Nokia Web Services Framework for Devices – a Service-oriented Architecture
## Contents

<table>
<thead>
<tr>
<th>Abstract</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td>3</td>
</tr>
<tr>
<td>What is a service-oriented architecture, and why is it good?</td>
<td>3</td>
</tr>
<tr>
<td>Services in the mobile environment</td>
<td>3</td>
</tr>
<tr>
<td><strong>Nokia’s device Web services implementation</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Service example</strong></td>
<td>5</td>
</tr>
<tr>
<td>The AOL Radio client</td>
<td>5</td>
</tr>
<tr>
<td>Mobile-based services</td>
<td>6</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>6</td>
</tr>
</tbody>
</table>

## Abstract

Nokia has a vision of the mobile phone as a full participant in networked service-oriented architectures. This document describes a new framework for part of the next-generation Nokia mobile software platforms. The framework provides rich support for service-oriented applications design in Java or C++ and a method of abstraction that allows developers to concentrate on application logic, rather than system calls. It also includes support for standards such as HTTP, XML and SOAP, as well as a framework for identity management, security, service discovery and session management. Finally, the new framework makes it easier to develop and deploy new services that are offered both to and by mobile phones.
Introduction

Future versions of Nokia mobile phone software will include a unified framework for developing applications for mobile phones, allowing them to become full participants in a networked service-oriented software architecture.

This framework should simplify the software development of applications that rely on components based both on the device and on remote systems, via a message-passing architecture that uses open standards such as HTTP, SOAP and XML – in other words, a service-oriented architecture.

What is a service-oriented architecture, and why is it good?

A service-oriented architecture comprises loosely coupled software systems that support direct interactions with other software systems using standardized, XML-based messages exchanged over standard Internet protocols such as HTTP.

Services tend to be autonomous and do not rely on an omnipresent system that has complete control over all parts of the architecture. This creates a style of development that is ideal for mobile phones.

A service-oriented architecture makes boundaries irrelevant, meaning that services may be separated by large geographical distances, across multiple business entities and running in distinctly different software environments. Messages between services are standardized so that such distinctions can be ignored by the application developer.

Services in the mobile environment

Many service providers currently offer services that allow information to be “pushed” to a mobile phone, or offered via a restricted web-browser interface, not always suited to mobile phones. With a service-oriented architecture, it is possible to offer services that fully use the power of the mobile phone, allowing it to become both a full consumer of services, but also a provider of services.

Web services promise to provide a way of connecting disparate environments, bridging mobile phone applications, mobile network servers and application servers. This is achieved by passing standardized messages between services. This allows service developers to concentrate on building their service, rather than having to also build networking and security support into the service.

It will be possible to download web service applications directly to devices, via another Web service. Such downloaded service may be able to both provide and consume web services. As such services are available via standardized, open interfaces, mobile network operators will be able to offer services used both by applications on the phone, and other networked service and content providers.

Enterprises that implement an infrastructure for authentication and security based on open standards will be able to develop vertical applications. The mobile workforce is a self-evident user group, but even external parties such as customers, partners and suppliers can use the applications, since a high level of security can be maintained.

Nokia mobile phones will offer services across the network. This would allow, for example, a network calendar service to send a message to a mobile user’s phone resident calendar, requesting a meeting be added to the recipient’s schedule.
Nokia’s Web services framework is based on open standards, and implements a service-oriented architecture that allows the application developer to be largely unaware of the complications involved in passing messages between services, or authentication, service discovery and other details.

Both Java and Symbian C++ APIs are provided, allowing programmers to choose the language most appropriate for the application. This enables both Java J2ME and C++ applications to take advantage of the service-oriented architecture and provides a unified model for multiple development environments.

Figure 2 shows that the framework comprises several sub-frameworks and components that client applications and other framework components can access through various application programming interface (API) methods.

On an architectural level, the service invocation framework is composed of framework providers that offer specific functions to calling applications:

- The Liberty framework provider offers APIs that allow the client to access services conforming to the various Liberty ID-FF and ID-WSF specifications.
- Service discovery is accessible to client applications, as well as to other parts of the architecture. The Liberty framework provider may need direct access to service discovery facilities to be able to provide service to a calling application.
- Service access may also have associated policy requirements, hence the policy services block.

Service discovery allows client applications to ask for a particular service type and be provided with both the information required to find that service (for example a concrete SOAP endpoint for the service) and any security tokens required for access by the service provider.

The XML services are provided to allow client applications and other parts of the framework to directly parse and manipulate XML.

At some point, messages will probably be passed over a communications channel. With SOAP messages, this will most likely use either the standard HTTP binding for SOAP, or the Liberty Reverse HTTP Binding for SOAP (known informally as PAOS).
Service example

The AOL Radio client

An early implementation was used for the development of an AOL Radio client. The AOL Radio client is a Symbian application that simply requests a connection to an AOL Radio service using the Liberty framework provider. This component takes care of all the details of service invocation including discovery, authentication, security, session management, SOAP processing and message transport. The AOL Radio client only needs to construct and parse the radio service specific messages; it does not need to worry about how this message is sent to the service, or how the message is authenticated.

Once the service has been accessed using the framework, radio service audio is streamed to the phone-based client. The client takes care of the AOL Radio service audio – the AOL Ultravox™ streaming protocol and the AAC decoding.

The code fragment shown below indicates what is involved in starting up a service client such as the AOL Radio client.

```java
public class RadioClient extends com.nokia.web_service.BaseClient {
    // the requested service type is for the AOL Radio service
    private static String SERVICE_TYPE = “http://radio.aol.com/xsd/2.0/ns/”;

    // other code

    public RadioClient() throws com.nokia.web_service.RemoteException {
        super(SERVICE_TYPE); // inherit the base client functionality
        this.parser = new Parser(this, getXMLReader()); // get an XML parser
        this.interface = new RadioUI(this); // create a new user interface
        this.getRadioChannels(); // get the list of radio channels available from the radio service
    }

    // more code
}
```

The AOL Radio service namespace is accessed via the SERVICE_TYPE constant. The application extends the functionality present in the service invocation framework itself, which discovers the AOL Radio service, and provides an application ‘handle’ to the service. At that point, the application may, for example, retrieve a list of channels from the radio service, which arrive in an XML document. The application itself does not need to know that the request for a channel listing went out across a network, or that the service required authentication before releasing a channel listing.
Mobile-based services

As mentioned above, Nokia sees the mobile phone not only as a Web Service Consumer (WSC), but also as a Web Service Provider (WSP). Thus, it will be possible for a mobile phone to host services that may be offered to other services on the network.

In the example below, the mobile phone is hosting services that offer the device location, and voluntary profile information describing the user of the device.

In step 1 of Figure 4, the device requests a portal page, and advertises to the portal that it provides Liberty ID-SIS location and personal profile services where the user’s information is stored.

The portal service is not an identity-based service, just one that can be personalized according to identity-related information, such as gender, etc.

The portal now knows that the device is offering a personal profile service and asks, via a web-service request, for the user’s gender and location.

In step 3, the profile service hosted by the device provides the location and gender attributes to the portal and in return (step 4), the user is treated to a personalized front page, based on gender and location.

Network service providers can now provide privacy-protected, personalized services to mobile phone users.

Nokia believes that a service-oriented architecture brings great benefits to mobile phone users, mobile network service providers and software developers. We see the mobile phone as a full participant in a network of services, both as a client for exciting new services such as AOL Radio, and a provider of services hosted on the device itself. All of this will be made possible by the existence of an application development framework that fully supports XML, Web services and the associated standard specifications for running services, such as the ID-FF and ID-WSF suites specified by the Liberty Alliance, and WS-Security specified by OASIS.

Summary