

Microsoft®
.net™

The image features the Microsoft .NET logo. The word "Microsoft" is in a black, sans-serif font with a registered trademark symbol (®). Below it, ".net" is written in a large, bold, lowercase sans-serif font. The letters "n", "e", and "t" are filled with a vertical color gradient: "n" is blue, "e" is yellow, and "t" is orange. A small trademark symbol (™) is positioned to the upper right of the "t". The background is a dark blue gradient with glowing, wavy lines and faint binary code (0s and 1s) scattered throughout.

**Any time, any place,
and on any device**

Microsoft .NET

Informationsintegration & Web-Portale

Guido Sautter

Informationssystem für den Alumni-Tag



Any time, any place, and on any device – Microsoft®

Die Welt hat sich verändert

Client/Server
Computing



Enterprise
Centralized Mainframes



Server-Centric
Web

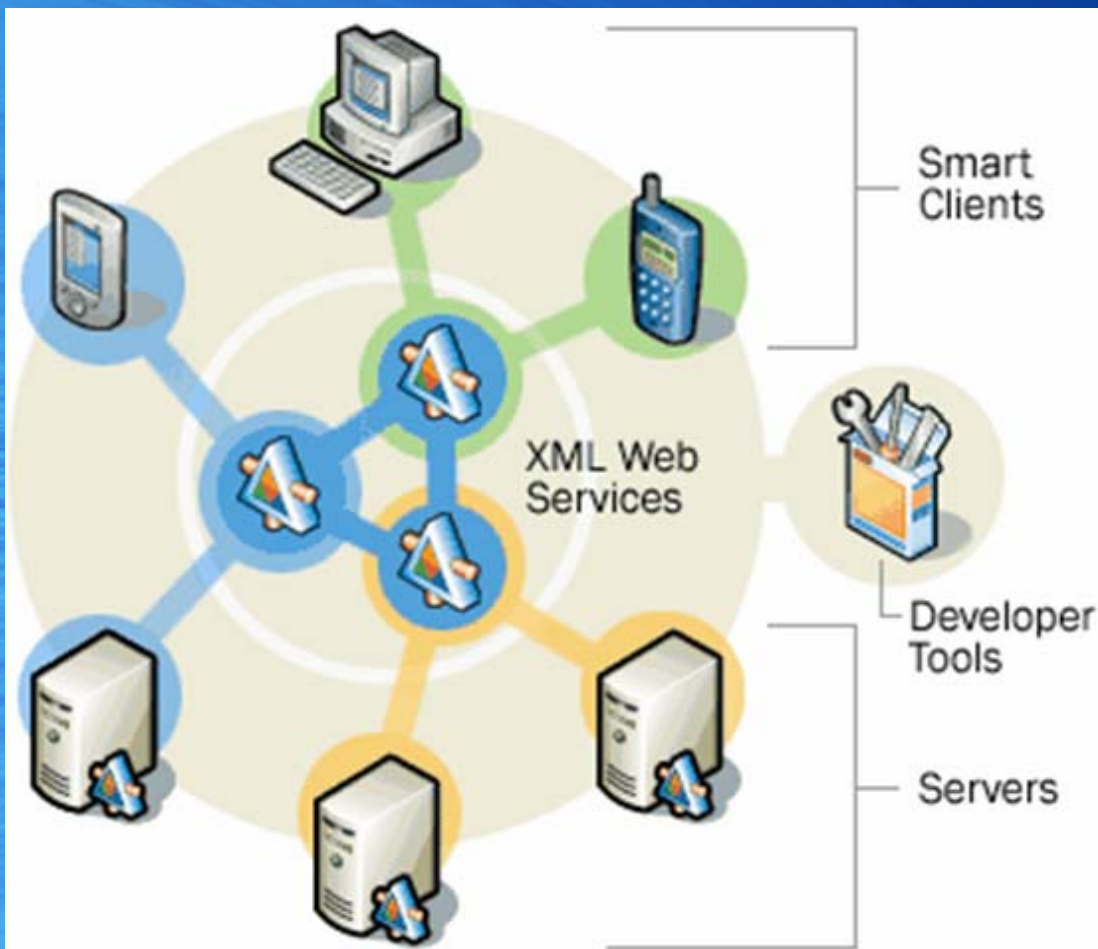


Any time, any place,
and on any device

Auftretende Probleme

- **Installation komponentenbasierter Anwendungen nicht einfach**
 - „DLL-Hell“ (Versionskonflikte, etc)
- **Paradigma der Web-Programmierung**
 - Zustandslose HTML-Seiten (oder Web-Applikationen) schwer zu entwickeln
- **Vielzahl von APIs**
 - zB COM, DCOM, Win32, MFC, etc
 - Viele davon Plattform-abhängig

Das .NET Konzept



XML Web services

are small, reusable applications written in XML, a universal language for data exchange. They allow data to be communicated across the Internet (or internal intranet) between otherwise unconnected sources that are enabled to host or act on them, for example:

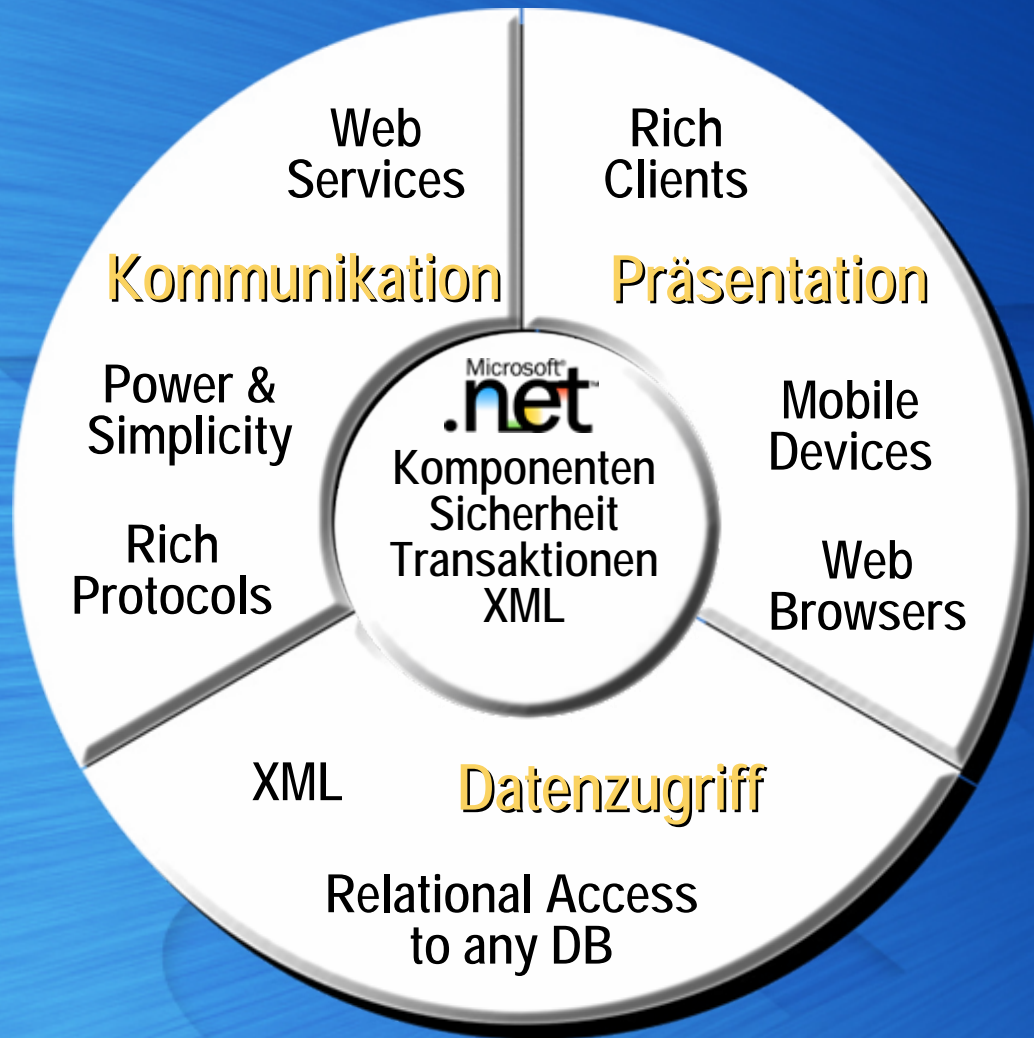
Client-to-client: "Smart" clients or devices can host and apply XML Web services that allow data to be shared anywhere, any time.

Client-to-server: XML Web services can share data from a server application to a desktop or mobile computing device via the Internet.

Server-to-server: XML Web services provide a common interface between existing applications within an environment of independent servers.

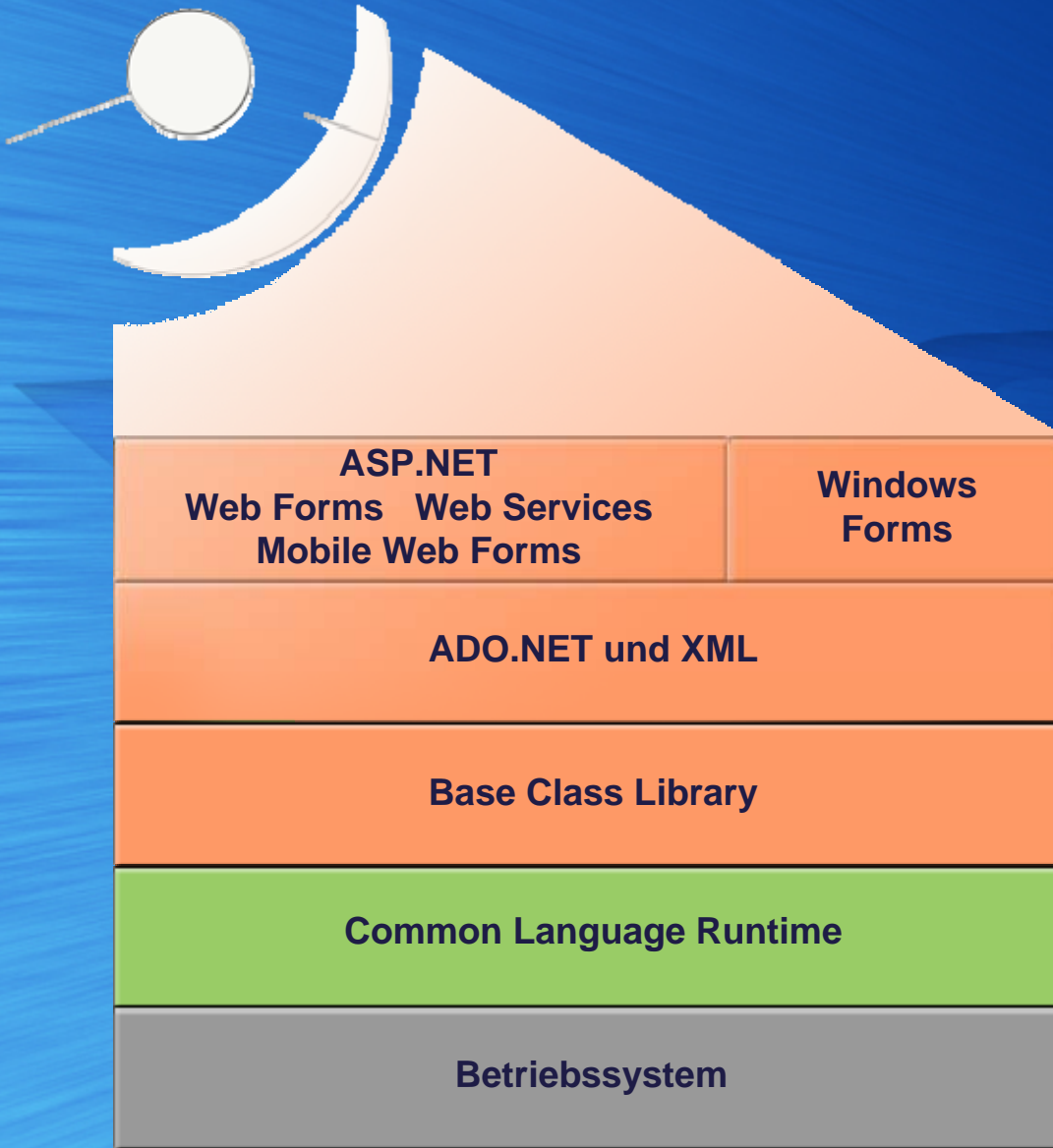
Service-to-service: XML Web services can work together in sequence to create a more complex data operation.

Das .NET Framework



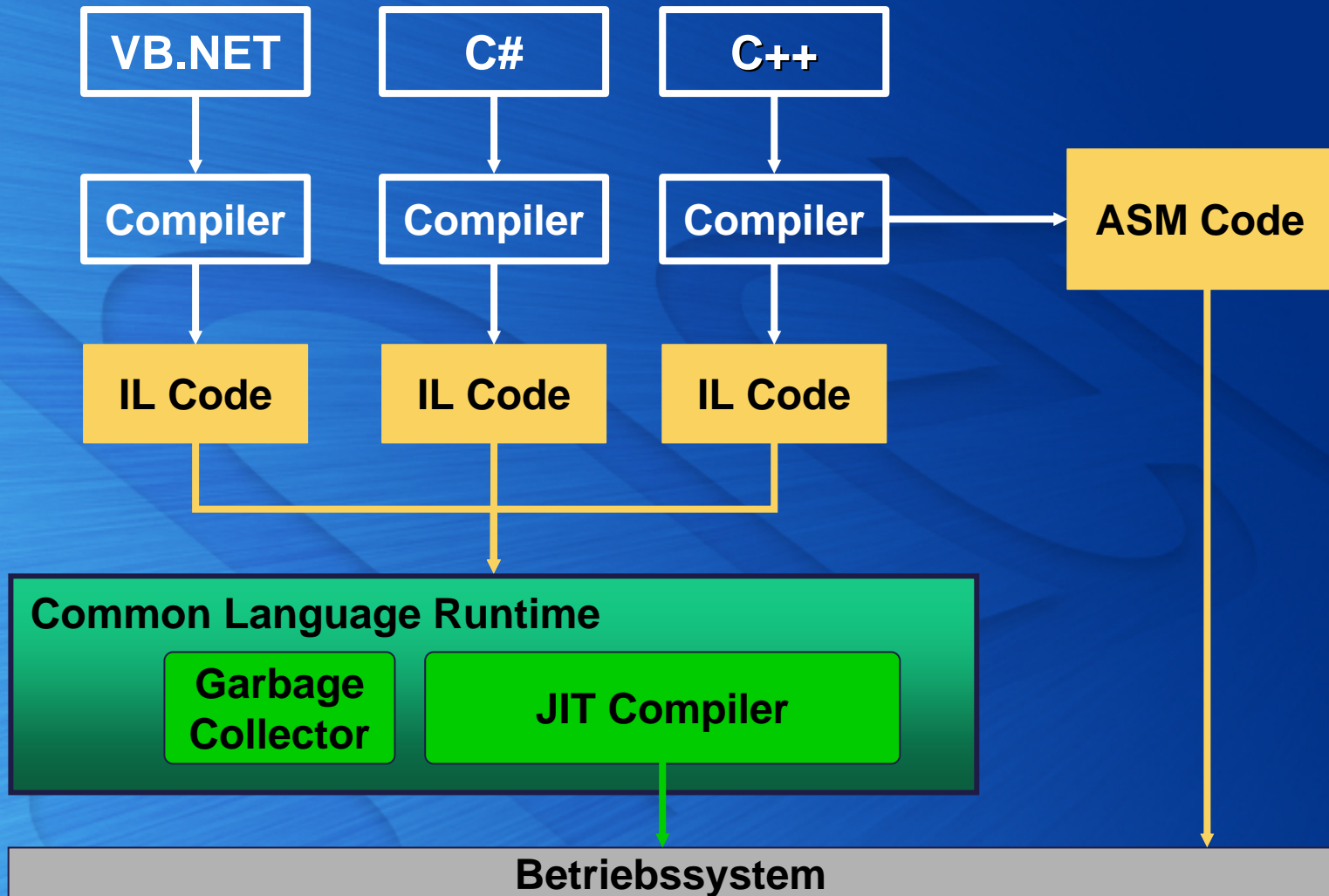
- **Einheitliches Programmiermodell**
- **Quellcode plattformunabhängig (nur CLR nötig)**
- **Für Clients, Server und Services**
- **Interoperabel durch XML Web Services**

Das .NET Framework



- **Gemeinsame Laufzeitumgebung**
 - **Common Language Runtime (CLR)** liefert einheitliches Typsystem (VB.NET, C#, J#,...)
 - **Verbesserte Sicherheit**
 - **Assemblies (Bibliotheken, dll oder exe)**
 - **Remoting (Entfernte Funktionsaufrufe)**
- **Reichhaltige Klassenbibliothek**

Common Language Runtime



Die Welt ist mehrsprachig

20+ Sprachen

Eine IDE



Ein Framework

Ein Debugger

C++ Visual Basic

C# J#

Delphi

Mondrian

Component Pascal

Pascal

Ada

Haskell

Perl

Python

Oberon

RPG

JScript

Scheme

Java

Smalltalk

Eiffel

Oz

Mercury

Cobol

C

Beispiel : Visual Basic .NET

```
Imports System

Module HelloWorld

    Sub Main(ByVal Args() As String)

        Dim I As Integer = 0

        For I = 0 To 10
            Console.WriteLine("Hello, world!")
        Next I

    End Sub

End Module
```

Beispiel : C#

```
using System;

public class HelloWorld
{
    public static void Main(string[] args)
    {
        int i = 0;

        for ( i = 0; i < 10; i++ )
        {
            Console.WriteLine ( "Hello, world!" );
        }
    }
}
```

Exkurs : Java

```
import java.lang.*;

public class HelloWorld
{
    public static void main(string[] args)
    {
        int i = 0;

        for ( i = 0; i < 10; i++ )
        {
            System.writeln ( "Hello, world!" );
        }
    }
}
```

Properties

- Mischung aus Methode und Attribut (smart field, Getter & Setter in einem)

```
public class Button : Control
{
    private string caption;

    public string Caption {
        get {
            return caption;
        }
        set {
            caption = value;
            Repaint();
        }
    }
}
```

```
Button b = new Button();
b.Caption = "OK";
String s = b.Caption;
```

Indexer

- **Arrayzugriff auf Listen-Property**

```
public class ListBox : Control
{
    private string[] items;

    public string this[int index] {
        get {
            return items[index];
        }
        set {
            items[index] = value;
            Repaint();
        }
    }
}
```

```
ListBox listBox = new ListBox();
listBox[0] = "hello";
Console.WriteLine(listBox[0]);
```

„Attribute“

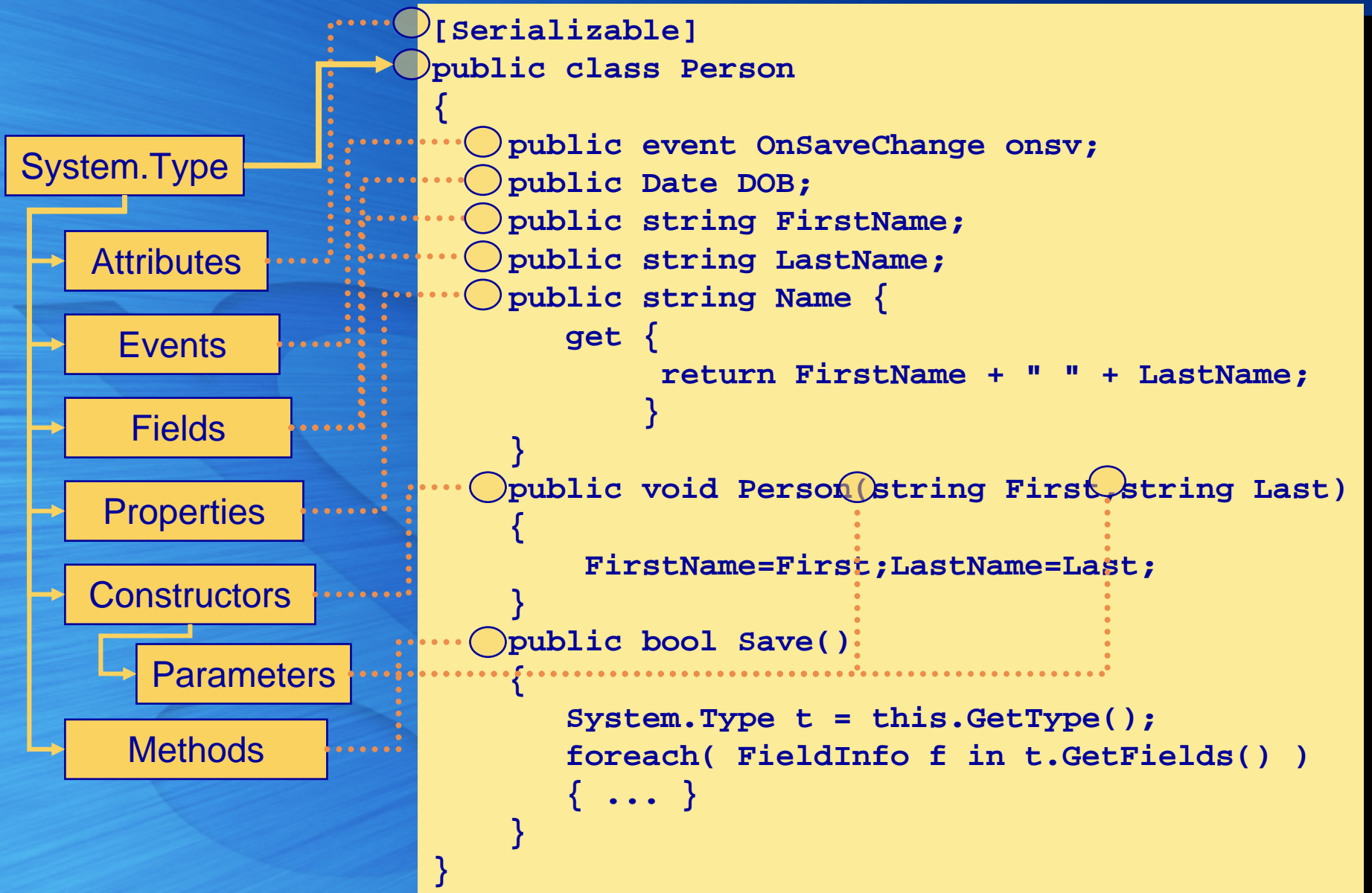
- Benutzerdefinierte Metadaten
- Anwendung:

```
[AuthorIs(„Michael“)]  
class MyClass { ... }
```

- Deklaration:

```
[AttributeUsage(AttributeTargets.All)]  
public class AuthorIsAttribute : Attribute {  
    private string m_Name;  
    public AuthorIsAttribute(string name) {  
        m_Name = name;  
    }  
}
```

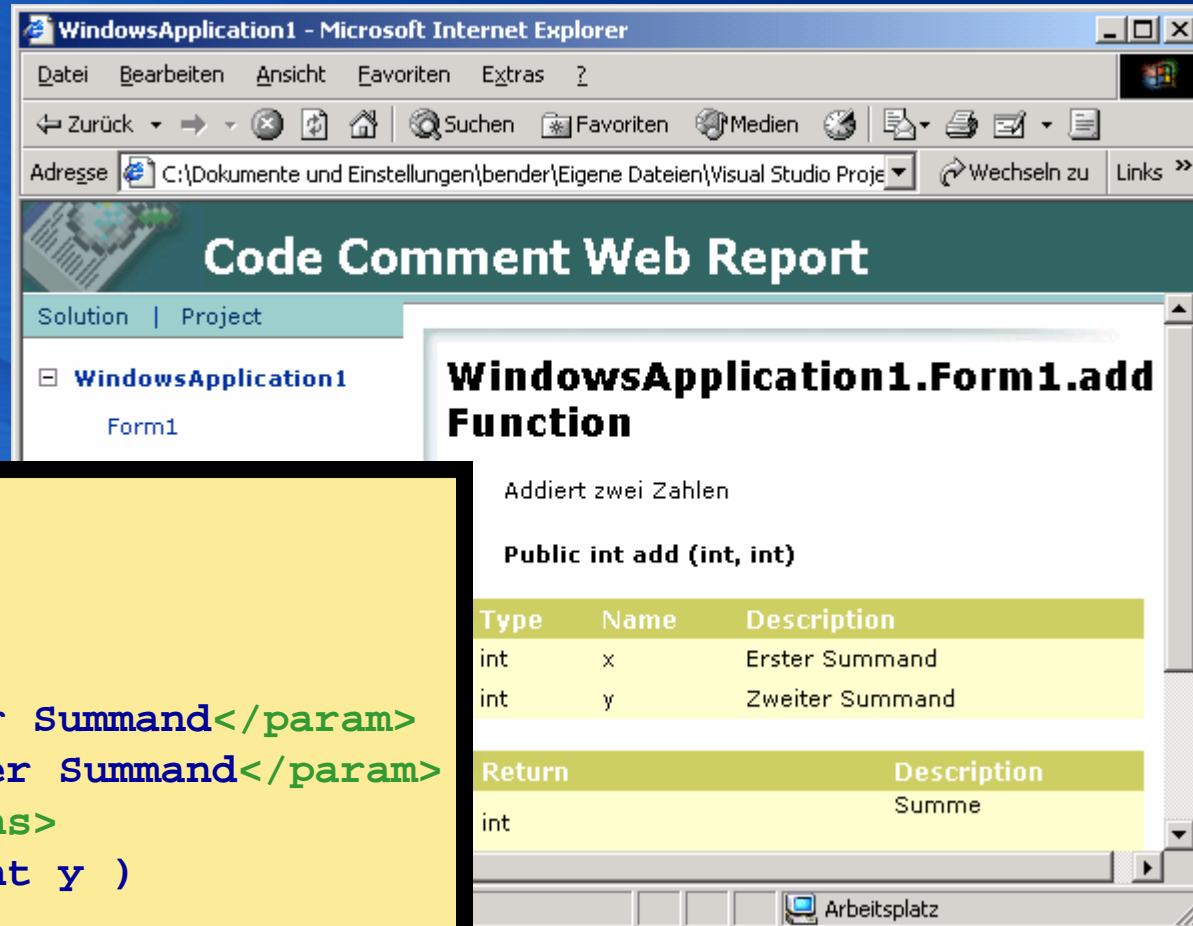
Reflection



Programmdokumentation

- In XML
- Automatisches Erstellen von Webseiten

```
...  
/// <summary>  
/// Addiert zwei Zahlen  
/// </summary>  
/// <param name="x">Erster Summand</param>  
/// <param name="y">Zweiter Summand</param>  
/// <returns>Summe</returns>  
public int add ( int x, int y )  
{  
    return x + y;  
}  
...
```



WindowsApplication1 - Microsoft Internet Explorer

Code Comment Web Report

Solution | Project

WindowsApplication1

Form1

WindowsApplication1.Form1.add Funktion

Addiert zwei Zahlen

Public int add (int, int)

Type	Name	Description
int	x	Erster Summand
int	y	Zweiter Summand

Return	Description
int	Summe

Arbeitsplatz

Die Base Class Library

System.Web

Services

Description

Discovery

Protocols

UI

HtmlControls

WebControls

Caching

Security

Configuration

SessionState

System.Windows.Forms

Design

ComponentModel

System.Drawing

Drawing2D

Printing

Imaging

Text

System.Data

OleDb

SqlClient

Design

SqlTypes

System.Xml

XSLT

Serialization

XPath

System

Collections

IO

Security

Runtime

Configuration

Net

ServiceProcess

InteropServices

Diagnostics

Reflection

Text

Remoting

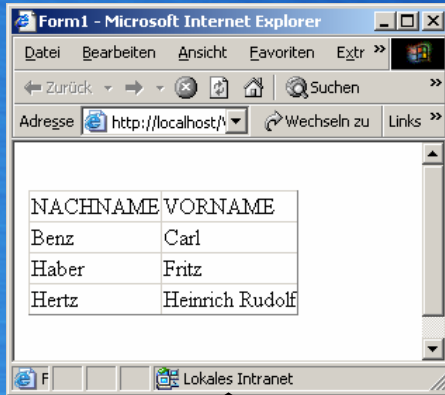
Globalization

Resources

Threading

Serialization

Beispiel-Szenario



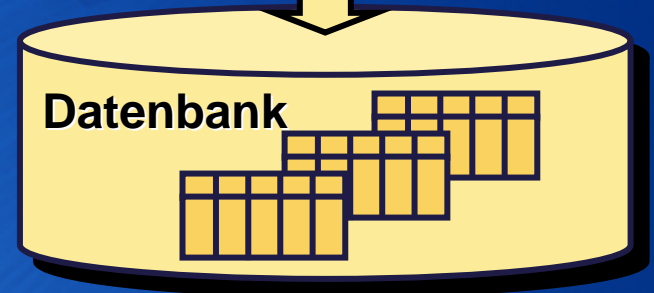
ASP.NET



Web Service

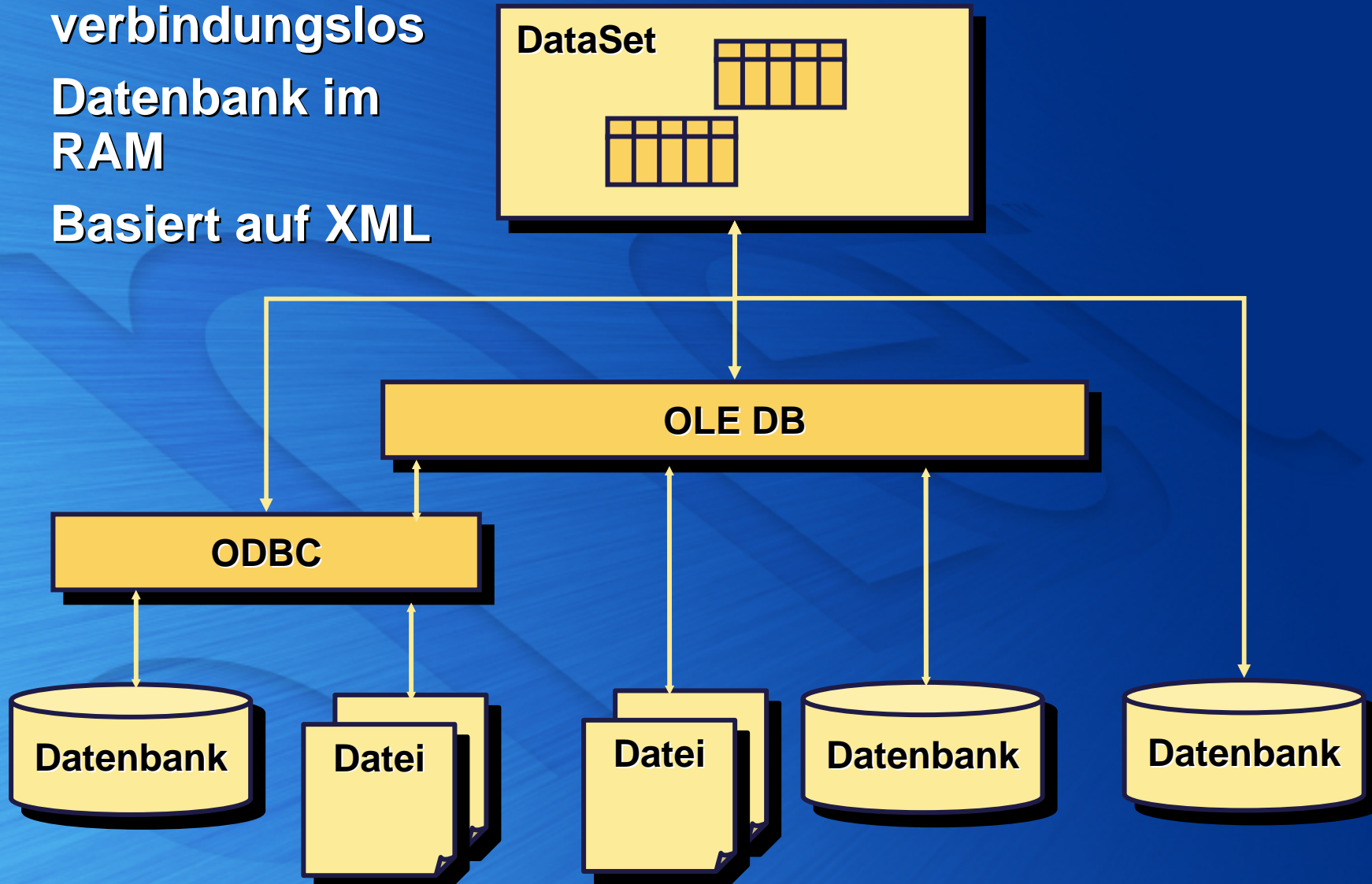


ADO.NET



Kernkonzept ADO.NET

- verbindungslos
- Datenbank im RAM
- Basiert auf XML



Data Provider

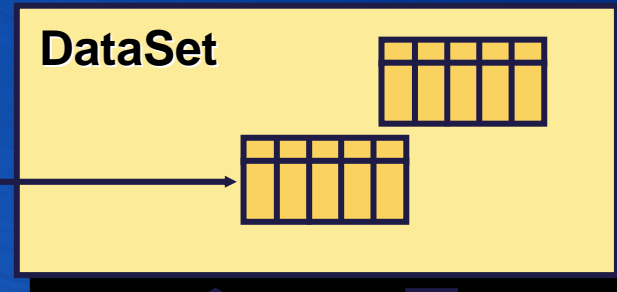
- **Verbindung zur Datenquelle**
- **Ausführen von SQL-Anweisungen**
- **Ermitteln von Daten zur weiteren Verarbeitung**
- **Beispiel Data Provider**
 - **OLE-DB über COM**
 - **Spezielle für SQL Server, Oracle, MySQL,...**
 - **ODBC Provider**
 - **...**

Data Provider Objekte

- **Providerspezifisch**
 - **Connection**
(z.B. `SqlConnection`, `OleDbConnection`)
 - **Command**
 - Führt SQL-Queries aus;
Verarbeitet Parameter
 - **DataReader**
 - Liest Datenströme
- **Providerunabhängig**
 - **DataSet**
 - Verbindungslos; „In Memory Cache“

DataAdapter

Relationaler Zugriff
auf kopierte Daten,
verbindungslos



Fill

Update

DataAdapter

Connection

Select ... Insert...
Update... Delete...

TableMapping

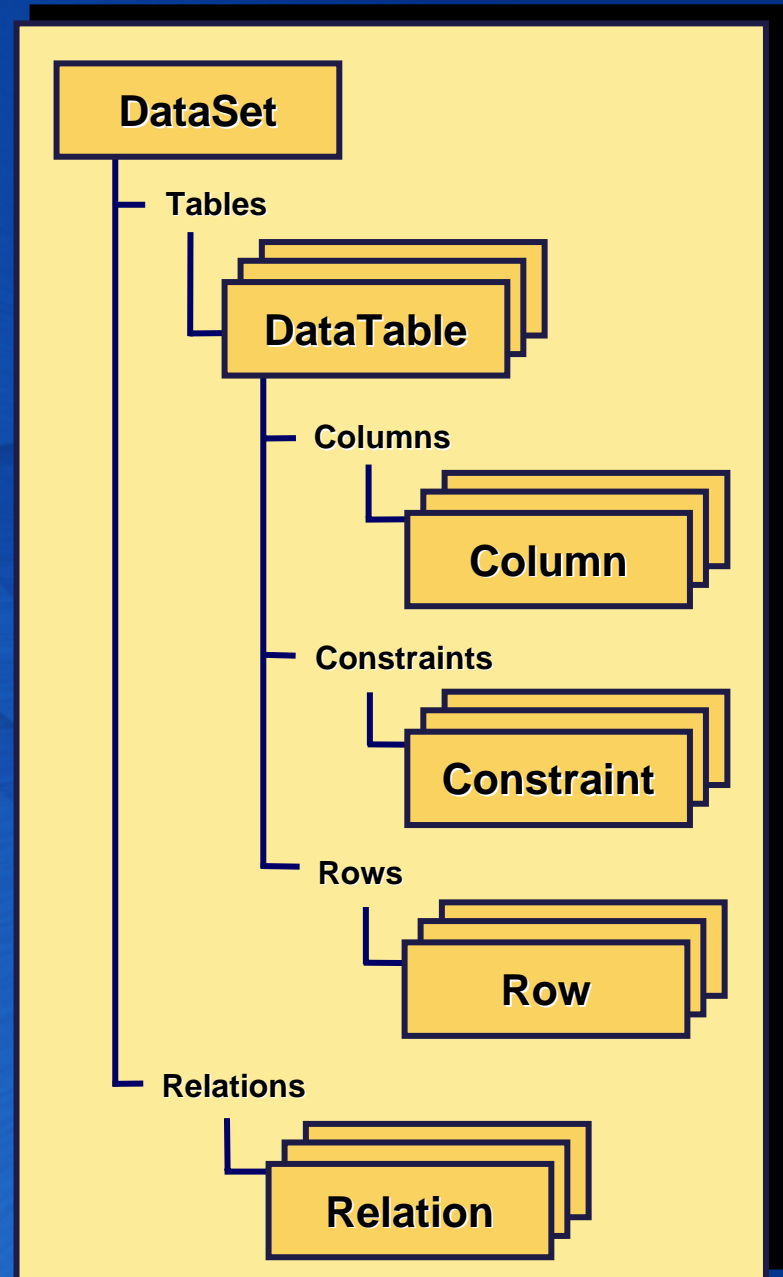
Datenbank



DataAdapter
ruft das entsprechende,
vorher festgelegte
SQL-Statement auf

DataSet

- **Relationale Sicht auf Datenquelle**
- **Daten \leftrightarrow XML
Schema \leftrightarrow XSD**
- **XSD-Schema eines DataSets kann...**
 - ... „von Hand“ erzeugt werden
 - ... vom DataAdapter generiert werden



DataTable

- **Repräsentiert eine Tabelle mit Daten im Arbeitsspeicher**
- **Besitzt Methoden um ...**
 - **Daten einzufügen,**
 - **Daten zu ändern und**
 - **Daten zu löschen**
- **Einfache Select-Anfragen**
- **Kann über DataRelation mit anderen Tabellen verbunden werden**

Was ist ein Web Service?

- Versuch einer Definition:

„A Web service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet-based protocols.“

[Quelle: W3C - Web Services Architecture Requirements]

Was ist ein Web Service?

- **Web Service Eigenschaften:**
 - **Protokolle: SOAP, XML, HTTP, HTML**
 - **Definition mit WSDL**
 - **Für eine Vielzahl von Clients verwendbar**
 - **Web Client Applikation**
 - **XML oder HTML Browser (Pocket PC, Handy ...)**
 - **...**
- **Deutliche Abgrenzung zu DCOM / CORBA Services**

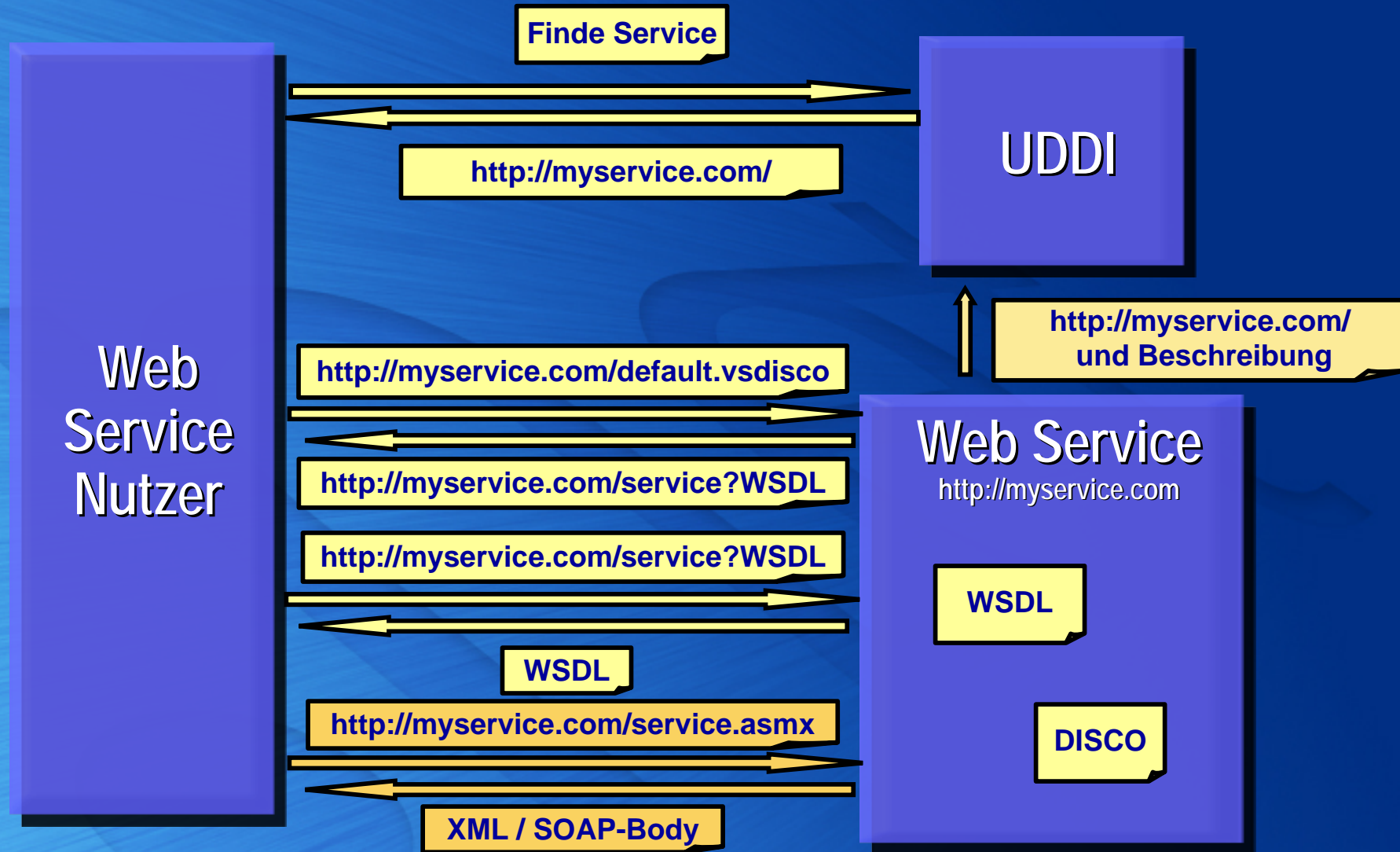
.NET Services



Welt der Web Services

- **Wie findet man einen Web Service?**
 - **UDDI**
 - Universal Description, Discovery and Integration
 - Weiße, Gelbe, Grüne Seiten
 - **Disco (MS-Standard, Discovery)**
 - Spürt gewünschten Service auf
- **Wie ruft man ihn auf?**
 - **WSDL**
 - Schnittstellenbeschreibung
 - Ermöglicht manuelle oder automatische Konfiguration der Clients

Web Service Aufruf



Registrierung

Analyse/Auswahl

Eigentlicher SOAP-Aufruf

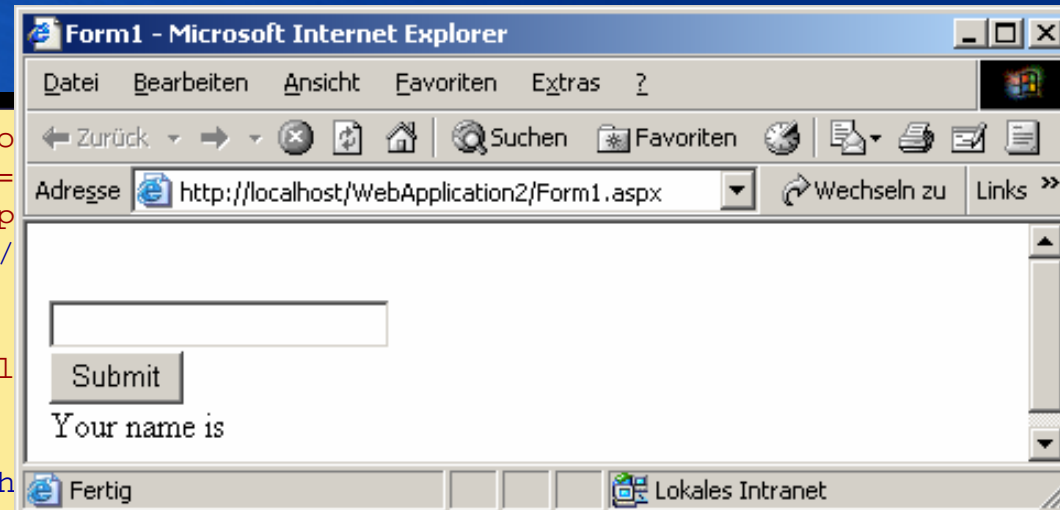
Web Forms mit ASP.NET

Form1.aspx

```
<%@ Page language="c#" CodeBehind="Form1.aspx.cs"
AutoEventWireup="true" Inherits="WebApplication2.Form1" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>WebForm1</title>
  </head>
  <body>
    <form id="Form1" method="post">
      <asp:TextBox id="TextBox1" runat="server"></asp:TextBox>
      <br>
      <asp:Button id="SubmitButton" runat="server"
        Text="Submit"></asp:Button>
      <br>
      <asp:Label id="NameLabel" runat="server">
        Your name is</asp:Label>
    </form>
  </body>
</html>
```

```
...
public class WebForm1 : Page
{
    protected System.Web.UI.WebControls.TextBox TextBox1;
    protected System.Web.UI.WebControls.Button SubmitButton;
    protected System.Web.UI.WebControls.Label NameLabel;

    private void SubmitButton_Click(object sender,
        System.EventArgs e)
    {
        NameLabel.Text += TextBox1.Text;
    }
    ....
}
```



Form1.aspx.cs

Web Forms mit ASP.NET

- **Web Forms sind ein Steuerelemente-basiertes Ereignis-gesteuertes Modell für Web User Interfaces:**
 - **Steuerelemente → HTML-Code**
 - **HTML-Aufrufe → Ereignisse**
- **Trennung HTML und Code in zwei unterschiedliche Dateien möglich**
- **HTML-Seiten erstellen entspricht Programmieren einer „normalen“ Applikation**

Steuerelemente (Controls)

- **ASP.NET enthält 45 Steuerelemente**
- **HTML-Controls**
 - **Controls / Properties entsprechen 1:1 HTML-Elementen**
- **Web-Controls**
 - **Grundlegende Controls (Button, Label, ...)**
 - **List-Controls**
 - **Rich-Controls (DataGrids, ...)**
 - **Validation-Controls**
 - **Zustandsbehaftet**

Data Binding

- **Datenquellen können an Steuerelemente gebunden werden**
 - z.B. DataSet, Collection, eigene Objekte
- **Die Datenbindung ist vielseitig**
 - **Einfaches Binden**
 - Vorname → TextBox.Text
 - **Komplexes Binden**
 - Tabelle Kunde → DataGrid.DataSource
- **Datenbindung an fast alle Eigenschaften eines Steuerelements**

Ereignisse (Events)

- **HTML-Elemente können HTML-Meldungen senden**
 - zB: Button → Click-Ereignis (HTML-POST)
- **Reaktion in der Web Form:**
 - In Funktion: `Button_Clicked(...)`
- **Event-Handler muss mit dem Button verbunden werden:**
 - `Button.Click += new EventHandler(Button_Clicked)`

Seitenaufruf bei WebForms

Seite laden und initialisieren

Page_Load

Steuerereignisse

- 1. Wechselereignis
oder
- 2. Aktionsereignis

Textbox1_Changed

Button1_Click

Page_Unload

Seite wegwerfen

Page_Load Ereignis

- Wird bei jeder neuen Instanz der Seite ausgelöst
 - Alle Eingabewerte sind bereits vorhanden
 - Noch keine Ausgabe an den Client erfolgt

```
void Page_Load (Object sender, EventArgs e)
{
    // IsPostBack ist ein Property der Page Klasse
    if (IsPostBack == false)
        // Hier wird z.B. eine Datenbank abgefragt,
        // mit dessen Werten einige Controls der Seite
        // initialisiert werden.
}
```

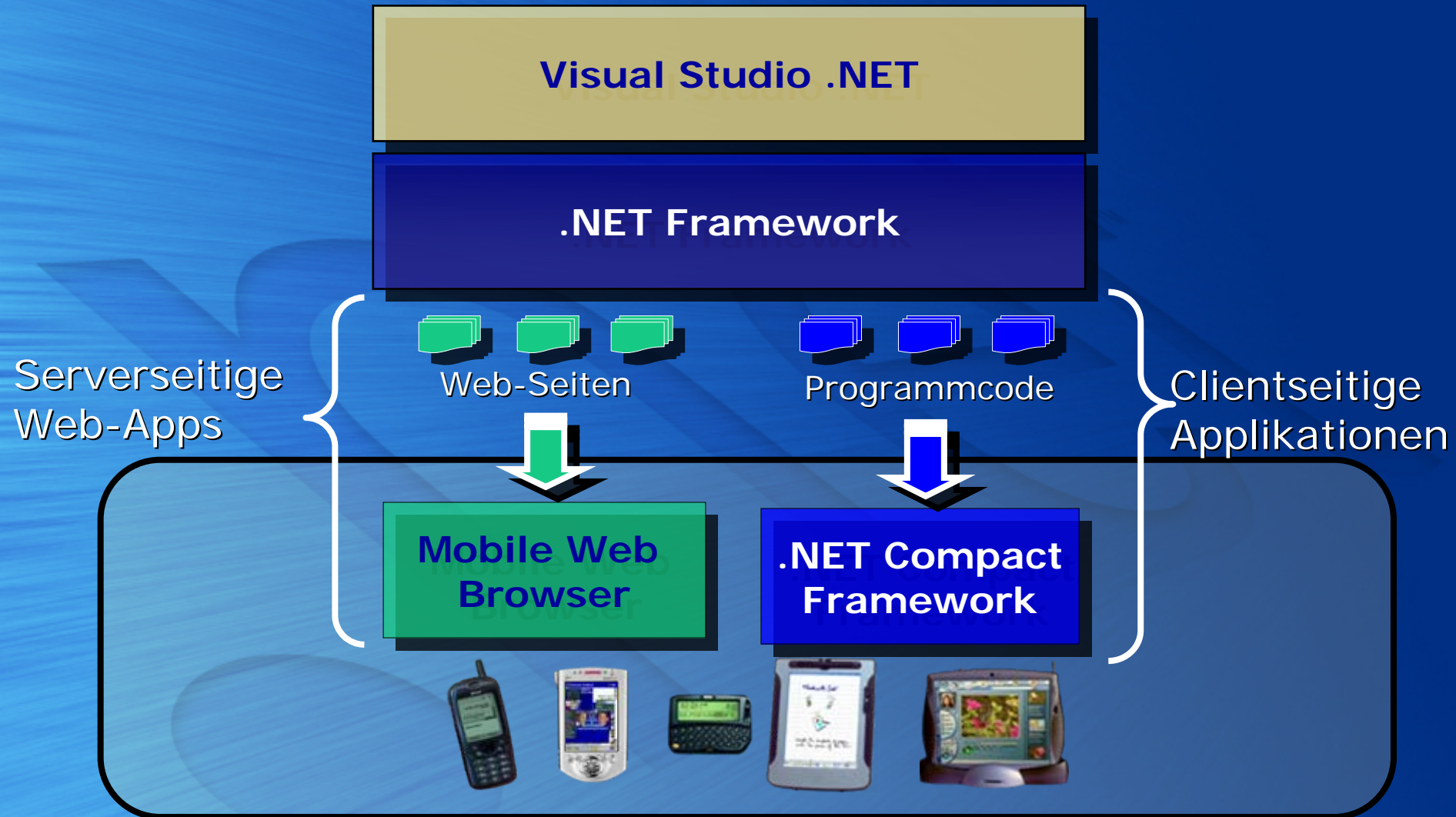
Die Page Klasse

- **Repräsentiert eine HTML-Seite**
- **Properties**
 - **IsPostBack**
 - POST-Kommando kommt von Webseite zurück
 - **Session**
 - Verwaltung von Sitzungsdaten
 - **Request**
 - Hereinkommender Aufruf
 - **Response**
 - Antwort an Client

Unterstützung mobiler Geräte

- **.NET Compact Framework**
 - Laufzeitumgebung auf mobilen Geräten
 - Anwendungen für ein mobiles Gerät
 - zB lauffähig auf Pocket PC (Windows CE) ...
 - Windows Mobile auf dem PDA
- **ASP.NET Mobile Web Forms**
 - Besondere ASP.NET Steuerelemente für Browser auf mobilen Geräten
 - Erzeugen dynamisch für jeden der Browser das richtige Ausgabeformat

Entwicklungsplattform



ASP.NET Mobile Controls



← Entwicklungs-
umgebung →

← Laufender
Betrieb →

Performance (10⁷ Runs)

- $(((\text{double}) i + (\text{double}) j) * 2.5) / (i + j + 1)$

- VB 6.0 80.313 sek

- VC++ 6.0 24.891 sek

- C++ .NET unmg. 24.890 sek

- C++ .NET mng. 24.850 sek

- C# 24.891 sek

- VB .NET 24.990 sek

VS 6.0

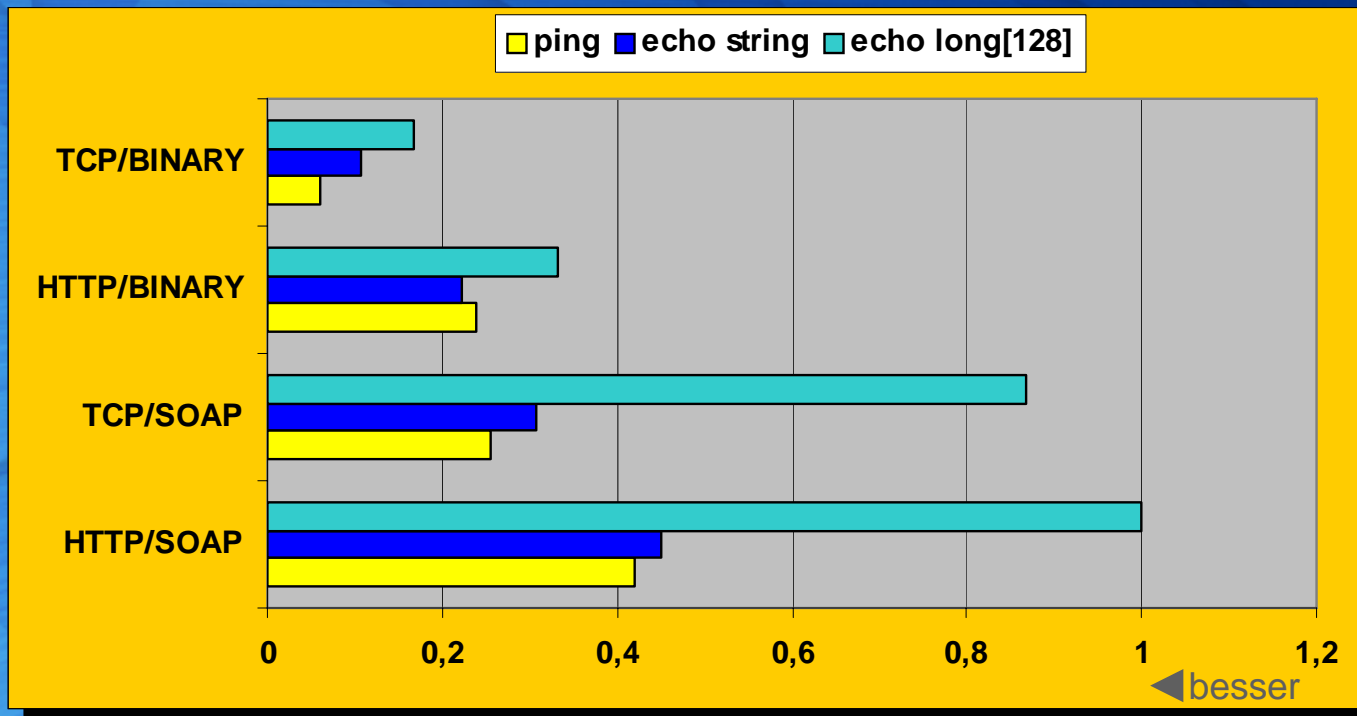
Asm.

VS .NET

CLR

Remoting: Binär vs. SOAP

- Laufzeit entfernter Funktionsaufruf:
 - Ohne Parameter
 - Ein-/Ausgabeparameter: string
 - Ein-/Ausgabeparameter: long[128]



.NET vs. J2EE : Pet Store

- Web-Portal für Online-Handel
- Messung:
 - Antwortzeiten des Servers unter Last



Java: Oracle, IBM



.NET: Microsoft

.NET vs. J2EE

.NET Pet Shop schlägt J2EE

1/4 weniger Code

Unter der Oberfläche komplett anders implementiert:

z.B. EJB/JDBC vs. ASP.NET/Stored procedures

Pet Store ist kein Benchmark, sondern Beispielanwendung für EJBs

zeitiger Nutzer

.NET

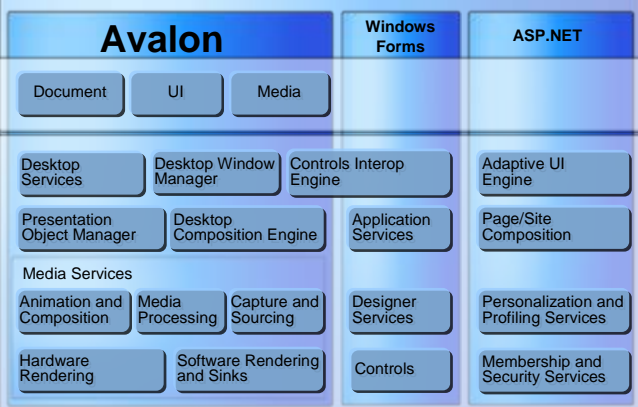
J2EE

Lessons Learned

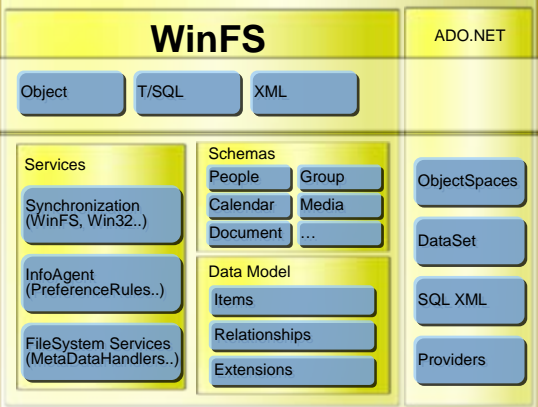
- **Einheitliche Laufzeitumgebung**
 - aber eigentlich keine neuen Konzepte ...
- **Einheitliches Programmiermodell**
 - aber Visual Studio noch fehleranfällig ...
 - „Komplexere“ Applikationen erfordern viel Handarbeit, viel Geduld und gute Nerven ... wie immer ...

Longhorn Architecture

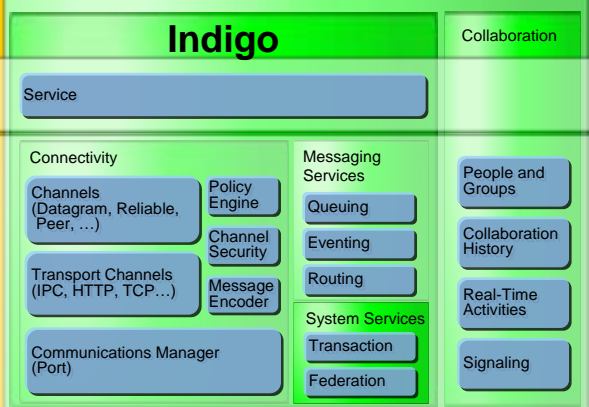
Presentation



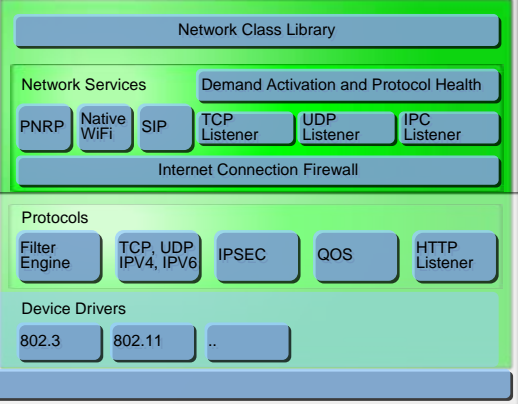
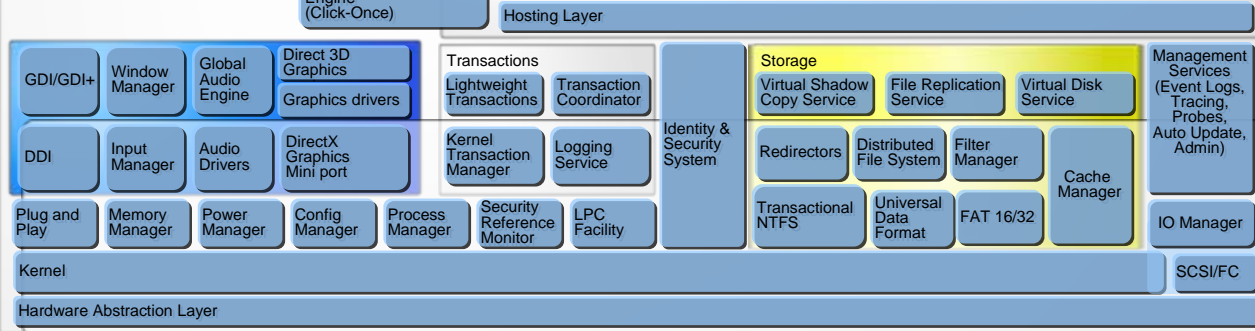
Data



Communication



Base Operating System Services



Literatur + Software

- **Microsoft Developer Network**
 - <http://msdn.microsoft.com>
- **Microsoft Software**
 - <http://www.studentprogram-south.de/cms/home/layout/64.aspx>

Fragen?

