

# **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**

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

```
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

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

@symbols = ('msft', 'intc', 'dell', 'hwp', 'cpth');
@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 give 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**



# Saved by the Web!

The New York Times on the Web - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address http://www.nytimes.com

NYTimes.com/jobs Click Here

# The New York Times

ON THE WEB

IndyMac Bank. click here now!

UPDATED SATURDAY, NOVEMBER 3, 2001 6:38 PM ET Personalize Your Weather

CLASSIFIEDS

Find a Job Post a Job Real Estate **NEW** Automobiles All Classifieds

NEWS

International National Politics Business Technology Science Health Sports New York Region Education Weather

Search  Past 30 Days

[Go to Advanced Search](#)

Welcome, [bal-nyt](#)

[Sign Up for Newsletters](#) | [Log Out](#)

## 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




(James Hill/NYT)

At a Northern Alliance camp in Afghanistan, rain canceled training for the day. The anti-Taliban rebels have been a

In the Magazine

## Post-Rudy

No mayor of New York has ever assumed a job as difficult as this one has become.



**A NATION CHALLENGED**

[Complete Coverage](#)

A special section on the Sept. 11 attacks and their aftermath, including an archive of past

Internet

# 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...**

# Yahoo's Stock Ticker

Yahoo! Finance - MSFT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://quote.yahoo.com/q?s=MSFT&d=t>

**YAHOO! FINANCE** [Site Map](#) - [Finance Home](#) - [Yahoo!](#) - [Help](#)

[Affordable Wall Street](#)    [symbol lookup](#)  
[Research Reports](#)

Sunday, November 4 2001 3:24am ET - U.S. Markets Closed.

Welcome, bal6765 [Net Worth](#) - [My Yahoo! View](#) - [Customize](#) - [Sign Out](#)


**Services** [Bill Pay](#) - [Funds Transfer](#) - [PayDirect](#) **Accounts** [ add: [bank](#), [credit cards](#) ] **Portfolios** [ [manage](#) - [create](#) - add: [brokerage](#), [401\(k\)](#), [mutual fund](#) ] [standard](#) | [Java Mgr](#), [Real-Time Package](#)

**Quotes** [Customize Finance](#)

Views: [Basic](#) - [DayWatch](#) - [Performance](#) - [Fundamentals](#) - [Real-time Mkt](#) - **[Detailed](#)** - [[Create New View](#)]

**MICROSOFT CP** (NasdaqNM:MSFT)- More Info: [News](#), [Profile](#), [Reports](#), [Research](#), [Insider](#), [Options](#), [Msgs](#) - Trade: [Choose Brokerage](#)

Last Trade Nov 2 · <b>61.40</b>	Change <b>-0.44 (-0.71%)</b>	Prev Cls 61.84	Volume 41,684,400	Div Date Mar 26, 1999
Day's Range 60.51 - 63.021	Bid 61.51	Ask 61.55	Open 61.93	Avg Vol 34,492,136
52-week Range 40.2500 - 76.1500	Earn/Shr 1.15	P/E 53.54	Mkt Cap 330.7B	Div/Shr N/A
				Yield N/A

MSFT 2-Nov-2001 (C) Yahoo!  
  
 Small: [[1d](#) | [5d](#) | [1y](#) | [none](#)]  
 Big: [[1d](#) | [5d](#) | [3m](#) | [6m](#) | [1y](#) | [2y](#) | [5y](#) | [max](#)]

[Add to My Portfolio](#) - [Set Alert](#) [Non-Tables Version](#) - [Download Spreadsheet](#)

Internet

# “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)



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 version="1.0" encoding="utf-8"?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.themindelectric.com/wsdl/net.xmethods.services.stockquote.StockQuote/" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://www.themindelectric.com/wsdl/net.xmethods.services.stockquote.StockQuote/"
  name="net.xmethods.services.stockquote.StockQuote" xmlns="http://schemas.xmlsoap.org/wsdl/">
  <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">
      <soap:operation soapAction="urn:xmethods-delayed-quotes#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>
```

# Auto-gen Code from WSDL

```
//  
// This source code was auto-generated by wsdl, Version=1.0.3430.0.  
//  
using System.Diagnostics;  
using System.Xml.Serialization;  
using System;  
using System.Web.Services.Protocols;  
using System.ComponentModel;  
using System.Web.Services;  
  
/// <remarks/>  
[System.Diagnostics.DebuggerStepThroughAttribute()]  
[System.ComponentModel.DesignerCategoryAttribute("code")]  
[System.Web.Services.WebServiceBindingAttribute(Name="net.xmlmethods.services.stockquote.StockQuoteBinding",  
    Namespace="http://www.theminelectric.com/wsdl/net.xmlmethods.services.stockquote.StockQuote/")]  
public class StockQuoteService : System.Web.Services.Protocols.SoapHttpClientProtocol {  
  
    /// <remarks/>  
    public StockQuoteService() {  
        this.Url = "http://64.39.29.211:9090/soap";  
    }  
  
    /// <remarks/>  
    [System.Web.Services.Protocols.SoapRpcMethodAttribute("urn:xmlmethods-delayed-quotes#getQuote", RequestNamespace="urn:xmlmethods-delayed-quotes",  
        ResponseNamespace="urn:xmlmethods-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);
```

```
    }
```

```
}
```

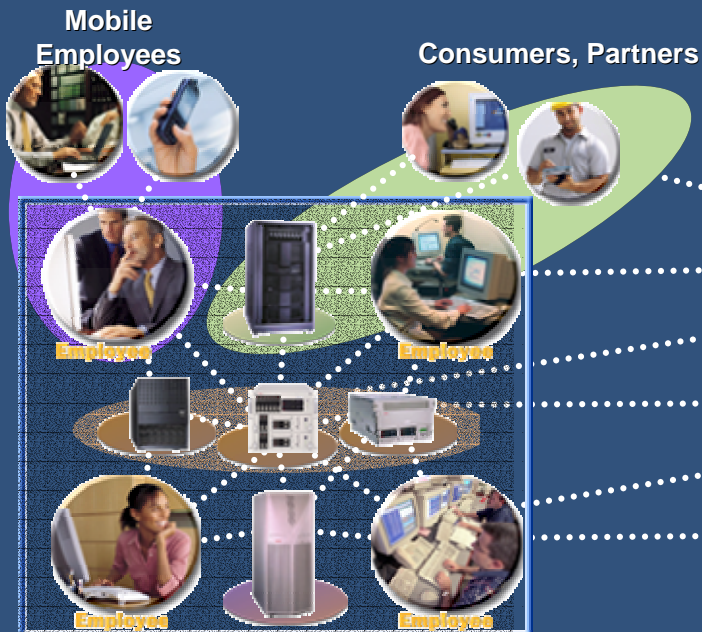
# For Businesses...

- **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**

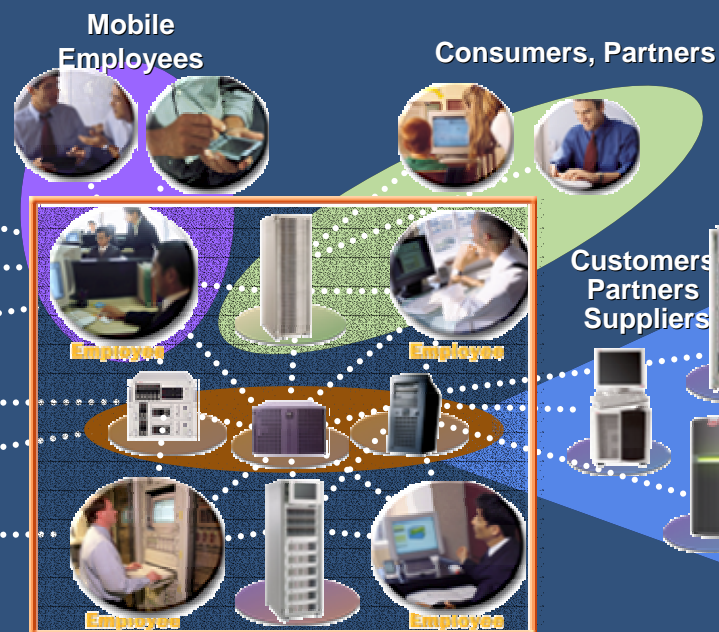
# New Applications

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

## Company A

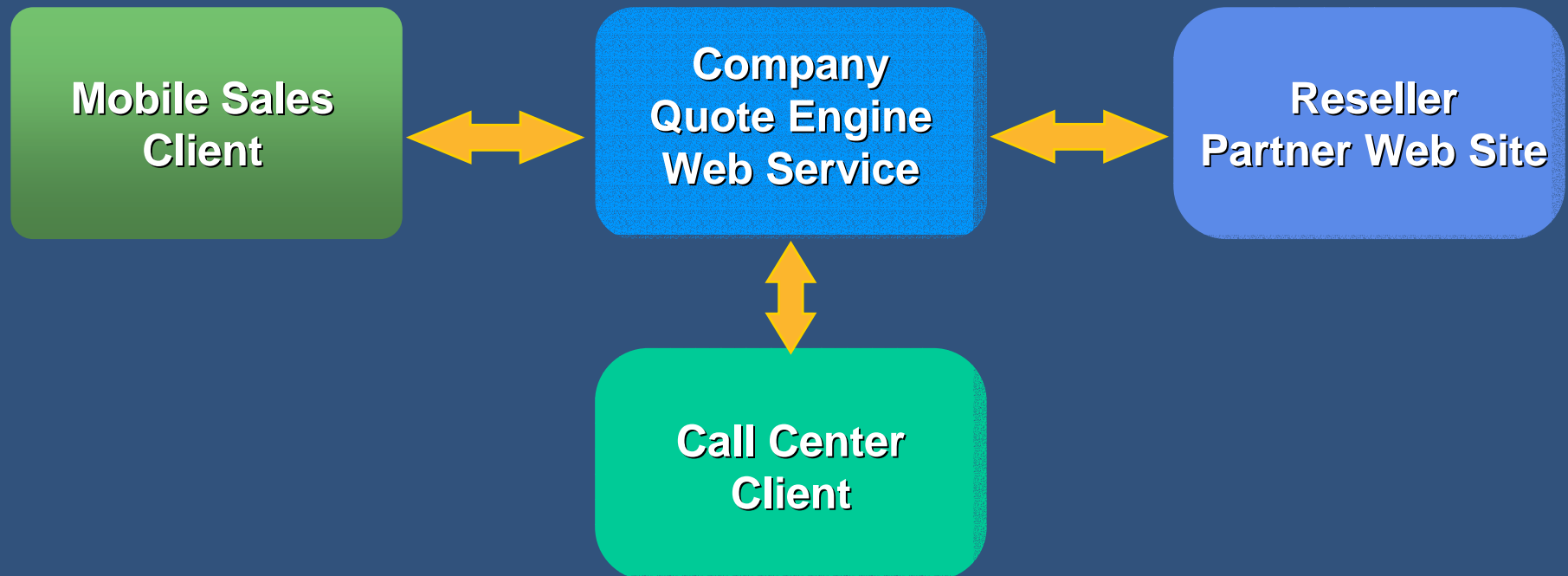


## Company B



# Web Services

## Simple Customer Scenario



- The same Web Service ...
  - “Enabled” an intranet application
  - “Embedded” in a mobile/offline solution
  - “Published” over the Internet to a partner



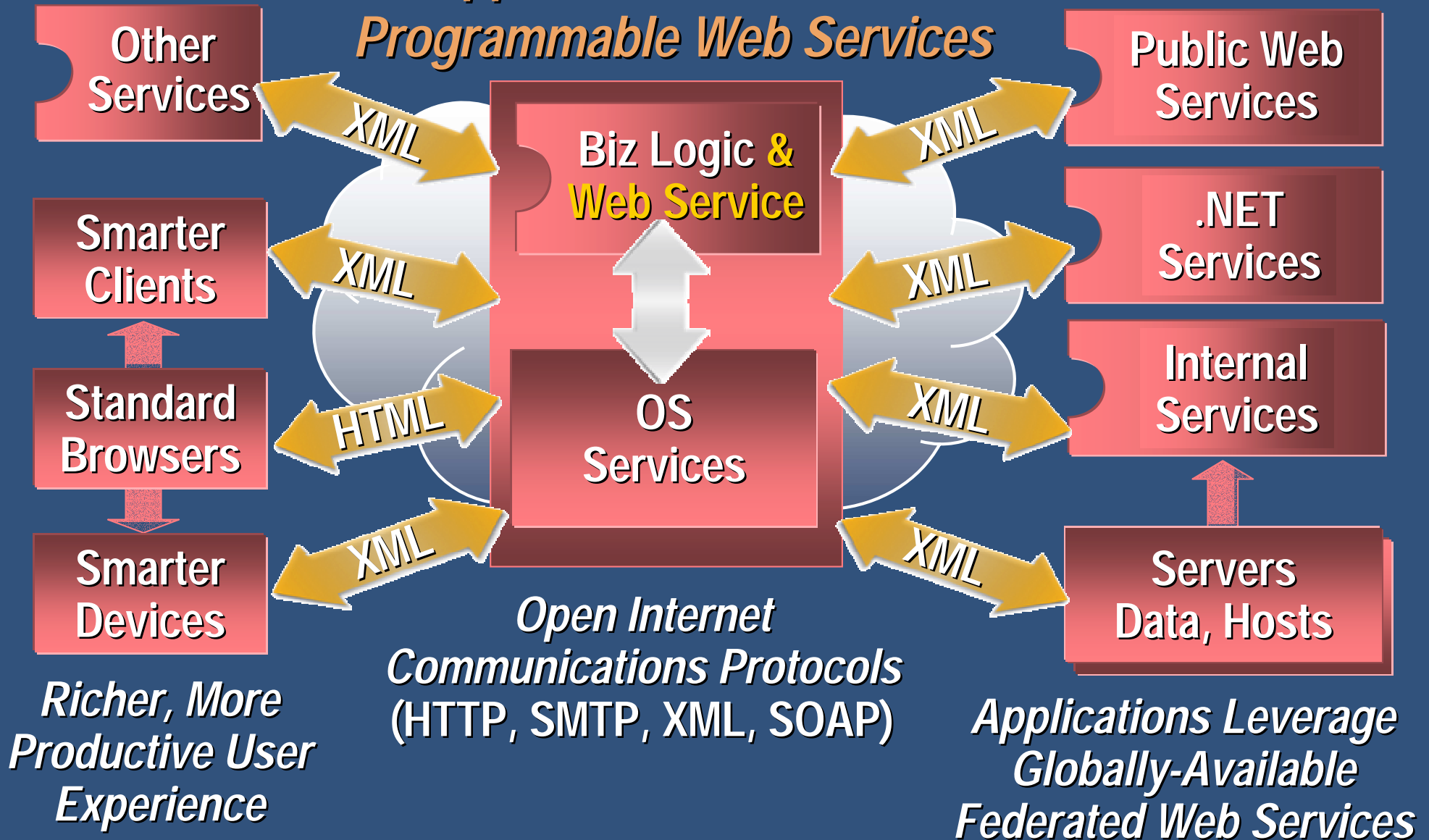
**What e-commerce business models do Web Services enable?**

# “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*



# 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

# Authentication

- 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**

The Microsoft .NET logo is displayed on a white rounded rectangular background. It features the word "Microsoft" in a standard sans-serif font above the ".NET" logo. The ".NET" logo consists of a black dot, followed by the letters "NET" in a stylized, lowercase font. The letters "N", "E", and "T" are filled with a horizontal gradient of colors: blue, green, and yellow respectively.

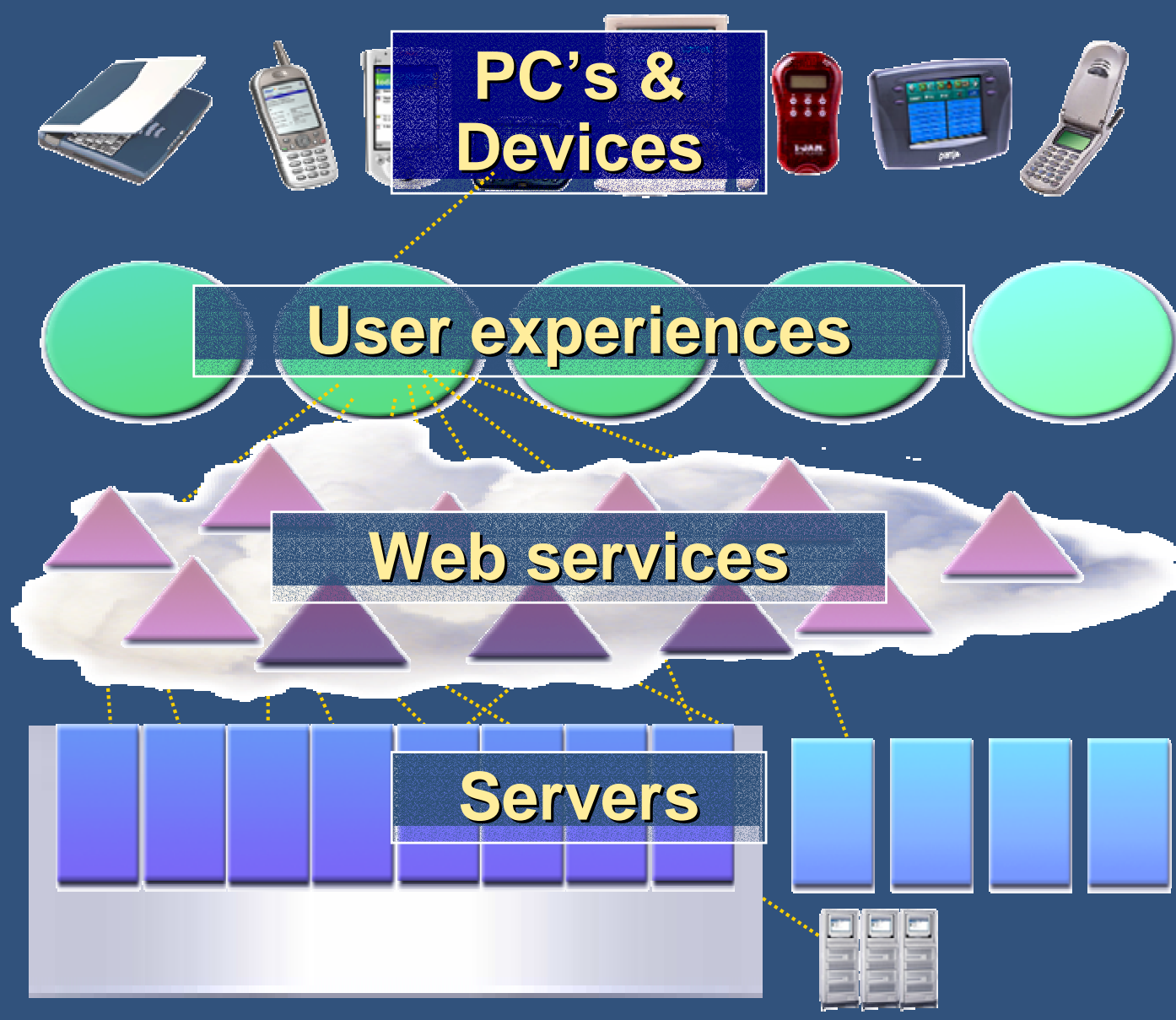
# Microsoft .NET

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



# A Platform For Web Services?



# Microsoft .NET

PC's & Devices



User Experiences

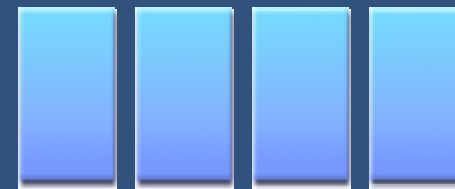


Web Services

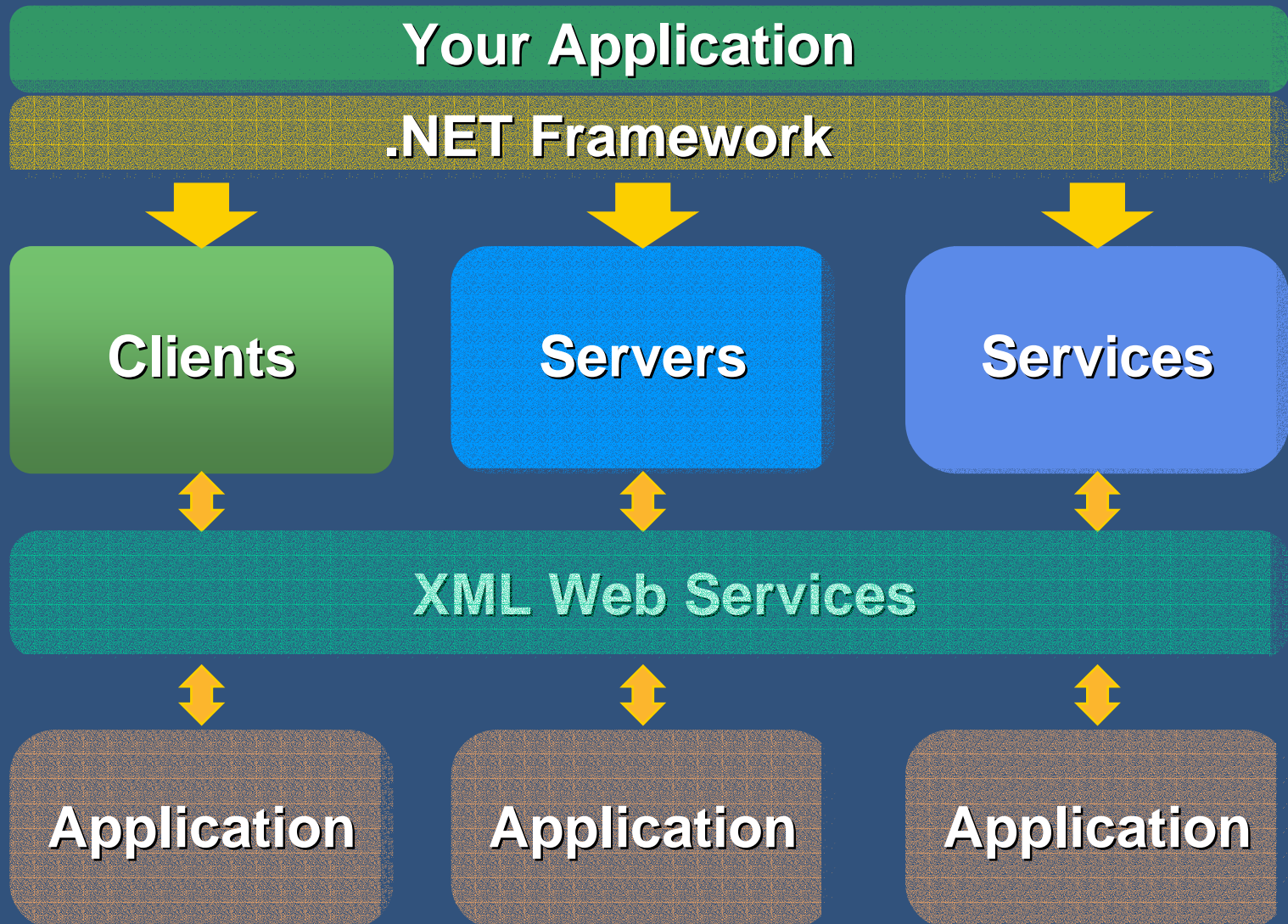


Visual Studio.NET  
.NET Framework

Servers



# Changing Application Architectural Model





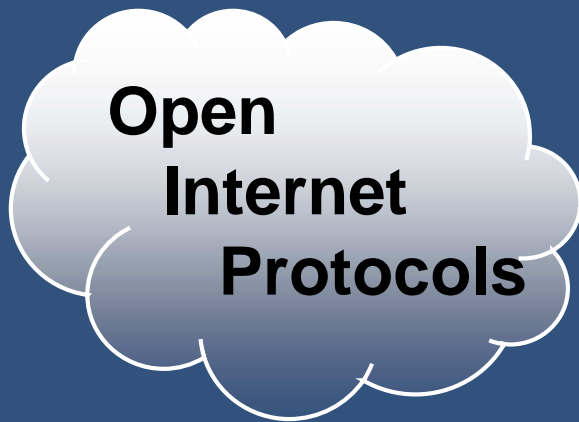
# .NET Clients

## New Breed of Smart Clients



- Windows-powered
- XML, service-aware
- Work well alone or with others

# .NET Foundation Services



Your Sales Data Service

Internal  
Corporate  
Services



Your Internal Billing Service



Passport – Identity Service



Directory and Search Service



Personalization Service

Microsoft  
Foundation  
Services



Software Delivery Service



Calendaring Service



Schematized Storage Service



Notification & Msg Service



Geographic Mapping Service

Web  
Services  
Built by 3<sup>rd</sup>  
Parties



Greenwich Mean Time Service



Credit Card Statement Service



...

# 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?**