Praktische Einführung in *.NET*

Kai Stammerjohann

Inhalt

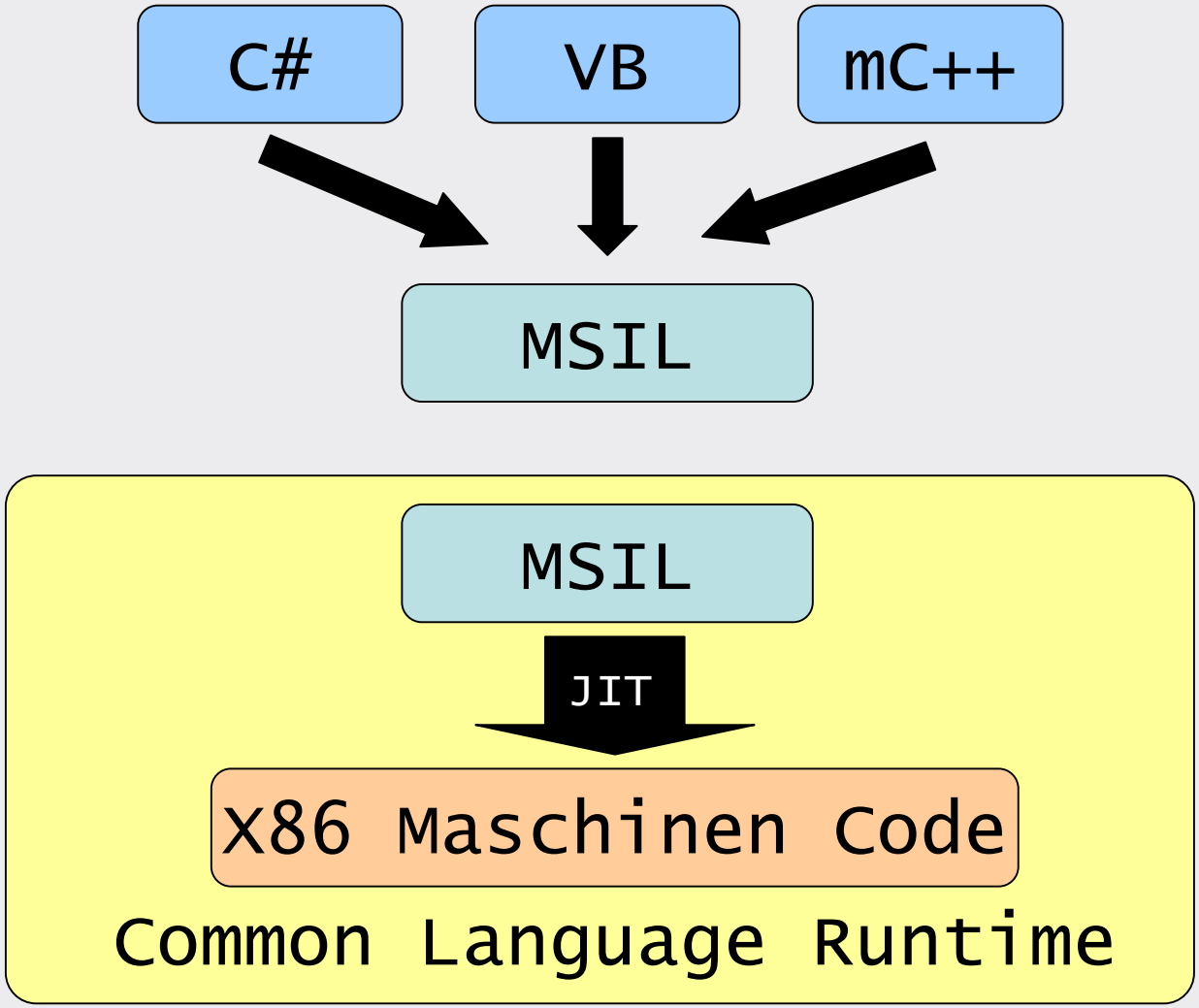
- CLR Eigenschaften
- Datenbankzugriff
- Web Service
- Webforms
- Schlussfolgerung

Inhalt

- CLR Eigenschaften
- Datenbankzugriff
- Web Service
- Webforms
- Schlussfolgerung

CLR Eigenschaften

Struktur



CLR Eigenschaften

Sprachübergreifende Programmierung

C#:

```
public class CSKlasse
{
        public void hallo(string text)
        {
                System.Console.WriteLine(text);
        }
}
```

mC++:

```
#using "CSKlasse.dll"

int main(void)
{
        CSKlasse *obj = new CSKlasse();

        obj->hallo("test");

        return 0;
}
```

CLR Eigenschaften

De-Compiler

Orginal:

```
public class CSKlasse
{
        public void hallo(string text)
        {
                System.Console.WriteLine(text);
        }
}
```

Zurückübersetzt:

```
C:\hs\Exemplar\release>exemplar CSKlasse.dll

public class CSKlasse {
        public void hallo(string text) {
                System.Console.WriteLine(text);
        }

        public CSKlasse() : base() {
        }
}
```

Inhalt

- CLR Eigenschaften
- Datenbankzugriff
- Web Service
- Webforms
- Schlussfolgerung

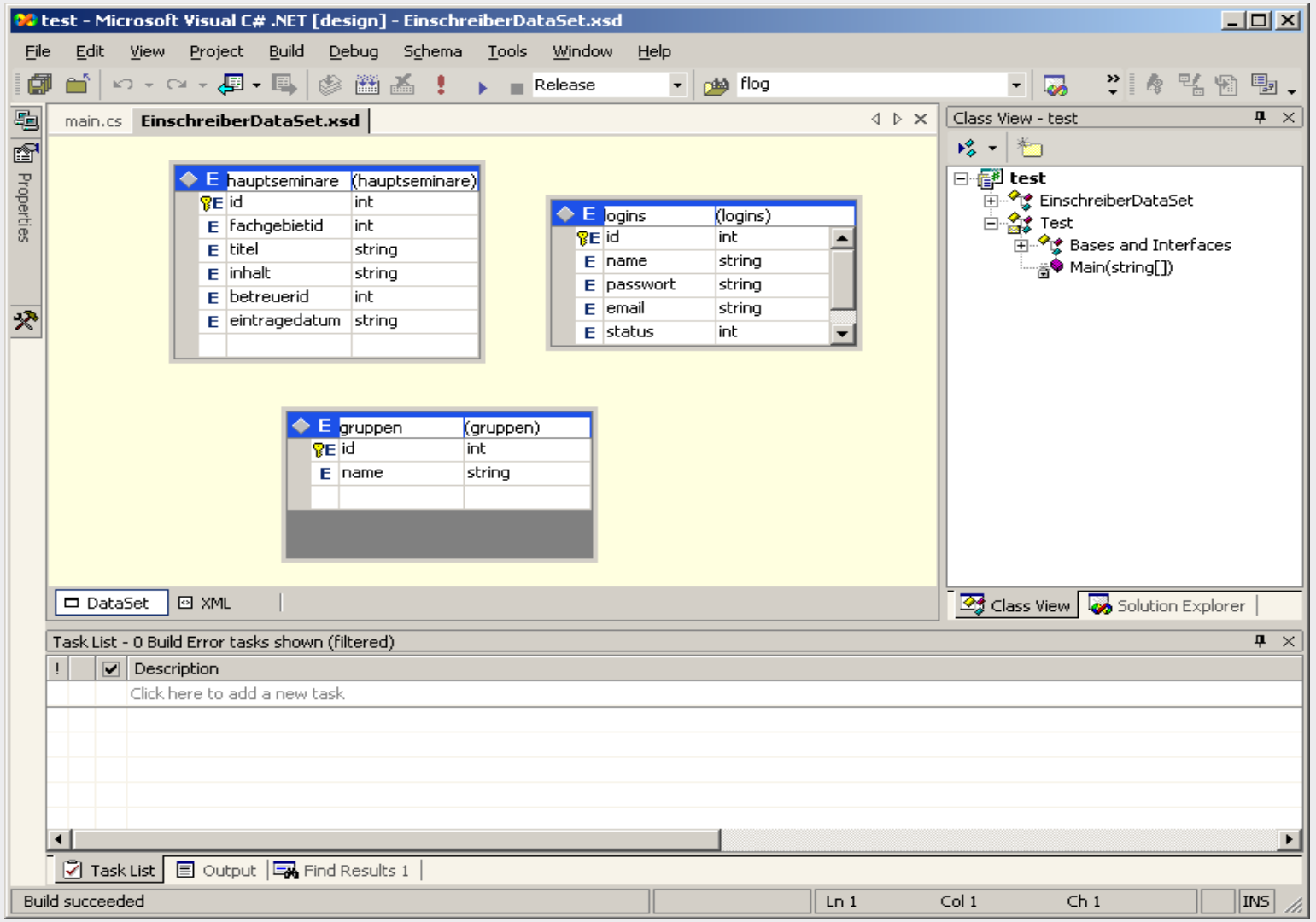
Datenbankzugriff

Herkömmlicher Datenbankzugriff

```
SqlConnection conn = new SqlConnection( ... );
SqlCommand cmd = new SqlCommand("SELECT titel FROM hauptseminare", conn);

conn.Open();

SqlDataReader reader = cmd.ExecuteReader();

while(reader.Read())
{
   Console.WriteLine("{0},{1}", reader.GetString(0), reader.GetString(1));
}

reader.Close();
conn.Close();
```

Datenbankzugriff

Datenbankschema

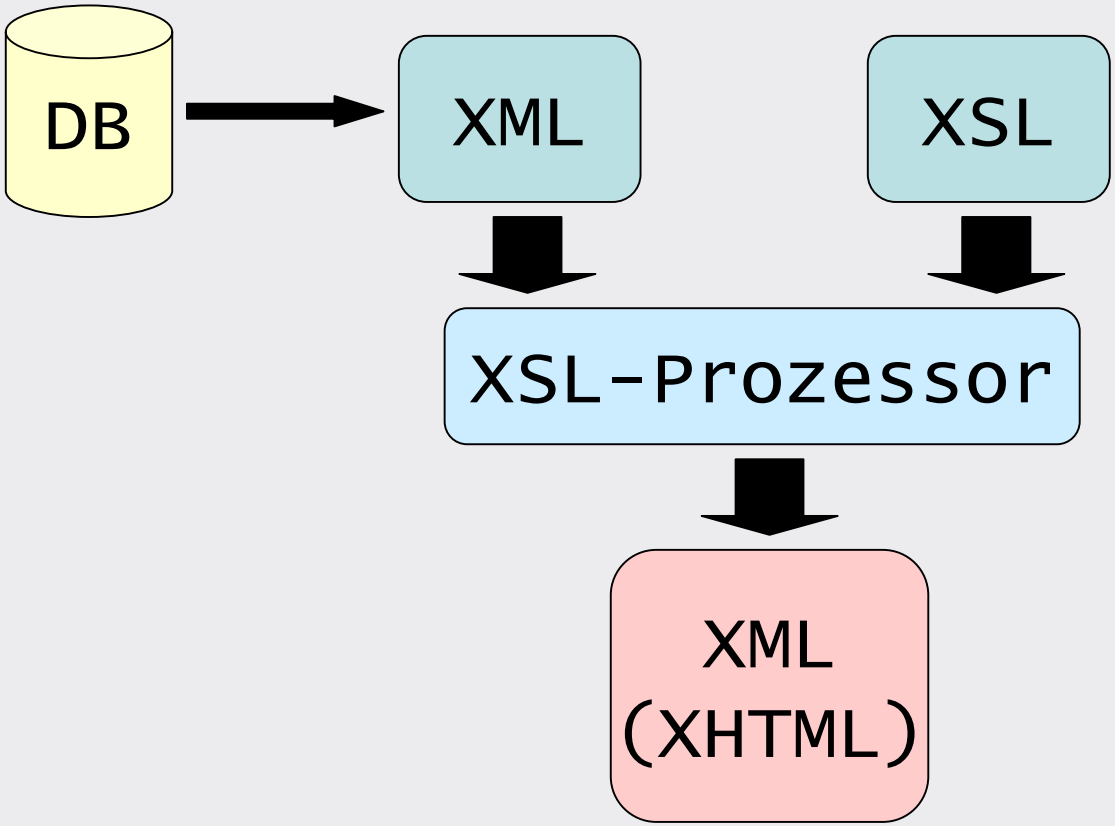


ADO.NET Datenbankzugriff

```
SqlConnection conn = new SqlConnection( ... );
SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM hauptseminare", conn);
EinschreiberDataSet ds = new EinschreiberDataSet();

conn.Open();
da.Fill(ds, "hauptseminare");
conn.Close();

foreach(EinschreiberDataSet.hauptseminareRow row in ds.hauptseminare)
{
   Console.WriteLine("{0}, {1}", row.titel, row.inhalt);
}
```

Datenbankzugriff

XML/XSL



Datenbankzugriff

XML/XSL Datenbankzugriff

```
SqlConnection conn = new SqlConnection( ... );
SqlDataAdapter da= new SqlDataAdapter("SELECT * FROM hauptseminare", conn);
EinschreiberDataSet ds = new EinschreiberDataSet();

conn.Open();
da.Fill(ds, "hauptseminare");
conn.Close();

XmlDataDocument xml = new XmlDataDocument(ds);
XslTransform xsl = new XslTransform();
XmlTextWriter writer = new XmlTextWriter("output.html");

xsl.Load("format.xsl");

xsl.Transform(xml, null, writer);

Writer.close();
```

Inhalt

- CLR Eigenschaften
- Datenbankzugriff
- Web Service
- Webforms
- Schlussfolgerung

Web Service

Definition

- Web Service = Dienst mit XML Schnittstelle
- Benutzung durch Programm oder menschlichen Anwender
- Kommunikation mit SOAP
- Registrierung in UDDI

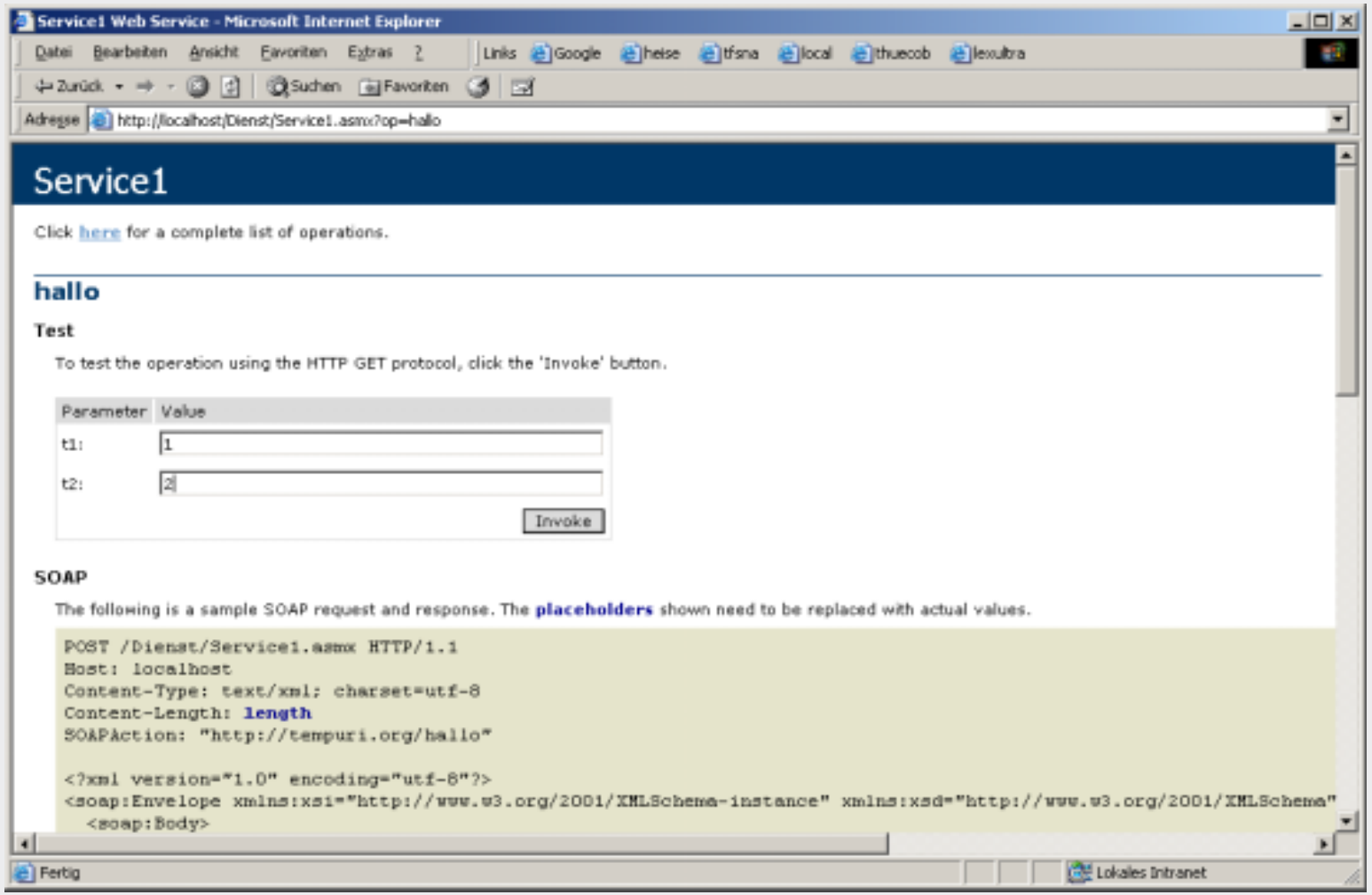
Web Service

Dienst Server

```
public class Service1 : System.Web.Services.WebService
{
        [WebMethod]
        public Result execute(Param param)
        {
                return param.execute();
        }

        [WebMethod]
        public string hallo(string t1, string t2)
        {
                return "server meint zu " + t1 + " das da " + t2;
        }
}
```

Web Service

Dienst Anfrage



Web Service

SOAP Anfrage Paket

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

        <soap:Body>

                <hallo xmlns="http://tempuri.org/">

                        <t1>1</t1>
                        <t2>2</t2>

                </hallo>

        </soap:Body>

</soap:Envelope>


entspricht: hallo("1", "2");
```

Web Service

SOAP Antwort Paket

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://uri/">server meint zu 1 das da 2</string>
```

Web Service

Dienstbeschreibung

- WSDL (Web Service Description Language)
- Beschreibung aller Funktionen des Dienstes
- Beschreibung aller Parameter und Rückgaben
- Beschreibung aller Datentypen

Web Service

Dienst Schnittstelle

```
public Result execute(Param param)
{
        return param.execute();
}
```

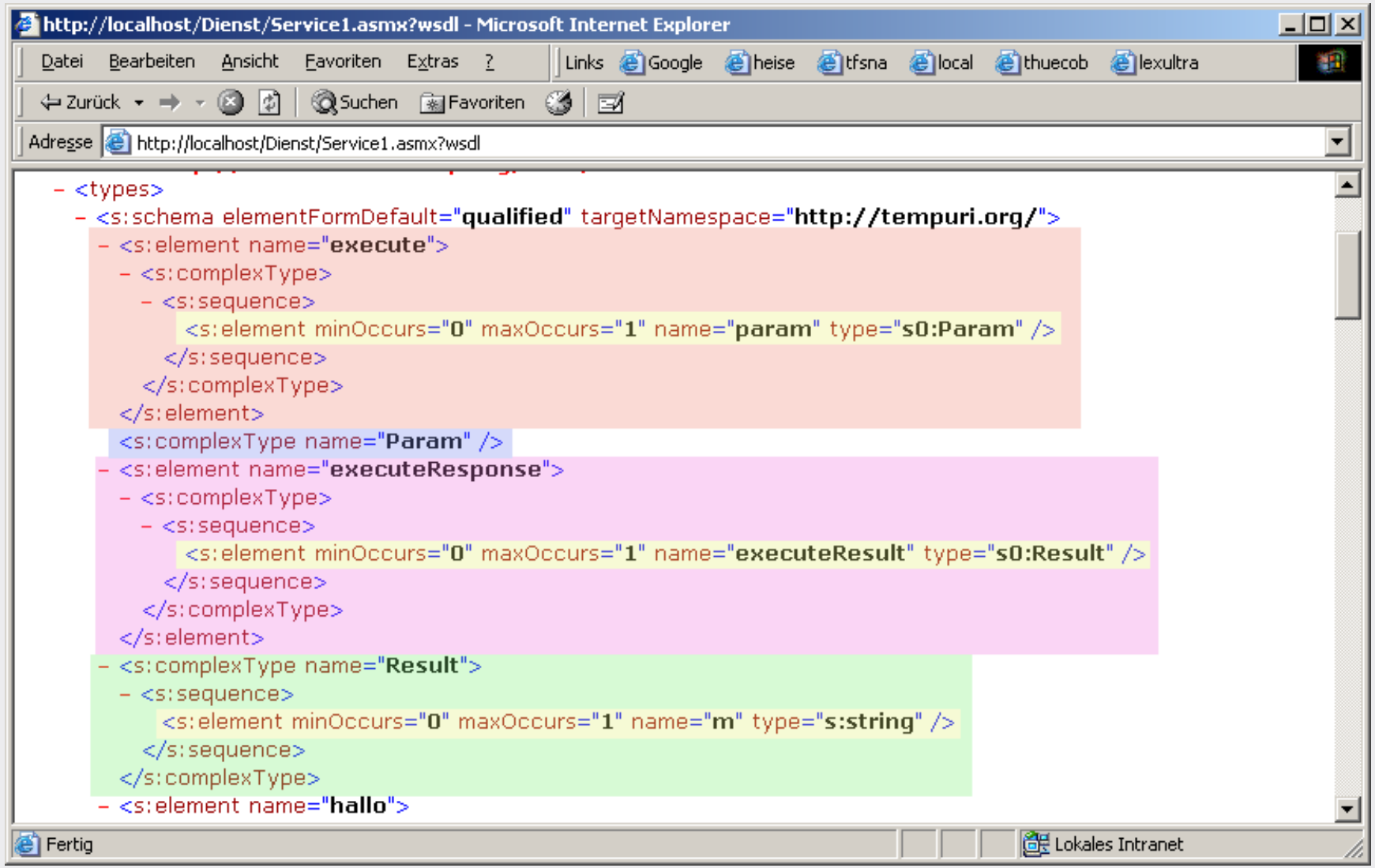
Web Service

Dienst Datentypen

```
public class Param
{
        public Result execute()
        {
                return new Result("ergebnis");
        }
}

public class Result
{
        public string m;

        public Result(string m)
        {
                this.m = m;
        }
}
```

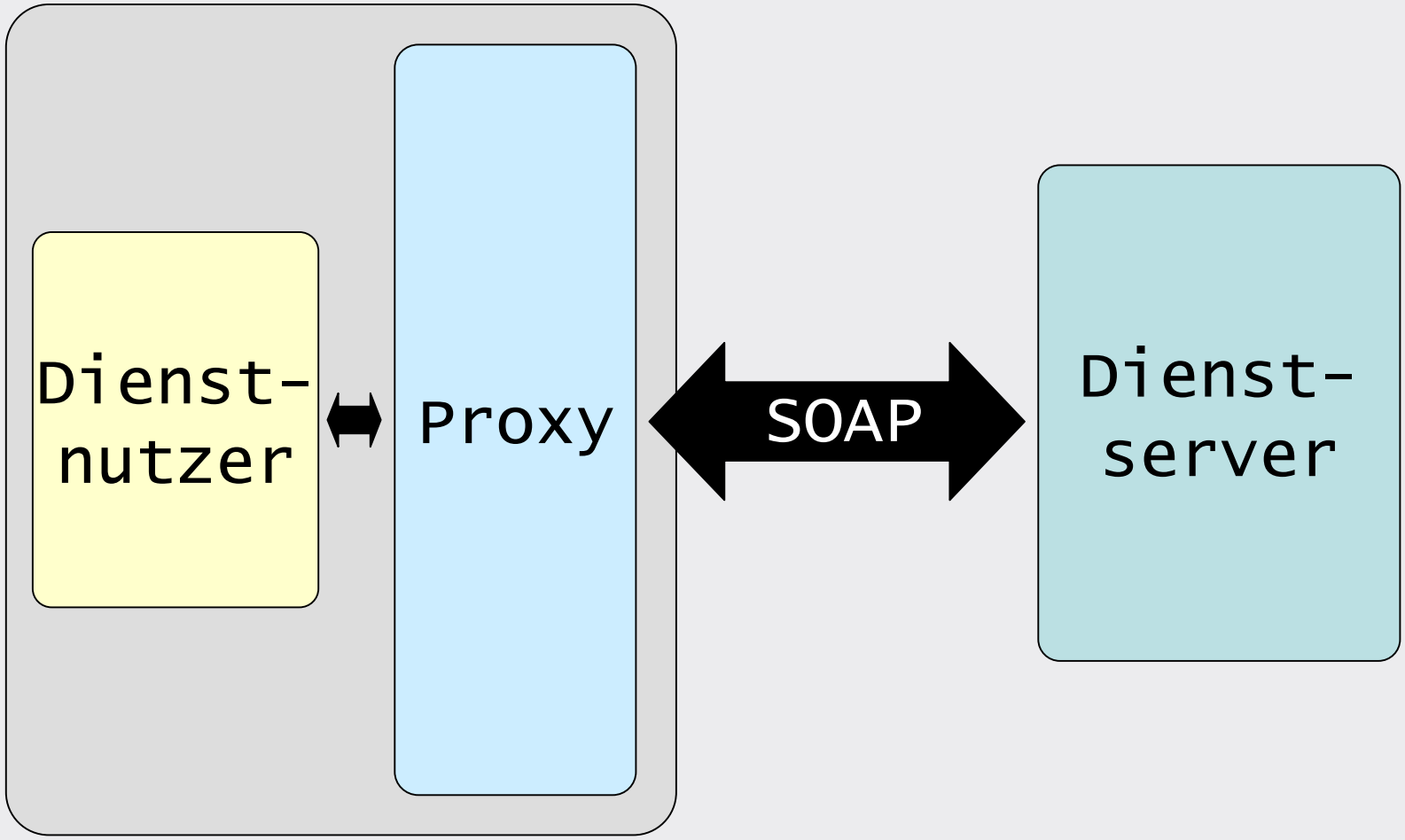
Web Service

WSDL



Web Service

Dienst Client - Proxy



Web Service

Dienst Client

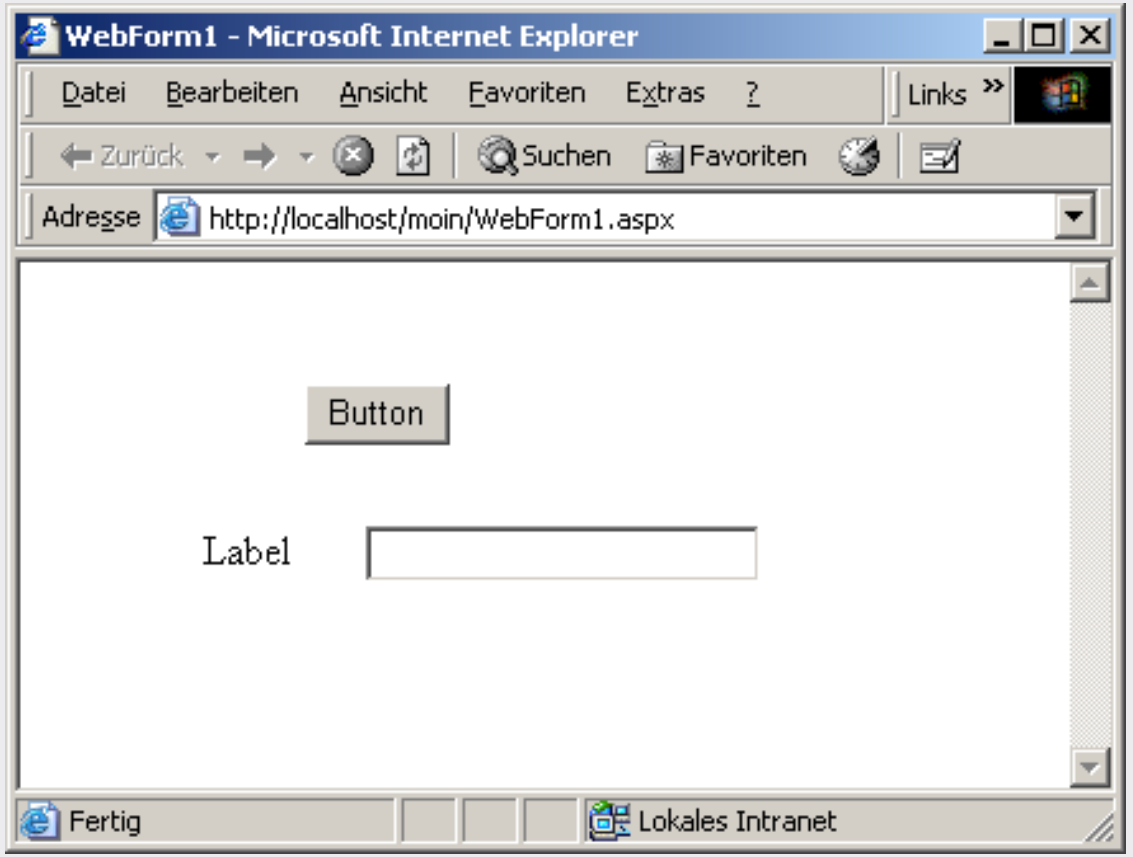
```
localhost.Service1 svc = new localhost.Service1();

localhost.Result result = svc.execute(new localhost.Param());

Console.WriteLine(result.m);

Console.WriteLine(svc.hallo("1", "2"));
```

Inhalt

- CLR Eigenschaften
- Datenbankzugriff
- Web Service
- Webforms
- Schlussfolgerung

Webforms

Webform, Webcontrol



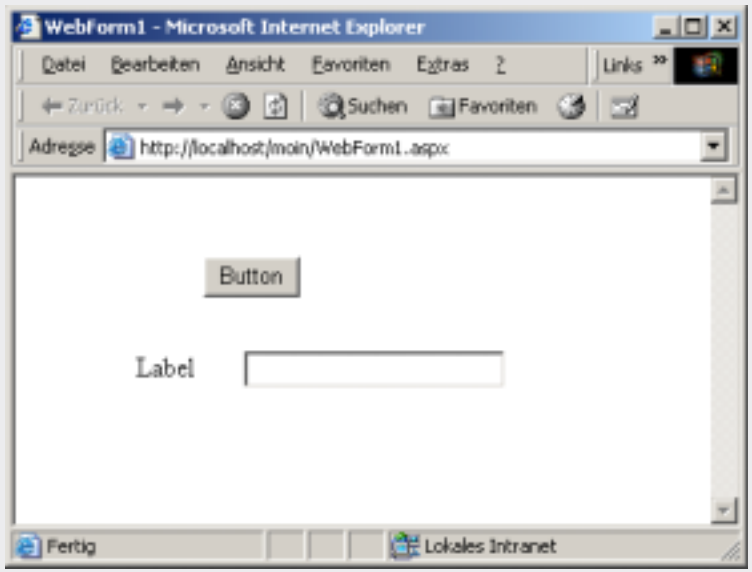
Webform, Webcontrol

- Zu jedem Webcontrol gibt es eine zugehörige Klasse
  Bsp: *<input>: System.Web.UI.WebControls.TextBox*
  <button>: System.Web.UI.WebControls.Button

- Parallel zum (WYSIWYG) Entwurf des Webforms wird eine zugehörige Webform Klasse erzeugt

- Webformklasse enthält jedes enthaltene Webcontrol als Variable entsprechenden Typs

Webform, Webcontrol Beispiel




```
public class WebForm1 : System.Web.UI.Page
{
    protected System.Web.UI.WebControls.TextBox TextBox1;
    protected System.Web.UI.WebControls.Button Button1;
    protected System.Web.UI.WebControls.Label Label1;

    . . .
}
```

Webforms

Benutzerinteraktion

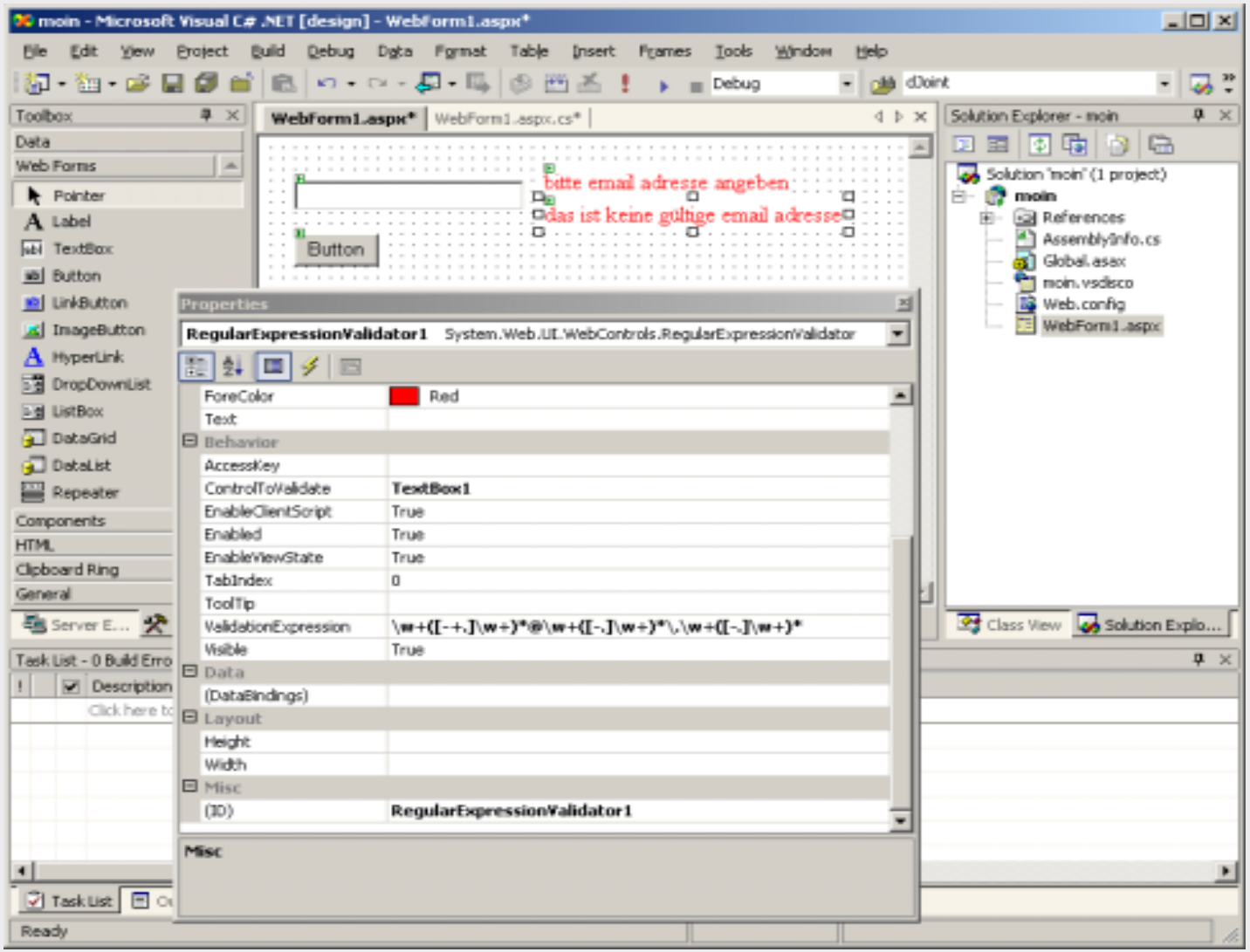
- verschiedene Eventhandler zu jedem Webcontrol
- zB OnClick Handler für Buttons

  Registrierung durch

```
this.Button1.Click += new System.EventHandler(this.Button1_Click);
```

Webforms

Validatoren



Inhalt

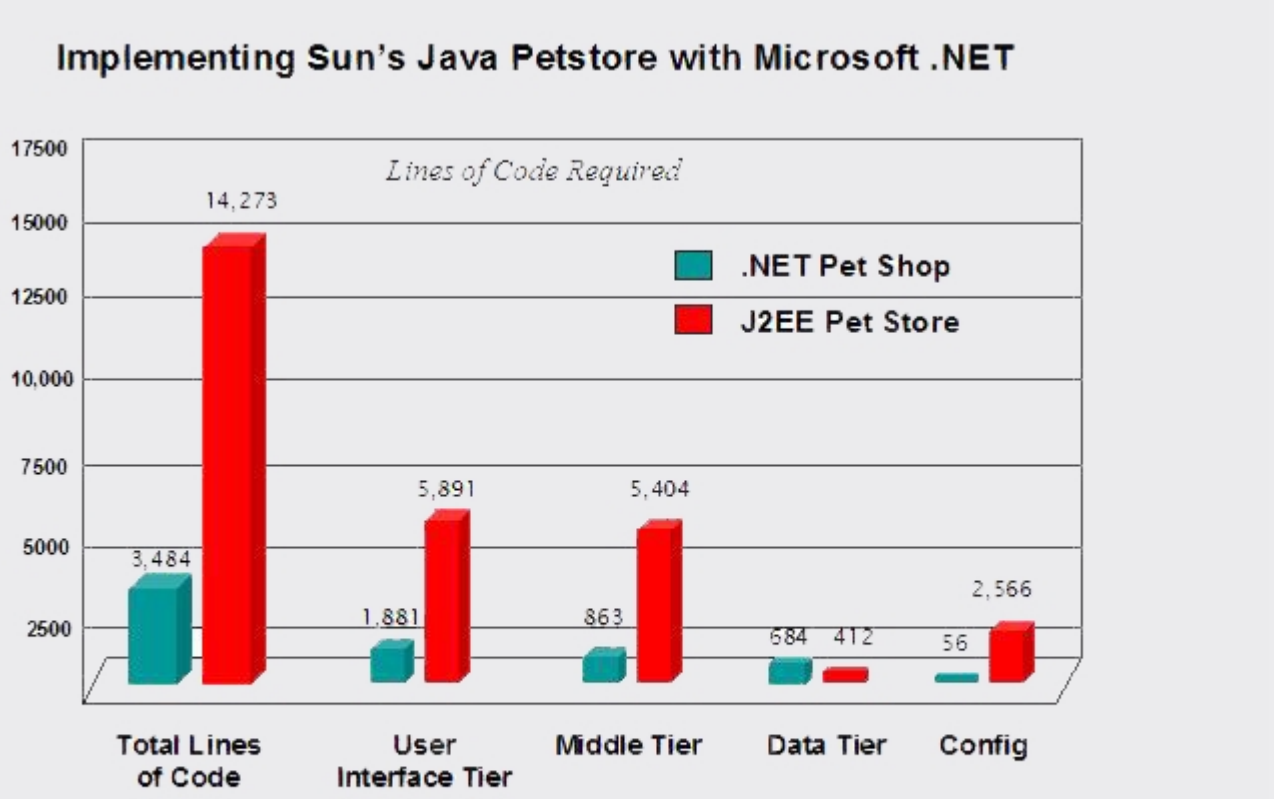
- CLR Eigenschaften
- Datenbankzugriff
- Web Service
- Webforms
- Schlussfolgerung

Schlussfolgerung

Alternative

- .NET Konkurenzplatform ist J2EE
- J2EE Referenzinternetanwendung PetStore
- Nachimplemtierung von VeriTest mit .NET

Schlussfolgerung



Ende

Fragen?