proxy.

This approach provides the advantage to overcome certain problems with mobile Web Services as described in the next chapter. Furthermore, this approach does not rely on an efficient server side implementation of Web Services on the mobile device, which allows to implement a very lightweight substitution to a usual application server were a usual Web Service is

Jansen M..
GETTING SERIOUS ABOUT PROVIDING MOBILE WEB SERVICE.
DOI: 10.5220/0003935402680271
In *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (WEBIST-2012), pages 268-271
ISBN: 978-989-8565-08-2

running in.

Since there is nothing like a free lunch, this approach also provides some drawbacks, e.g. the presented approach implements a polling mechanism that permanently polls for new service requests. Therefore, this approach provides an overhead with respect to the network communication and the computational power of the mobile device. At least the part with the computational overhead can dramatically be reduced by adjusting the priority of the polling mechanism, according to the priority of the provided Web Service.

Another drawback of the presented approach is that it relies on a publicly available proxy infrastructure for the part of the framework that dynamically generates the Web Service proxies. This drawback might be overcome, e.g. if mobile telecommunication companies provide this kind of infrastructure centrally.

In contrast to the before mentioned approaches, the approach presented in this paper differs with respect to one major aspect: from a network technical point of view, there is no server instance installed on the mobile device. Therefore, a certain Web Service client does not call the Web Service on the mobile device directly but calls a centrally deployed proxy. The Web Service running on the mobile device polls in regular intervals for any new message requests of interest. The exact sequence of the different messages and events are described in more detail in section 4.1. Since especially polling mechanisms also provide a certain drawback, one of the major questions concerning the presented approach is the question of benefits and drawbacks of the polling mechanism and in particular whether the benefits justify the drawbacks.

As already mentioned before, one of the major problems by dealing with Web Services on mobile devices is the fact that mobile devices usually switch their networks pretty often. Therefore, the Web Service running on a mobile device is usually not available under a fixed address, which leads to a number of problems for the consumer of a mobile Web Service. Besides the usual network switch, also the fact that mobile devices are usually not meant to provide 24/7 availability, but are designed towards providing his user with the possibility to exploit certain service, e.g. phone calls, short messages, writing and receiving emails, … yield to the problem that mobile devices might get switched off by the user. Hence, the provided Web Service might not only be available under different network addresses but might not be available at all.

All these drawbacks can be solved by using the here presented approach. By using the central proxy, the service requests of a certain Web Service client can be stored and if the mobile Web Service is running, he can pull for service requests that of interest to him. Since from a technical point of view the Web Service provider only acts as a client to the Web Service proxy, the potentially changing network addresses of the mobile do not provide a problem at all.

Furthermore, one of the major drawbacks of the described polling mechanism can be limited by adjusting the priority of the Web Service running on the mobile device, resulting in a lower frequency of the polling for the service request.

3 SCENARIO DESCRIPTION

The major idea for the implementation of the middleware is to provide a Web Service proxy, according to the proxy design pattern (Gamma et al., 2005), in order for being able to overcome certain problems in mobile scenarios as described by (Svensson, 2009). One major problem here is that mobile devices usually switch networks pretty often, e.g. at home the mobile phone might be connected to a WiFi network, at work the connection might be established through another WiFi network and on the way from work to home the mobile phone might be connected to a GPRS/UMTS-network. Each of these different networks, provide different IP addresses and possibly different network constellations, e.g. private IP addresses with network address translation (NAT), where the Web Services running on the device are not directly accessible from the internet, or public IP addresses.

In order to overcome the problem of constantly changing IP addresses, the presented approach implements a Web Service proxy that dynamically creates a proxy for each Web Service that gets deployed on a mobile device. The created proxy allows to receive service requests as a representative to the actual service, store this service request along with the necessary data. In the next step, the mobile Web Service provider continuously polls for requests to its services and sends the result back to the dynamically generated Web Service proxy. Afterwards the proxy send the result back to the original client.

4 IMPLEMENTATION

The major goal of the work presented here is to provide a solution to the formerly described scenario. Therefore, we implemented a middleware that allows the provision of Web Services on mobile devices. Here, the standard protocols (e.g. WSDL for the description of the Web Service interface, SOAP/REST as the standard network protocol and http as the usual transport protocol) are used in order to provide no additional effort on the client side for requesting a mobile Web Service.

The following three sections provide a short introduction to the services offered by the middleware, afterwards the communication between the mobile Web Service provider and the Web Service client/consumer is described and last but not least some details about the Java based implementation for the test scenario are presented.

4.1 Communication between the Mobile Web Service and Its Clients

In order to explain the communication that is necessary for service request from the Web Service client to the mobile Web Service provider, we modelled the communication flow within the sequence diagram shown in Figure 1.

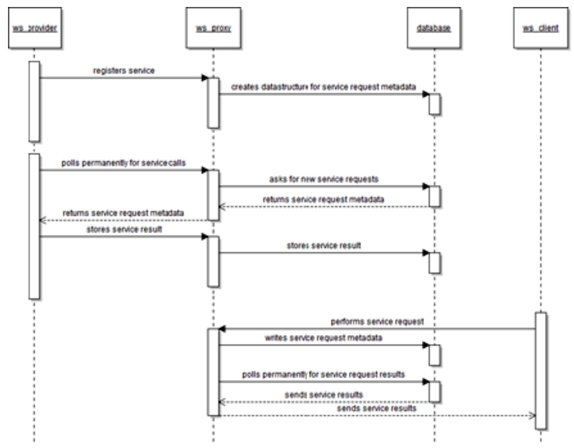
ws_provider
ws_proxy
database
ws_client
registers service
creates datastructure for service request metadata
polls permanently for service calls
asks for new service requests
returns service request metadata
returns service request metadata
stores service result
stores service result
performs service request
writes service request metadata
polls permanently for service request results
sends service results
sends service results

Figure 1: The UML sequence diagram for the communication between a Web Service provider and its client.

First of all the mobile Web Service provider needs to register its service with the Web Service proxy. As part of the service registration process the Web Service proxy creates the necessary data structure for storing the service requests in the database.

After the mobile Web Service provider registered his service, it permanently polls the Web Service proxy for new service request. The Web Service proxy asks the database if a new service request for the according mobile Web Service provider is available and if so, returns the requests metadata to the mobile Web Service provider. After receiving the metadata of a new service request, the mobile Web Service provider performs the service and sends the result of the service to the Web Service proxy that directly stores the result in the database.

From a client point of view, the Web Service client simply calls the service provided by the Web Service proxy. While receiving a new service request, the Web Service proxy stores the necessary request metadata in the database. Afterwards the Web Service proxy directly starts to permanently poll the database for the result of the according service request. Once the mobile Web Service provider has finished performing the request and stored the result (via the Web Service proxy) in the database, the Web Service provider is able to send the result of the according service request back to the client.

4.2 An Example Implementation

In order for being able to test the described approach according to its performance, we implemented the Web Service proxy in Java. Additionally, the mobile Web Service provider was implemented for Android. Here, we focused on an intuitive and easy way for the implementation of the Web Service. Therefore, we oriented ourselves on the JAX-WS (Java API for XML-Based Web Services) described in he Java Specification Request 224 (JSR 224).

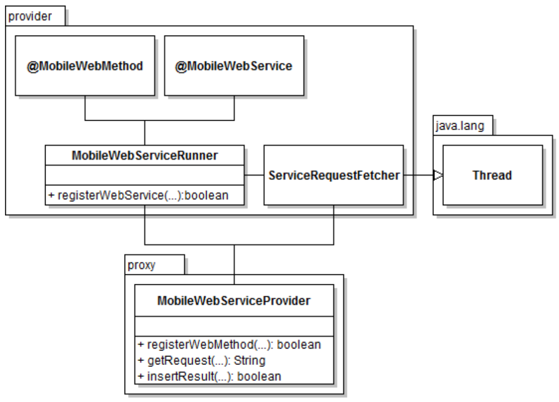

Figure 2: UML class diagram of major parts of the example implementation.

The major idea that we used from JAX-WS was that a Web Service can easily be implemented by the use of two different annotations: the *@MobileWebService* annotation marks a class as a Web Service and methods within this class can be marked as methods available through the mobile

Web Service with the help of the *@MobileWebMethod* annotation.

The basic relationships between the major classes of the example implementation are shown in. For the sake of simplicity and transparency, less important classes (and methods of each class) have not been modelled.

Basically, the implementation consists of two packages. One package that is usually deployed on a server that is reachable from the internet via a public IP address, this is the proxy package. Here, we find one class that implements the necessary methods for the registration of a new mobile Web Service, the permanent polling from the mobile Web Service for the service request metadata and the method that allows to store the result of the service request in the database. All of these methods are themselves reachable as Web Services, so that the communication between the instance running the mobile Web Service and the Web Service proxy is also completely Web Service based.

The provider package holds one of the major classes, the *MobileWebServiceRunner* where the mobile Web Service gets deployed to. This class is comparable to an application server in usual Web Service environments, but with a lower footprint. This is extremely important to mobile devices due to their limited resources. Additionally, this package provides the two formerly mentioned annotations allowing to make a class as a mobile Web Service and accordingly a certain method of such a class as a mobile Web Method. Last but not least, this package also implements the *ServiceRequestFetcher* class. This class inherits the *java.lang.Thread* class since its responsibility is to permanently poll the Web Service provider for new service requests.

5 CONCLUSIONS

The presented approach is capable of solving some of the problems that usually occur while providing Web Services on mobile devices, e.g. the problem of permanently changing IP addresses. Furthermore, the overhead that is inherent in the presented approach does not seem to be a show stopper. As shown, the performance in usually available mobile networks like UMTS or GPRS, is comparable to usual Web Service calls.

Therefore, it could be said that the presented approach provides an interesting alternative to the usual Web Service provisioning by mobile devices where the mobile device acts as a server also from a technical point of view. It eliminates certain problems that usually occur if mobile devices provide Web Service provider infrastructures, and the resulting drawbacks from the performance point of view are acceptable.

ACKNOWLEDGEMENTS

This work was partly supported by an Amazon AWS research grant.

REFERENCES

AlShahwan, F., Moessner, K., 2010. Providing SOAP Web Services and REST Web Services from Mobile Hosts, In: *Fifth International Conference on Internet and Web Applications and Services* (ICIW).

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995. Design Pattern – Elements of Reusable Object-Oriented Software, *Addison-Wesley*.

IDC Worldwide Quarterly Mobile Phone Tracker, January 27, 2011.

Li, L., Chou, W., 2010. COFOCUS – Compact and Expanded Restful Services for Mobile Environments, In: *Proceedings of the 7th International Conference on Web Information Systems and Technologies*, Noordwijkerhout, The Netherlands

McFaddin, S., Narayanaswami, C., Raghunath, M., 2003. Web Services on Mobile Devices – Implementation and Experience, In: *Proceedings of the 5th IEEE Workshop on Mobile Computing Systems & Applications* (WMCSA'03), Monterey, CA

Srirama, S., Jarke, M., Prinz, W., 2006. Mobile Web Service Provisioning, In: *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services* (AICT/ICIW 2006), Guadeloupe, French Caribbean

Svensson, D., 2009. Assemblies of Pervasive Services. Dept. of Computer Science, Institutional Repository – Lund University.

Tudor, B., Pettey, C., 2010. Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010, Smartphone Sales Increased 96 Percent, Gartner, http://www.gartner.com/it/page.jsp?id=1466313, last visited 19.11.2011