

Software &
Engineering
Architecture

Mobile Web Services

Dr. Thomas Wieland
Siemens AG, Corporate Technology
OOP 2002, München

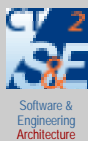
Software &
Engineering
Architecture

Agenda

- **Why Mobile Web Services?**
- **How to implement them**
 - Microsoft Mobile Internet Toolkit for .NET
 - Microsoft Smart Device Extensions
 - PocketSOAP
 - kSOAP
 - SOAP on a Linux PDA
- **What do you need them for?**

Software &
Engineering
Architecture

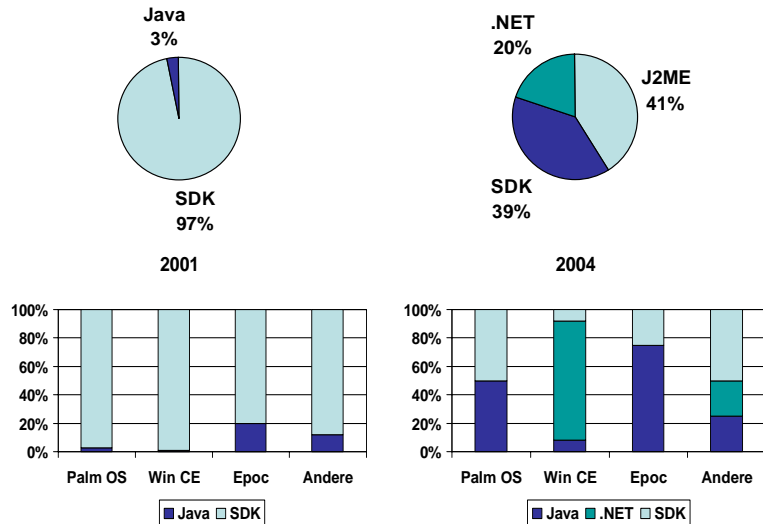
Why Mobile Web Services?

Software &
Engineering
Architecture

Challenges In the Development Of Mobile Applications

- **Differences of the end devices**
 - Often only limited CPU and memory resources are available
- **Business logic almost completely on the server**
- **User is in motion**
 - Session management and roaming necessary
- **Easy of use**
 - No ample client installation should be required
 - Easy to update
- **Efficient development**
 - Leverage investments in knowledge for the desktop world
 - Use experiences of your developers

Market Study: Intelligence On Mobile Devices



Mobile Web Services

5

© Siemens AG, T. Wieland, 27.01.2002

En Vogue: Web Services

- A web service is an application that allows to call its methods via a web interface.
- This way information residing on a web server can be used by other *programs*.
- The call of the methods is done via SOAP, the description of the interfaces in WSDL, both XML languages.
 - That means: web service = SOAP + WSDL
- Also the connection to mobile devices is possible.

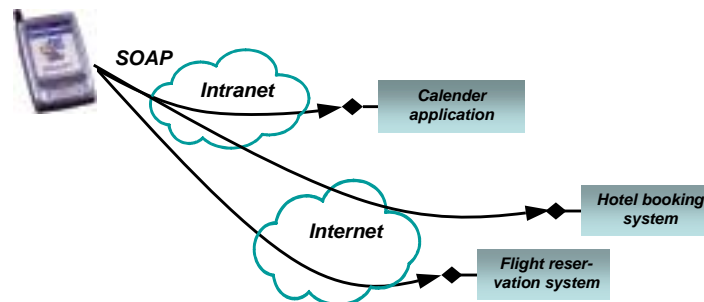
Mobile Web Services

6

© Siemens AG, T. Wieland, 27.01.2002

Example Of a Web Service

- Data from a personal calendar can be cross-checked with flight schedules.
- If something changes on one side, the other can be notified immediately.



Mobile Web Services

7

© Siemens AG, T. Wieland, 27.01.2002

Advantages Of the Web Service Paradigm

- Infrastructure services like user authorisation, backup etc. can be seamlessly integrated from external sources.
- IT specialists can concentrate on the business rather than on infrastructure
- By combining internal and external services, completely new types of applications can arise.
- The interoperability of heterogeneous platforms is increased considerably
 - Especially small (smart) devices are easier to integrate

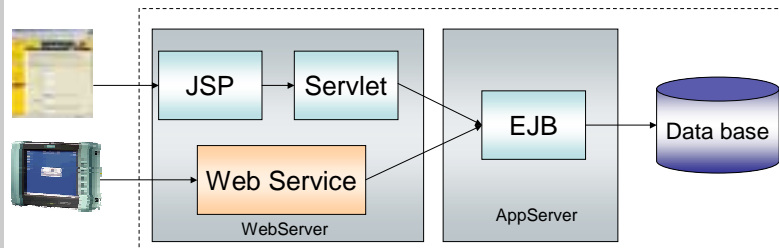
Mobile Web Services

8

© Siemens AG, T. Wieland, 27.01.2002

How Can I Use Web Services Today?

- Many web sites offer input via forms
- As an alternative to the form, the input can also be accepted as a web service call
- Example:
 - Postal services offers form to confirm reception of a package
 - Offered as a web service, a special client application on a mobile device can send this information directly to the server (e.g. after caching them locally while disconnected)



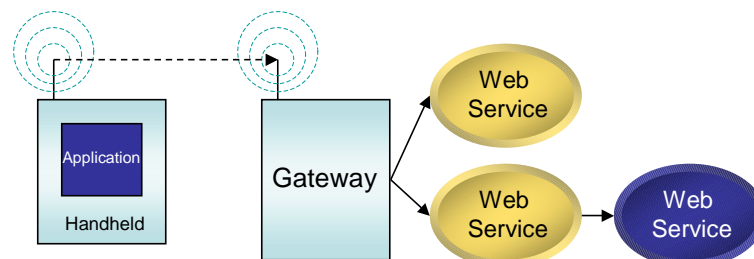
Mobile Web Services

9

© Siemens AG, T. Wieland, 27.01.2002

Usage Of Mobile Web Services

- Applications are installed on the device
 - Perhaps via automatic deployment (e.g. as a MIDlet or a downloaded .NET component assembly)
- Applications provide the coordination logic only to guarantee fast reactions to user interactions
- Business logic is distributed on the network in form of web services



Mobile Web Services

10

© Siemens AG, T. Wieland, 27.01.2002

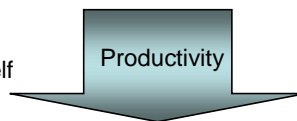
Implementations For Mobile Devices



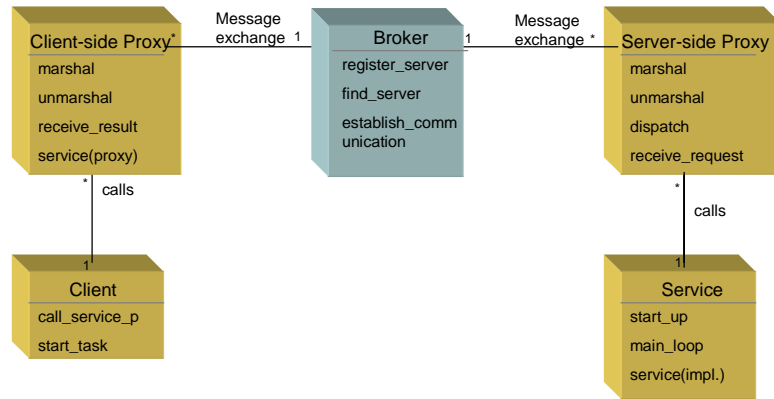
Ingredients For A SOAP Environment

- **An XML parser**
 - Used to process the incoming SOAP messages
 - Can be programmed by yourself, but is less flexible then
- **A listener for the HTTP port**
 - In general port 80
 - Catches all messages on this port
 - May throw away all non-SOAP messages
- **A client proxy**
 - Offers an interfaces for local access similar to the one of the server
 - Packs the calls to a SOAP message
 - Processes the incoming replies

Disadvantage
of a do-it-yourself
solution:



The Broker Pattern



Source: F. Buschmann et al.: "Pattern-oriented software architecture", Wiley, 1996

Available Toolkits

- **Microsoft Mobile Internet Toolkit for .NET**
 - Offers web forms that run on the server
- **.NET Smart Device Extensions**
 - Comprises the Compact Framework (.NET for small devices)
 - Currently still a "technology preview"
- **PocketSoap**
 - Complete open source SOAP implementation for pocket PCs (e.g. Compaq iPaq) under Windows CE
 - Also used/recommended by Microsoft
- **kSOAP**
 - Open source version for the J2ME/CLDC, e.g. on mobile phones



Microsoft Mobile Internet Toolkit

- **Add-on for Visual Studio .NET**
 - Offers "mobile web forms"
 - Recognizes the devices, generates WML 1.1, HTML 3.2 or cHTML
- **Some concept as for web forms**
 - Controls run at the server (attribute `runat="server"`)
 - All operation via ASP.NET pages
- **Web services can be called from these pages**
 - The mobile device does not call the web service directly
 - The logic still resides on the server
 - Corresponding to an addition level of indirection for content adaptation



Example: Temperature Converter °F -> °C

```
<%@ Page Codebehind="Tempconv.cs" Inherits="TempConv.TempPage"
Language="C#" %>
<%@ Register TagPrefix="mobile"
Namespace="System.Web.UI.MobileControls"
Assembly="System.Web.Mobile" %>

<mobile:Form runat="server">
  <mobile:Label runat="server">Temperature in °F</mobile:Label>
  <mobile:Textbox runat="server" id="TempEdit"/>
  <mobile:Command runat="server" OnClick="Respond" Text="Convert"/>
</mobile:Form >

<mobile:Form runat="server" id="ResponseForm">
  <mobile:Label runat="server" id="INewTemp">0</mobile:Label>
</mobile:Form>
```

Example: Temperature Converter °F -> °C



Mobile Web Services



19

© Siemens AG, T. Wieland, 27.01.2002



Software & Engineering Architecture

.NET Smart Device Extensions

- **Portable and small .NET CLR for "smart" devices**
 - Comprises Compact Framework (slim .NET framework)
- **Leverages Visual Studio .NET**
 - "Managed" application are executed directly
 - Visual Studio .NET used for debugging
 - Offers pocket PC emulator
- **Works complementary to the operating system**
 - Windows CE responsible for scheduling, display of the user desktop, input processing, resource management, etc.
- **Supports ADO.NET and SQL Server data persistence in disconnected situations**
- **Does not require a pre-installed Compact Framework on the device, but copies the runtime environment and libraries during deployment!**

Mobile Web Services

20

© Siemens AG, T. Wieland, 27.01.2002



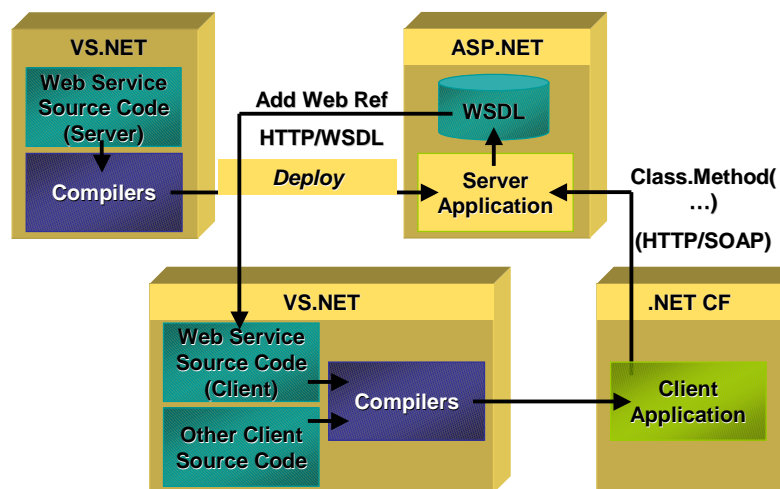
Software & Engineering Architecture

Web Service in SDE

- SDE supports web services as client
- Data and user interfaces can be separated
 - A "rich client" can access numerous services
- Separation according to the publisher/consumer design pattern
- Use web services like in the Internet
- Client application can control packeting of data transmission
- .NET CF used the proxy code generated by Visual Studio .NET
 - Supports synchronous as well as asynchronous interfaces



Development Of Web Services



PocketSOAP

- **Open source project under MPL (mozilla license), developed by Simon Fell**
 - Installed as a COM component, so it is available for all applications
 - Programming it in VB is the easiest way, but via COM interop also accessible from .NET (CF) code
 - Uses the Expat XML parser
- **Large feature set; PocketSOAP (1.2) supports**
 - Simple data types, base64 binary data, arrays, multi-dimensional arrays, partial and sparse arrays, complex types and many more
 - SOAP headers
 - HTTP incl. proxies, authentication, proxy authentication and SSL
- **Runs under Windows 9x/ME/2000/XP and CE 3.0/PocketPC 2002**
 - Supports StrongARM, SH3 and MIPS architectures
- **More tools from the same author available**
 - WSDL proxy generator
 - SOAP server services
 - Proxy trace server to trace SOAP messages



Software &
Engineering
Architecture

Simple Calls From VB Script

```
dim envelope as PocketSOAP.Envelope
set envelope = CreateObject("PocketSOAP.Envelope")
envelope.methodName = "getQuote"
envelope.URI = "urn:xmethods-delayed-quotes"
envelope.Parameters.Create "symbol", "MSFT"
```

```
dim transp as PocketSOAP.HTTPTransport
set transp = CreateObject("PocketSOAP.HTTPTransport")
transp.Send "http://services.xmethods.net/soap", envelope.serialize
envelope.Parse transp
wscript.echo e.Parameters.Item(0).Value
```



Software &
Engineering
Architecture

SOAP Server On The PocketPC

- **By simple means, a SOAP server even on a PocketPC becomes possible**
 - WinSock control listens to port 80
 - Listener receives and send HTTP messages
 - SOAP parser employing MSXML
 - Response function builds simple reply string
- **See Christian Forsberg's article**
 - <http://www.microsoft.com/MOBILE/developer/technicalarticles/hostingweb.asp>
- **Great variety of applications**
 - Context information about the user (is at office, meeting, customer visit)
 - Status request
 - ...



Software &
Engineering
Architecture

kSOAP



- **Grown out of the Enhydra project**
 - Open source software under EPL
 - Meanwhile released in version 0.99 (Dec. 01)
- **Subset of SOAP 1.1**
 - Support for data types like large decimals or base64 encodes arrays is optional and can be added
 - Multi-dimensional arrays are not supported
 - Good interoperability with other implementations
- **Developed for Java 2 MicroEdition (J2ME), CLDC**
 - Especially for devices like mobile phones
 - Via an API converter (ME2SE) it can also be used with Java standard edition



Software &
Engineering
Architecture

Example with kSOAP

```

HttpTransport transport = new HttpTransport ("http://212.99.131.154:7300/",
      "\http://www.shinkatech.com/Currency/CurrencyConverter.calculateExchangeRate");

SoapObject convert = new SoapObject (serviceNamespace, "convert");

convert.addProperty ("currency", fromField.getString ());
convert.addProperty ("amount", new Float (amountField.getString ());
convert.addProperty ("toCurrency", toField.getString ());

SoapObject request = new SoapObject (serviceNamespace, "calculateExchangeRate");
request.addProperty ("convert", convert);

ClassMap classMap = new ClassMap();
classMap.prefixMap = new PrefixMap (classMap.prefixMap, "m",
      "http://www.shinkatech.com/CurrencyConverter/message/");
transport.setClassMap (classMap);

String result = transport.call (request).toString ();
result = result.substring(0,result.indexOf(".")+3);

resultItem.setLabel ("amount :");
resultItem.setText (" "+result);

```

Source: enhydra.org

SOAP On The Linux PDA

- **Agenda VR 3 runs Linux**
 - Special Linux adaptation (based on linux-vr.org), MIPS processor
 - Connectivity with serial or IR interface or built-in modem
 - Graphical UI using Fast Light Toolkit (ftk.org)
- **C++ developed uses GCC for cross compiling**
- **Approx. 3 MB Platz in the file system of VR 3**
- **For SOAP, the libxml2 can be used**
 - From Gnome project (xmlsoft.org)
 - Fast and comfortable parser
 - Small footprint
 - Comprises even HTTP sender and listener



Example: Create Request

```
xmlDocPtr getRequest(xmlChar* term, xmlChar* srcLang, xmlChar* trgLang) {
    xmlDocPtr doc;
    xmlNodePtr tree, subtree, subsubtree;
    xmlNsPtr SOAP, xsd, xsi, ns1;
    doc = xmlNewDoc((xmlChar*)"1.0");
    doc->children = xmlNewDocNode(doc, NULL, (xmlChar*)"Envelope", NULL);
    SOAP = xmlNewNs(doc->children, (xmlChar*)"http://schemas.xmlsoap.org/soap/envelope/",
        (xmlChar*)"SOAP-ENV");
    xmlSetNs(doc->children, SOAP);
    tree = xmlNewChild(doc->children, SOAP, (xmlChar*)"Body", NULL);
    subtree = xmlNewChild(tree, NULL, (xmlChar*)"getTranslations", NULL);
    ns1 = xmlNewNs(subtree, (xmlChar*)"urn:demo1:translate", (xmlChar*)"ns1");
    xmlSetNs(subtree, ns1);
    xmlSetNsProp(subtree, SOAP, (xmlChar*)"encoding-style",
        (xmlChar*)"http://schemas.xmlsoap.org/soap/encoding/");

    subsubtree = xmlNewChild(subtree, NULL, (xmlChar*)"term", term);
    xmlSetNs(subsubtree, NULL);
    xmlSetNsProp(subsubtree, xsi, (xmlChar*)"type", (xmlChar*)"xsd:string");

    subsubtree = xmlNewChild(subtree, NULL, (xmlChar*)"srcLanguage", srcLang);
    xmlSetNs(subsubtree, NULL);
    xmlSetNsProp(subsubtree, xsi, (xmlChar*)"type", (xmlChar*)"xsd:string");

    subsubtree = xmlNewChild(subtree, NULL, (xmlChar*)"targetLanguage", trgLang);
    xmlSetNs(subsubtree, NULL);
    xmlSetNsProp(subsubtree, xsi, (xmlChar*)"type", (xmlChar*)"xsd:string");
    return doc;
}
```

Source: K. Banke: "Web Services auf dem PDA von Agenda", Linux Enterprise, 12/2001

What Do You Need Web Services For?

Success Criteria For Mobile Web Services

- **Mobile Web Services should use the context of the user**
 - In the interaction of several services, context information should be transmitted and correlated
- **Infrastructure services should be available globally**
 - Like the EJBs, Web Services need base services for security, persistence and transactions
- **Web Services should model the workflow of the business**
 - Mobile Web Services must use the mobility
 - Do not just offer another access method!
- **Existing systems must be integrated**
 - Locally and for backend computing, standard middleware (J2EE, CORBA 3, COM+) is much more efficient
 - Connectors ought to be created automatically

Success Criteria For Mobile Web Services (2)

- **Mobile Web Services must gradually leave the client/server model behind**
 - Web services can also be used for peer-to-peer (P2P) solutions
 - They enable the devices to synchronize their data automatically
- **Availability and Maintainability must be guaranteed**
 - Applications that use web services depend on their permanent availability
 - New level of mutual dependency (Service level agreements required!)
 - Service providers must take care of failure and load balancing, maybe also of quality of service (like in streaming apps)
 - Versioning must provide downward compatibility
- **Development has to be simple, not more difficult than for the server**
 - So we need flexible development environments (independent of product families) with generators and debugging capabilities

Challenge: Context awareness

- **Ability of service to adapt to a (changing) context automatically**
- **Definition (after Dey and Abowd):**
 - “Context is a piece of information that describes the situation of an entity. The entire set of configuration attributes of this entity is called its context.
 - A system is called ‘context aware’ if it uses the context to provide the user with relevant information and/or services where the relevance depends on the role of the user.”

Relationship Of The Context

- **Context may relate to many properties**
 - Location: absolute or relative to another service/device
 - Bandwidth/network: e.g. switch between GSM/GPRS, UMTS and WLAN
 - User paradigm: desktop vs. mobile phone
 - Unknown target platforms
 - Role of the user: customer, reporter, project lead, father etc.
 - Preferences/interests of the user

Challenge: Session Management

- **Sessions may be interrupted**
 - User leaves area of the network (and enters perhaps area of another one)
 - Device is turned off
 - By user: phone call, meeting, event in his surroundings
 - Due to shortage of energy
 - User switches device
 - Cell phone ⇔ PDA ⇔ Desktop PC
- **So the server must actively manage the sessions**



Application Scenarios

- **Route planning**
 - Planning software can request weather and traffic information as web services and include it in the planning
- **Hospital, Emergency**
 - Doctor can access patient data from everywhere
- **Mobile maintenance of industrial plants**
 - Total status of a machine, latest calibration data etc. can be accessed by the service technician
- **Insurance agent visiting customers**
 - Offers can be calculated based on current discount rates, latest account balance etc.
- **Games**
 - Interaction with other users
- **Spontaneous P2P communication**
 - Exchange of addresses, preferences etc., e.g. at trade fairs



Summary



Conclusion

- **Web services are well-suited even for mobile devices**
 - They bring up new challenges with respect to
 - Network connectivity
 - Context awareness
 - Session management
- **The current toolkits (PocketSOAP, kSOAP) support web services only rudimentary**
 - Unsufficient WSDL integration
 - No proxy generation
- **SOAP server on mobile devices only in a very basic form**
 - Currently not supported by programming environments
 - But: large potential for enterprise connectivity and P2P



Links

- **Microsoft**

- Web Services: <http://msdn.microsoft.com/webservices>
- Mobile Web Forms: <http://msdn.microsoft.com/vstudio/nextgen/technology/mobilewebforms.asp>
- PocketSOAP: <http://www.microsoft.com/mobile/developer/technicalarticles/pocketsoap.asp>
- Smart Device Extensions: <http://msdn.microsoft.com/vstudio/nextgen/device/smartdev.asp>

- **PocketSOAP**

- <http://www.pocketsoap.com>

- **kSOAP**

- <http://ksoap.enhydra.org>

- **Phone Emulator**

- http://www.openwave.com/products/developer_products/sdk/



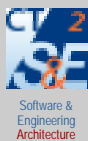
Links (2)

- **Agenda Handheld**

- <http://www.agendacomputing.com>
- Linux VR: <http://www.linux-vr.org>
- Fast Light Toolkit: <http://www.fltk.org>
- XMLLIB: <http://www.xmlsoft.org>

- **Siemens CT**

- <http://www.ct.siemens.de>
- <http://www.plug-n-play-technologies.com>



Links (3)

- **“Towards a better understanding of context and context-awareness”**
 - Anind Dey and G.D. Abowd. Techn. Report, GeorgiaTech, 1998.
- **“Context-Aware Computing Applications”**
 - Bill N. Schilit, Norman I. Adams, and Roy Want. In Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994. IEEE Computer Society.
<ftp://ftp.parc.xerox.com/pub/schilit/wmc-94-schilit.ps>

Thank You

Questions or remarks?



Download of the slides: <http://www.drwieland.de>