Microsoft .NET and XML Web Services

Kimmo Bergius
Product Marketing Manager
Microsoft Oy
kimmob@microsoft.com
Agenda

- Microsoft .NET
  - The what, why, and how?
- XML in .NET
- Web Services in .NET
- Wrap-up

- Slides available at Microsoft Finland presentation archive ([http://www.microsoft.fi](http://www.microsoft.fi))
- For questions kimmob@microsoft.com

Technology requirements

- Computing power
  - Doubles in 18 months
  - Scalability of Intel/PC Servers
- Everything is networked
  - Cheap and available Internet
  - Wireless
- Multitude of devices
  - PDA, mobile phone, PC, laptop, TV, toaster, gas pump...
- Internet Standards
  - XML, ...
What is .NET?

- .NET is Microsoft’s new architecture for developing applications on top of Windows and the Microsoft product line
- .NET is the new implementation of Windows DNA

.NET – Why?

- Technology requirements
  - Integration
  - New application types
  - New user interfaces
- Developer requirements
  - RAD – ease of app development
  - Support for many languages...
  - ...but – shared features and the same deve environment
- Windows DNA
  - Has been developed furing a long time
  - Lots of features == Complex?
Microsoft .NET

- New application development environment
  - Windows Apps (PC/other devices)
  - Web/Internet/intranet
  - Web Services
- Standards in a central role
  - XML etc.
  - Web Services
- Enhance developer productivity
  - Choose any language
  - Shared libraries
  - Shared dev environment
    - Language neutral
    - All development in the same
      - language
      - class library
      - dev tool

.NET Framework

- Web Service
- ASP.NET
- ADO.NET: storage and XML
- Base Class Library
- Common Language Runtime
- MSMQ (Message Queuing)
- COM+ (Transactions, Partitions, Object Pooling)
- IIS
- WMI
- Win32

Shared class libraries
- Same programming model from one language to the other
- Integration between languages
- Extensibility
- Tools

Common Language Runtime
- Basic functionality: program compilation and execution, memory management, security
Execution model

Support for many languages

- .NET is language neutral
  - All .NET languages are equal
  - Integration between languages
  - One dev environment

- Microsoft languages
  - Visual Basic®, C++, C#, J#, JScript®

- Third parties
  - APL, COBOL, Eiffel, Fortran, Haskell, ML, Perl, Python, RPG, Scheme, Smalltalk, …
.NET Framework class library

- Used in all .NET languages
  - Even though the class library is mostly written in C#...
  - ...it can be used in any .NET language
- Shared between dev environments
- Arranged in a single namespace
  - (Almost) all under System.
- Contains:
  - ASP.NET
  - ADO.NET
  - Windows Forms
  - etc...

User Interfaces

- Four different .NET UI’s
  1. ”Traditional” Windows App
  2. ”Mobile” Windows App
  3. ”Traditional” Web UI
  4. ”Mobile” Web UI
- Others
  - Functionality through Web Services...
  - ...and the UI can be almost anything and anywhere, on any device!
New User Interfaces

/.NET Framework

ADO.NET
- Fastest way to use SQL Data
- Support for XML
- Managed Providers

System.Xml
- XSLT 1.0, XPath 1.0, XSD 1.0, DOM2
- Extensible

XSLT/XPath
- Relational data transformation
- W3C compliant
.NET Supports XML!

- XML 1.0
  - http://www.w3.org/TR/1998/REC-xml-19980210
- XML Namespaces
  - http://www.w3.org/TR/1999/REC-xml-names-19990114/
- XML Schemas
  - http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/
  - http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/
- XPath expressions
  - http://www.w3.org/TR/1999/REC-xpath-19991116
- XSL/T transformations
  - http://www.w3.org/TR/1999/REC-xslt-19991116
- DOM Level 1 and Level 2 Core
  - http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/
- SOAP 1.1

XML and data

Diagram showing relationships between controls, designers, code generation, DataSets, XmlDataDocuments, XmlReaders, Command, Connection, DataAdapters, DataReaders, XSL/T, X-Path, etc.
XML and .NET

- XML used heavily in .NET
  - Object serialization format
    - for example DataSet
  - .NET specific, easier API on top of traditional XML API
    - No support for SAX (MSXML 4.0)
    - XML DOM (MSXML.DLL) can be used
  - Remoting/RPC uses XML
    - SOAP and XML Web Services

Web Service

- Component call over HTTP
- WebApplication = application used by a user
- WebService = application used by another application
- Call to a function/method is packeted as an XML message that is sent using SOAP over HTTP or SMTP. Data returned packeted and sent the same way
- Uses SOAP (Simple Object Access Protocol), which is language, OS or architecture neutral
Web Service - Protocols

- Universal Description Discovery and Integration (UDDI)
- Simple Object Access Protocol (SOAP)
- Web Service Description Language (WSDL)
- eXtensible Markup Language (XML)
- Tietojen välityskanava: Internet

Web Service 1

- Web Service (.ASMX files in ASP .NET)
  - Code for services methods
  - .NET wizard creates the project, programmer adds the code
  - Two files are created - codebehind

```
namespace WebService1 {
        public Service1() { InitializeComponent(); }
        private IContainer components = null;
        private void InitializeComponent() {
            if (disposing && components != null)
                components.Dispose();
        }
        protected override void Dispose(bool disposing) {
            if (disposing && components != null)
                components.Dispose();
            base.Dispose(disposing);
        }
        [WebMethod]
        public string GetId(string strKey, string strType) {
            return strKey.ToUpper() + strType.ToUpper();
        }
    }
}
```
Web Service 2

- Web Service Description Language (WSDL)
  - Description of the service
  - How to call the service
  - Methods, parameters and return values
  - XML
  - .NET environment creates automatically
  - http://wmli003600/WebService1/Service1.asmx?wsdl

Web Service 3

- Discovery Documents (DISCO)
  - Publishes Web Services
  - Can be called directly (.vsdisco)
  - Describes the services in a server

http://wmli003600/focus/WebService1/WebService1.vsdisco

<?xml version="1.0" encoding="utf-8" ?>
<discovery xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns="http://schemas.xmlsoap.org/disco/">
  <contractRef ref="http://wmli003600/WebService1/Service1.asmx?wsdl"
    docRef="http://wmli003600/WebService1/Service1.asmx"
    xmlns="http://schemas.xmlsoap.org/disco/scl"/>
</discovery>
Web Service 3

- Universal Description, Discovery and Integration
- UDDI describes available services
- Yellow pages for Web Services
- Can be used externally (uddi.microsoft.com, uddi.ibm.com)
- Can also be used internally as a local directory for web services

Calling Web Services

- Basic call: HTTP Get and HTTP Post
  - Basic types (int, string, jne)
  - Basic type tables
  - Can be called in a “traditional” way
  - http://wmli003600/WebService1/Service1.asmx/GetId?strKey=s1&strType=as
- Default: Simple Object Access Protocol (SOAP)
  - Structures
  - Enumerations
  - Classes
  - XML DataSets
  - Arrays
  - Can contain all kinds of structures
Web Service

- Web Service is a functional component or program, that offers its functionality through the Web Service interface
- Web Services are used through the network (Internet/intranet)
- Data is transferred through XML
- Uses standard protocols, such as SOAP, HTTP, etc.
- Main goal is integration between systems
  - Without knowing the architecture behind a system

Where can Web Services be used

- Integration scenarios
  - Service to a user – combining services (B2C)
  - Services between organisations (B2B)
  - Integration of internal systems (EAI, A2A)

- Write once – use from anywhere
Cool – are we done?

- Nope, Web services will evolve
- Lots of things coming
  - Security
  - Routing & Referral
  - Attachments
  - Transactions
  - Reliable messaging

Web Services futures

XML
Web Services
Integrate systems

Implementation
Products, solutions

Compatibility
Industry support

Infrastructure
Protocol development
Modular Example

```xml
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <S:Header>
    <m:path xmlns:m="http://schemas.xmlsoap.org/soap/觇"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <m:action>http://tickers-us.org/getQuote</m:action>
      <m:to>soap://tickers-us.org/stocks</m:to>
      <m:from>mailto:johnsmith@isp</m:from>
      <m:id>uuid:84b9f5d0-33fb-4a81-b02b-5b760641c1d6</m:id>
    </m:path>
      <wssec:BinarySecurityToken
        ValueType="wssec:X509v3"
        EncodingType="wssec:Base64Binary">
        dWJzY3JpYmVyLVBlc…… ..eFw0wMTEwMTAwMD...
      </wssec:BinarySecurityToken>
    </wssec:Security>
  </S:Header>
  <S:Body>
    <app:TrafficStatus xmlns:app="http://highwaymon.org/payloads">
      <road>520W</road><speed>3MPH</speed>
    </app:TrafficStatus>
  </S:Body>
</S:Envelope>
```

.NET and standards

- **C# ja “CLI”: ECMA**
  - Submitted 2000
  - Standard approved 2001
  - ISO standard about to be approved
  - Used in many projects, other platforms

- **XML, SOAP, WSDL: W3C, IETF, OASIS**

- **Additional development: WS-I, standardization bodies**

- **Partners in development: HP, Intel, IBM, Ariba...**
Existing code

- Old Windows DNA applications
  - Can be called from .NET
  - .NET components can be called from DNA
- Third party applications
  - Ready-made Interfaces
  - Database connections
  - XML:
    - Data transfer, for example BizTalk
    - Web Services

.NET - tools

- Visual Studio .NET
  - One programming environment
    - All applications
    - All languages
  - Extensible
    - Visual Studio is the basic tool
    - Additional features from partners, such as Rational, Compuware
    - New languages, testing, dev project management
- MSDN Universal
Source Code

- **CE.NET**
  - Download (http://www.microsoft.com/embedded)
  - Platform Builder for custom versions
  - OS Source code
- **Shared Source CLI beta**
  - CLI – CLR, basic class library, C# compiler
  - FreeBSD/Windows NT
  - Download (http://msdn.microsoft.com)
- **Windows NT/2000**
  - Source code for research use
  - Big customers can also obtain source code
  - Requests => Email (kimmob@microsoft.com)

.NET - Benefits

- Many application types
  - User Interfaces
  - Traditional applications
  - Web Services
- Shared programming model
  - All application types
- Support for multiple languages
  - Investments in training
- Easier programming
- Protect old investments – integration to existing code
- Integration to other worlds and architectures
- Performance
- Security
- Newest technology!
How to go forward?

- Web
  - [http://www.microsoft.com/net](http://www.microsoft.com/net)
  - [http://msdn.microsoft.com](http://msdn.microsoft.com)
- Developer products
  - Visual Studio .NET
  - Microsoft Developer Network – MSDN
- Programs
  - Developer programs – MSDN
  - Partner programs– Microsoft Certified Solution Provider program
- kimmob@microsoft.com

Kysymyksiä ja vastauksia?