Web Services

Brian A. LaMacchia
Microsoft
Five Questions

- What is a Web Service?
- Why are Web Services interesting?
  - Why should I care about them?
- What e-commerce business models do Web Services enable?
- What security and privacy issues need to be addressed for Web Services to be successful?
- What the heck is Microsoft’s .NET Platform all about, and how does .NET relate to Web Services?
What is a Web Service?

- Software Design Principles
  - Abstraction
  - Componentization
    - In your own programs
    - Reusable software components
- Current web usage
  - User-oriented browsing
  - User-oriented data publication
Software Design Principles

- Abstraction
  - Procedural abstraction

```java
public static int Square(int x) {
    return x * x;
}
```

```java
int y = Square(3); // y is 9
```

- We abstract & reuse useful functions all the time in programs
- Abstraction hides implementation details
Abstraction Hides Details

public static float GetQuote(String symbol) {
    // implementation goes here
    // details are hidden from caller
}

public static void Main(String[] args) {
    float msftPrice = GetQuote("MSFT");
    Console.WriteLine("MSFT: {0:F2}", msftPrice);
}

C:\>test.exe
MSFT: 61.40

- Only need to worry about inputs to & outputs from a method or function
Componentization

- We share code among programs by creating software components
- Ex: Software libraries that you link against when you compile programs, or that you reference dynamically

```perl
#!/usr/pkg/bin/perl
use Finance::YahooQuote;

@symbols = ('msft', 'intc', 'dell', 'hwp', 'cptb');
@q = getquote(@symbols);
foreach $a (@q) {
    print $$a[0]." ".$$a[2]." ".$$a[5]."\n";
}
```
Software Components

- Reusable components are valuable
  - Save time
    - Coding
    - Debugging
  - Save testing effort
  - Share knowledge
- You can sell components
  - Markets exist for software libraries
- You can given them away to sell something else
  - Ex: Device drivers
Components are “local”

- **Local code execution**
  - Execute on your machine
  - Code (source or object) must “live” on your local machine
    - Maybe do just-in-time download/install

- **Not so great if the data is remote**
  - Ex: Stock analysis component
    - Wants as much historical data as possible
    - Is everyone going to cache the last 100 years of the NYSE on their hard disks?
  - Ex: complex searches against the NYT archives
    - Want code to run on the server
U.S. Sets Up Plan to Fight Smallpox in Case of Attack
By LAWRENCE K. ALTMAN
Officials at the Centers for Disease Control and Prevention are training doctors to recognize the disease and vaccinating small teams of experts.

- A Muscular Lobby Tries to Shape Nation's Bioterror Plan

THE PLOT
Hijackers' Meticulous Strategy of
Web Usage Today

- Web usage today is browser-oriented
  - Users browse for information
  - Vast databases are accessed through HTML gateways & user-friendly displays
- Example: Yahoo’s stock ticker
  - Yahoo has tons of stock price history sitting behind http://quote.yahoo.com/
  - That info is easily available to anyone who browses to the right page
  - But look at how the info is presented...
Welcome, bal6765

Quotes

MICROSOFT CP (NasdaqNM:MSFT)

<table>
<thead>
<tr>
<th>Last Trade</th>
<th>Change</th>
<th>Prev Close</th>
<th>Volume</th>
<th>Div Date</th>
<th>Ex-Div Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 2 - 61.40</td>
<td>-0.44 (-0.71%)</td>
<td>61.84</td>
<td>41,684,400</td>
<td>Mar 26, 1999</td>
<td>Mar 29, 1999</td>
</tr>
<tr>
<td>Day’s Range</td>
<td>Bid</td>
<td>Ask</td>
<td>Avg Vol</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>60.51 - 63.021</td>
<td>61.51</td>
<td>61.55</td>
<td>34,492,136</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52-week Range</td>
<td>Earn/Shr</td>
<td>P/E</td>
<td>Mkt Cap</td>
<td>Div/Shr</td>
<td>Yield</td>
</tr>
<tr>
<td>40.2500 - 76.1500</td>
<td>1.15</td>
<td>53.54</td>
<td>330.7B</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Add to My Portfolio - Set Alert
“Screen-scraping”

- Data is formatted for easy use by people, not programs
- Programs need to mimic users to...
  - Parse/make use of the data
  - Call server-side functions (plot a chart, compute something, perform a search, etc.)
- “Scrape the screen”
  - Download the HTML and then pattern-match against it
  - Inefficient and fragile
  - Data type semantics are lost
- We need a better mechanism for making Web-published data and component software functions available to programs
What is a Web Service?

- Software components (application logic) accessible via standard Web protocols
  - “Programming the Web”
  - Better: “remote procedure calls over the Web”
- Web sites with no user interface
- Available to any client that speaks the necessary Web protocols (XML, SOAP)
- Platform independent components
- Enable highly distributed systems
Finding & talking to Web Services

- Clients need answers to three separate questions:
  - What services are available?
  - How do I communicate with this particular service?
  - Let’s talk! (Give me some data...)
Three Standards

- UDDI (Universal Description Discovery and Integration)
  - Yellow pages directory for services

- WSDL (Web Service Description Language)
  - Document describing the message exchange contract

- SOAP (Simple Object Access Protocol)
  - XML-based protocol for messaging

- All based on XML (the foundation)
Web Services (In Practice)

Find a Service
- http://www.uddi.org
- Link to WSDL document

How do we talk? (WSDL)
- http://yourservice.com/?WSDL
- XML with service descriptions

Let me talk to you (SOAP)
- http://yourservice.com/svc1
- XML/SOAP BODY

Web Service Consumer
- UDDI

Design-Time or Dynamic
- Runtime
Why are Web Services interesting?
For Developers…

- Access to a “Web-wide library of software components”
- Smart development tools can…
  - Help you locate useful Web Services
  - Download service descriptions (WSDL)
  - Automatically generate code from the WSDL to talk to the service using SOAP over HTTP
  - On the server side, automatically generate WSDL for a service from its source code
WSDL for a Quote Service

```xml
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdls/soap/
 xmlns:tns="http://www.themindelectric.com/wsdlnet.xmethods.services.stockquote.StockQuote/
 xmlns:s="http://www.w3.org/2001/XMLSchema"
 xmlns:tm="http://microsoft.com/wsdl/mime/textMatching"
 xmlns:mime="http://schemas.xmlsoap.org/wsdls/mime/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 targetNamespace="http://www.themindelectric.com/wsdlnet.xmethods.services.stockquote.StockQuote/
 xmlns="http://schemas.xmlsoap.org/wsdls/">

<types />

<message name="getQuoteResponse1">
  <part name="Result" type="s:float"/>
</message>

<message name="getQuoteRequest1">
  <part name="symbol" type="s:string"/>
</message>

<portType name="net.xmethods.services.stockquote.StockQuotePortType">
  <operation name="getQuote" parameterOrder="symbol">
    <input message="tns:getQuoteRequest1"/>
    <output message="tns:getQuoteResponse1"/>
  </operation>
</portType>

<binding name="net.xmethods.services.stockquote.StockQuoteBinding" type="tns:net.xmethods.services.stockquote.StockQuotePortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
  <operation name="getQuote">
    <input>
      <soap:body use="encoded" namespace="urn:xmethods-delayed-quotes" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:xmethods-delayed-quotes" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
  </operation>
</binding>

<service name="net.xmethods.services.stockquote.StockQuoteService">
  <documentation>net.xmethods.services.stockquote.StockQuote web service</documentation>
  <port name="net.xmethods.services.stockquote.StockQuotePort" binding="tns:net.xmethods.services.stockquote.StockQuoteBinding">
    <soap:address location="http://64.39.29.211:9090/soap"/>
  </port>
</service>
</definitions>
```
/// This source code was auto-generated by wsdl, Version=1.0.3430.0.
///
/// using System.Diagnostics;
/// using System.Xml.Serialization;
/// using System;
/// using System.ComponentModel;
/// using System.Web.Services;

/// <remarks/>
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Web.Services.WebServiceBindingAttribute(Name = "net.xmethods.services.stockquote.StockQuoteBinding",
Namespace = "http://www.themindelectric.com/wsdl/net.xmethods.services.stockquote.StockQuote")]
public class StockQuoteService
{
    /// <remarks/>
    [System.Diagnostics.DebuggerStepThroughAttribute()]
    [System.ComponentModel.DesignerCategoryAttribute("code")]
    [System.Web.Services.WebServiceBindingAttribute(Name = "net.xmethods.services.stockquote.StockQuoteBinding",
    RequestNamespace = "urn:xmethods-delayed-quotes",
    ResponseNamespace = "urn:xmethods-delayed-quotes")]
    [return: System.Xml.Serialization.SoapElementAttribute("Result")]
    public System.Single getQuote(string symbol) {
        object[] results = this.Invoke("getQuote", new object[] {
            symbol});
        return ((System.Single)(results[0]));
    }

    /// <remarks/>
    public System.IAsyncResult BegingetQuote(string symbol, System.AsyncCallback callback, object asyncState) {
        return this.BeginInvoke("getQuote", new object[] {
            symbol, callback, asyncState});
    }

    /// <remarks/>
    public System.Single EndgetQuote(System.IAsyncResult asyncResult) {
        object[] results = this.EndInvoke(asyncResult);
        return ((System.Single)(results[0]));
    }
}
Use the Web Service in your own programs

using System;
using System.IO;

public class Quote {

    public static void Main(String[] args) {
        StockQuoteService service = new StockQuoteService();
        float msftPrice = service.getQuote("MSFT");
        Console.WriteLine(msftPrice);
    }
}
For Businesses…

- Three keys to next generation applications:
  - “Any-to-Any” integration
  - Integral assumption of development
  - Must tie together “islands of data, devices, OS, businesses, people”
- Intelligent devices
  - Many types, with varying capabilities, but all speak common protocols
  - Anytime, anywhere access
  - Access and action
- Open and accessible to all
  - Open, internet based standards
  - Broad accessibility
New Applications

- Shift to decentralized/distributed
- Span multiple clients, servers, services
- Federate across organizations
- Build systems that play in larger solutions

Company A
- Mobile Employees
- Consumers, Partners

Company B
- Mobile Employees
- Consumers, Partners
- Customers, Partners, Suppliers
The same Web Service ...

- “Enabled” an intranet application
- “Embedded” in a mobile/offline solution
- “Published” over the Internet to a partner
What e-commerce business models do Web Services enable?
“How do I make money from Web Services?”

- Every data exchange is potentially a revenue opportunity
  - Both the raw data and the exchange/translation can have value
    - Ex: stock quotes are essentially free, but stock alerts sent to my phone have value
  - Still need someone willing to buy it
- Web Services help in two ways:
  - Increase availability of data
    - “It’s on the web!”
  - Enabled clients = potential customer pool for your data
Next Gen Web Applications

Applications Become Programmable Web Services

Other Services

Smarter Clients

Standard Browsers

Smarter Devices

XML

XML

XML

XML

Biz Logic & Web Service

OS Services

Open Internet Communications Protocols (HTTP, SMTP, XML, SOAP)

Richer, More Productive User Experience

Public Web Services

.NET Services

Internal Services

Servers Data, Hosts

Applications Leverage Globally-Available Federated Web Services
Revenue models

- Short term will likely look similar to current DRM content models
  - Subscriptions
  - Per-copy/per-transaction (depending on overall value of the copy/transaction)
  - Perhaps some metered usage
- Long term might change depending on micro-payments
- New twist: aggregating clients & services
  - There’s money in creating clients that are smart about how they combine data from various services (ex: comparison shoppers)
- Prob. no advertising revenue (no eyeballs!)
What security and privacy issues need to be addressed for Web Services to be successful?
Security & Privacy

- **Protocol-level**
  - Integrity & secrecy of message traffic
  - Authentication

- **Data-level**
  - Integrity & secrecy of collected data
  - Data privacy
    - Collection/sharing of information
Integrity & secrecy of message traffic

- Need robust security protocols for SOAP messages
  - XML Digital Signature standard
  - XML Encryption standard (in process)
  - Need protocol pieces
    - Replay attack defenses, etc.
- This is all do-able, just requires effort leading to an interoperable standard
We need to reliably identify the entity that is making a service request.

What’s the requesting entity?
- Could be “user,” “machine” or “application” depending on context.
- What does authentication mean in each of these contexts?

Once the entity is identified, need to determine what it’s allowed to do.
- Trust management engine.
Protecting stored data

- Today, almost all Web sites use the “Trust us, your data is safe” method of data protection.
- Servers holding aggregated data are prime targets for attack.
- No real incentive for services to deploy real security measures (e.g. PK crypto).
- We must design centralized data stores for per-user encrypted data.
- Key management tools/UI still a problem.
Data Privacy

- Control over collection & distribution of personal information
  - Lorrie Cranor covered this on Tuesday in her talk on P3P
- Essentially this is a DRM-type problem and likely require DRM-like solutions
  - Digital rights management is primarily concerned with distribution of valuable content to “untrusted” users
  - Data privacy is primarily concerned with the distribution of valuable personal data to “untrusted” centralized services.
What the heck is Microsoft’s .NET Platform all about, and how does .NET relate to Web Services?
Three Pillars of .NET

1. XML Web Services

2. New Applications = Clients + Servers + Services

3. Great User Experiences
Microsoft .NET
A platform for distributed Web Services

- Best of breed development tools for building Web Services
  - .NET Framework
  - Visual Studio .NET
- Software for new “smart clients”
  - Native support for Web Services
- .NET Foundation Services
  - Provide basic building blocks to kick-start the industry
Changing Application Architectural Model

Your Application

.NET Framework

Clients

Servers

Services

XML Web Services

Application

Application

Application
.NET Clients
New Breed of Smart Clients

- Windows-powered
- XML, service-aware
- Work well alone or with others
.NET Foundation Services

Open Internet Protocols

Your Sales Data Service
Your Internal Billing Service

Passport – Identity Service
Directory and Search Service
Personalization Service
Software Delivery Service
Calendaring Service
Schematized Storage Service
Notification & Msg Service
Geographic Mapping Service
Greenwich Mean Time Service
Credit Card Statement Service
...

Internal Corporate Services
Microsoft Foundation Services
Web Services Built by 3rd Parties
Summary

- **Web Services architecture**
  - Componentizing web-accessible data
  - Built on XML-based protocols
- **Reasons to move to Web Services**
  - Quickly tie together data islands
  - “Any-to-Any” integration
- **Microsoft .NET Platform**
  - Comprehensive attempt to provide all the parts needed to enable Web Services
    - Development tools, hosting servers, building block services, etc.
Questions?