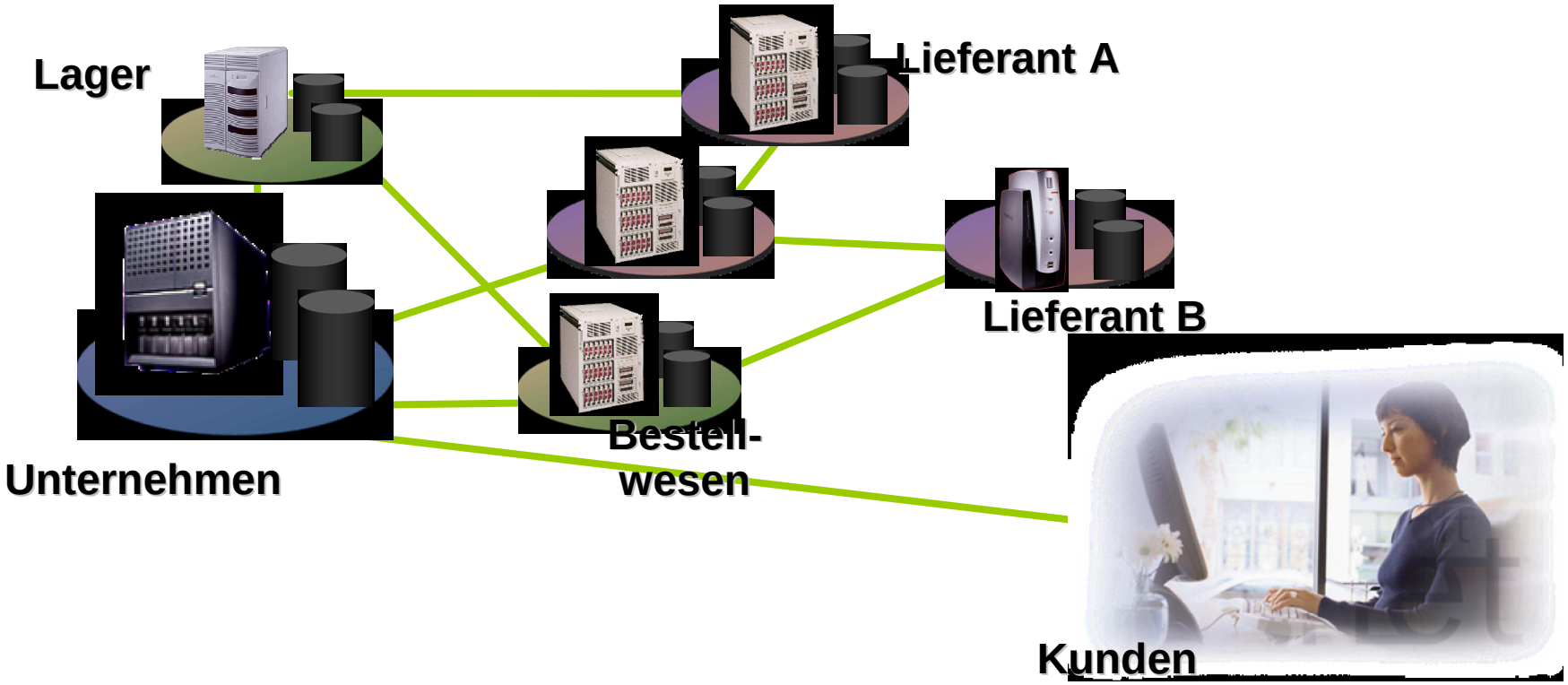
Webservices mit .NET

Dienste im Netz, die über Standard-Protokolle erreichbar sind

Datenaustausch basiert auf XML- Dokumenten

Plattform- und unternehmensübergreifende Transaktionen



MICRO CONSULT

Web Services bieten ...

- programmiersprachenunabhängige,
- protokollunabhängige,
- intelligente,
- kontextsensitive,
- XML- und webbasierte

Inter-Applikationskommunikation

Web Service Directory :	**UDDI**
Beschreibung von WS :	**WSDL**
Datenaustausch zw. WS :	**SOAP**
Universelles Datenformat :	**XML**
Kommunikationsnetz :	**Internet/HTTP**

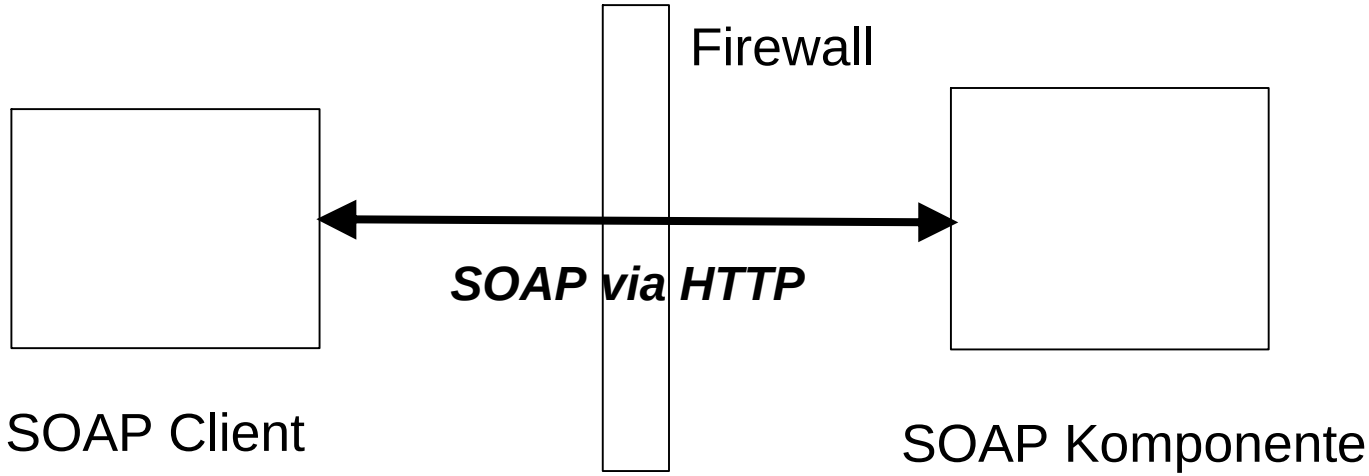
⇨ *einfache, offene und akzeptierte Standards*

Simple Object Access Protocol

- Lightweight Message Protokoll
- Typisierter Datenaustausch zwischen Applikationen
- Besonders geeignet für RPC Funktionalität
- Lose Kopplung
- Verteilte Architektur
- Transport: Binding über darunterliegendes Transportprotokoll
- Struktur wird in XML definiert

Problem :

- Wie können Komponenten oder Dienste auf fernen Rechnern problemlos angesprochen werden?
  - DCOM → Aufwendig und schwierig ; Stichwort : ***Firewall***
  - SOAP → Einfacher (kein Firewallproblem)



Das Simple Object Access Protocol (SOAP)

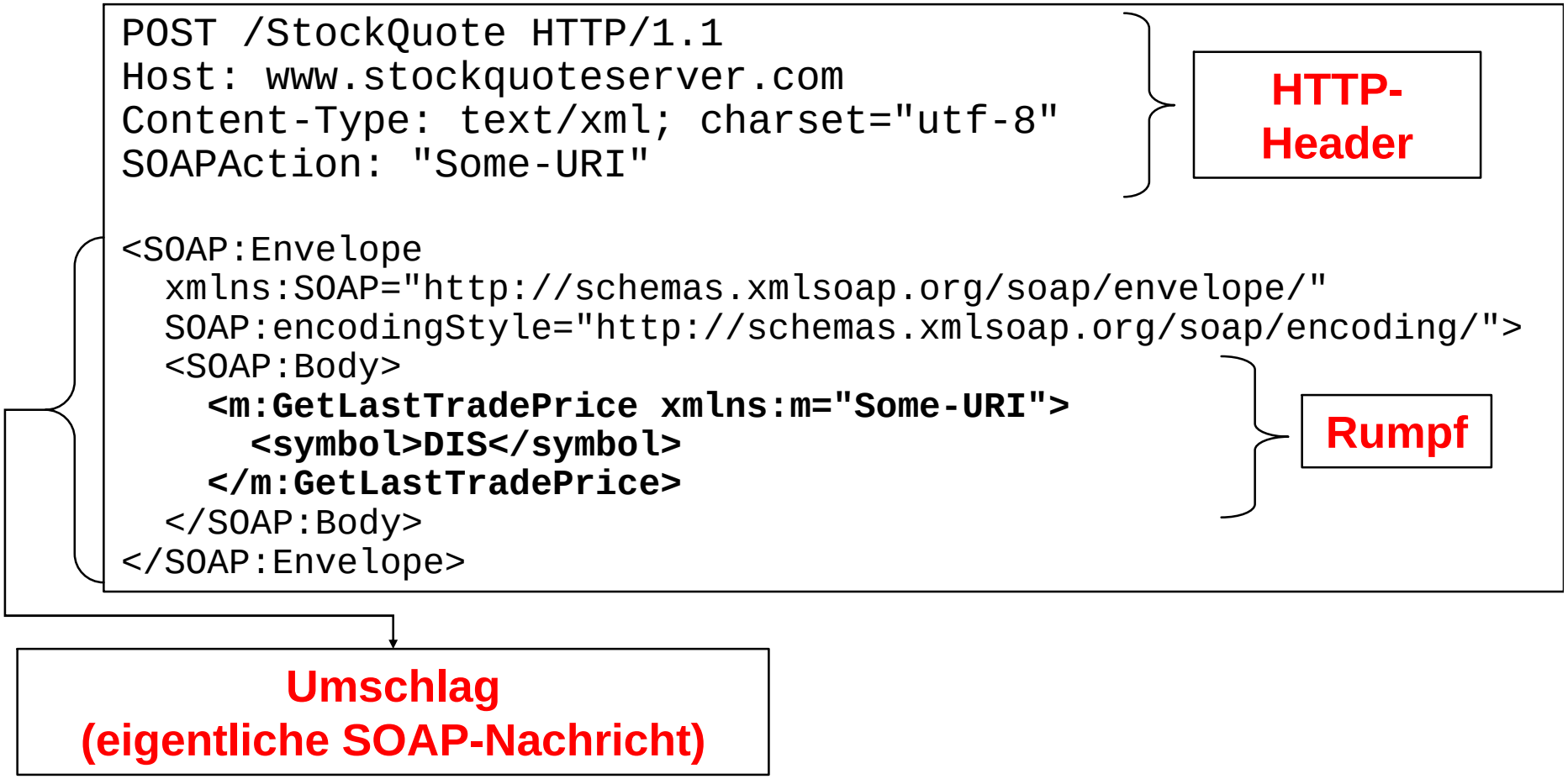
- SOAP stellt einfachen, auf XML basierenden Mechanismus zum Austausch von strukturierten Informationen zwischen zwei Mitgliedern einer dezentralisierten, verteilten Umgebung zur Verfügung. (W3C)

Ziele von SOAP:

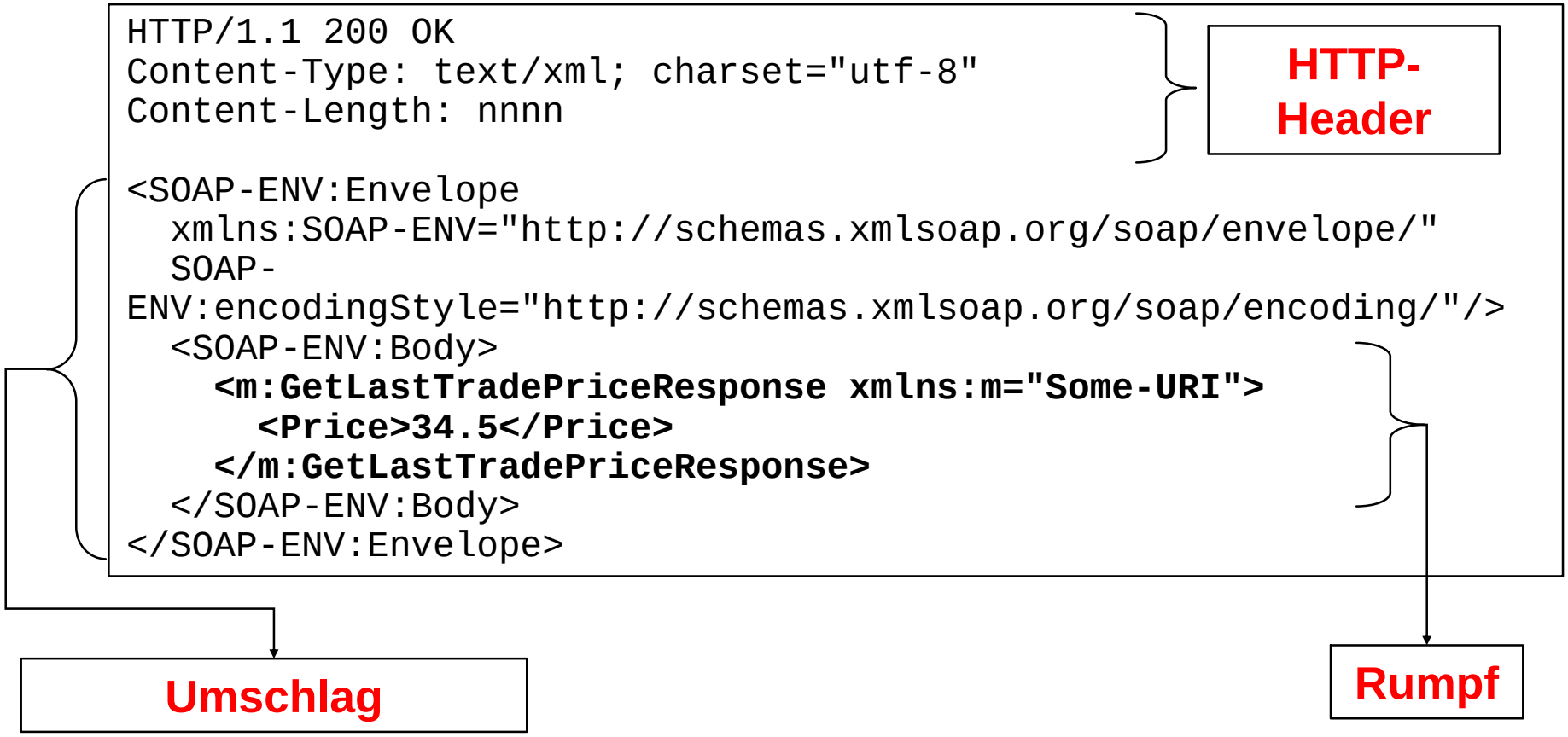
- Einfachheit und Erweiterbarkeit
- Definition eines Standardprotokolls für den Aufruf von Objekten
- Basis bilden Internetstandards : **HTTP für Transport** und **XML für Datenkodierung**

SOAP ist Basistechnologie der WebServices innerhalb von .NET !

Ein einfacher SOAP – Request über HTTP:



Ein einfacher SOAP – Response des Servers über HTTP :

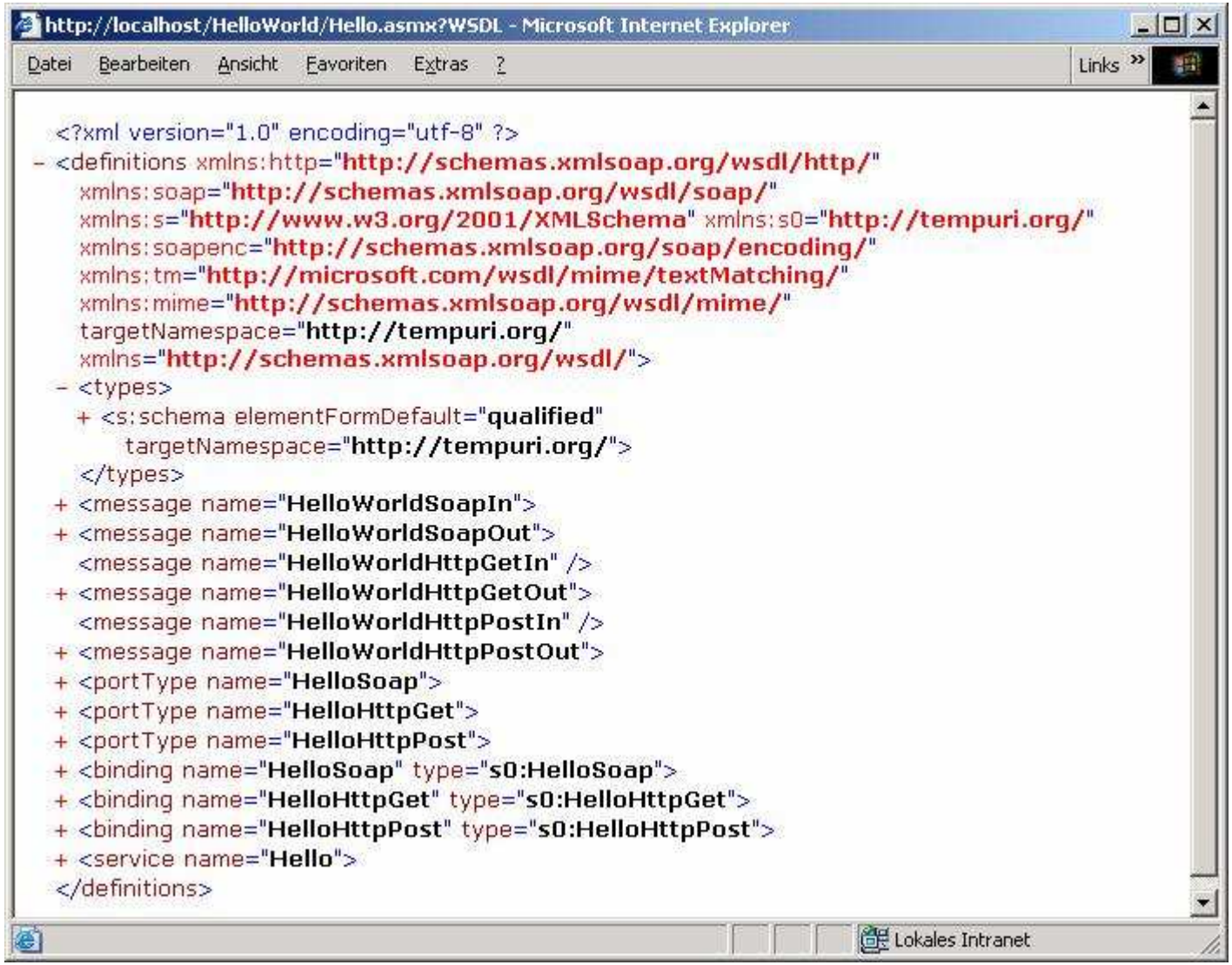


- Web Service Description Language
- W3C-Spezifikation (http://www.w3.org/TR/wsdl)
- XML basiert
- Beschreibung von Web-Services (wie IDL in CORBA)
- Kann aus Klassen gewonnen werden (und umgekehrt)

Gibt Anworten auf die Fragen:

1. Wo liegt der Web-Service genau?
2. Wie kann man mit ihm arbeiten, welche Schnittstellen hat er?
3. Mit welchen Protokollen kann man den Web-Service verwenden?

Beispiel WSDL



MICRO CONSULT

Web Services Description Tool innerhalb .NET

- Erzeugen von .NET Client Proxies
  - Komplette Proxyklasse
    (synchrone und asynchrone Aufrufe)
  - Notwendiger Code für Netzwerkkommunikation
- Quelldatei in angegebener Sprache

Verschiedene Eingabeparameter, u.a.

- URL einer WSDL Datei
- Implementierungssprache der Proxyklasse
- Aufruf-Protokoll
- Benutzername und Passwort bei Basic Authentication

Global und herstellerübergreifend

- UDDI
  - Universal Discovery, Description and Integration
  - Globales Web Services Verzeichnis

Lokal und proprietär (MS)

- DISCOvery “Protokoll” (Vorläufer von UDDI)
  - .vsdisco Dateien in Visual Studio .NET
  - myWS.asmx?disco
  - Mit disco.exe erzeugte Dateien
- XML Grammatik
  - Enthält Verweise zu WSDL Ressourcen
- Abgelegt im Server Rootverzeichnis

Universal Description, Discovery and Integration

Dient zur Registrierung von Web Services; zentrales Verzeichnis von Unternehmen und Web Services

UDDI API als Web Service nach aussen offengelegt

Globale Business-Registry:

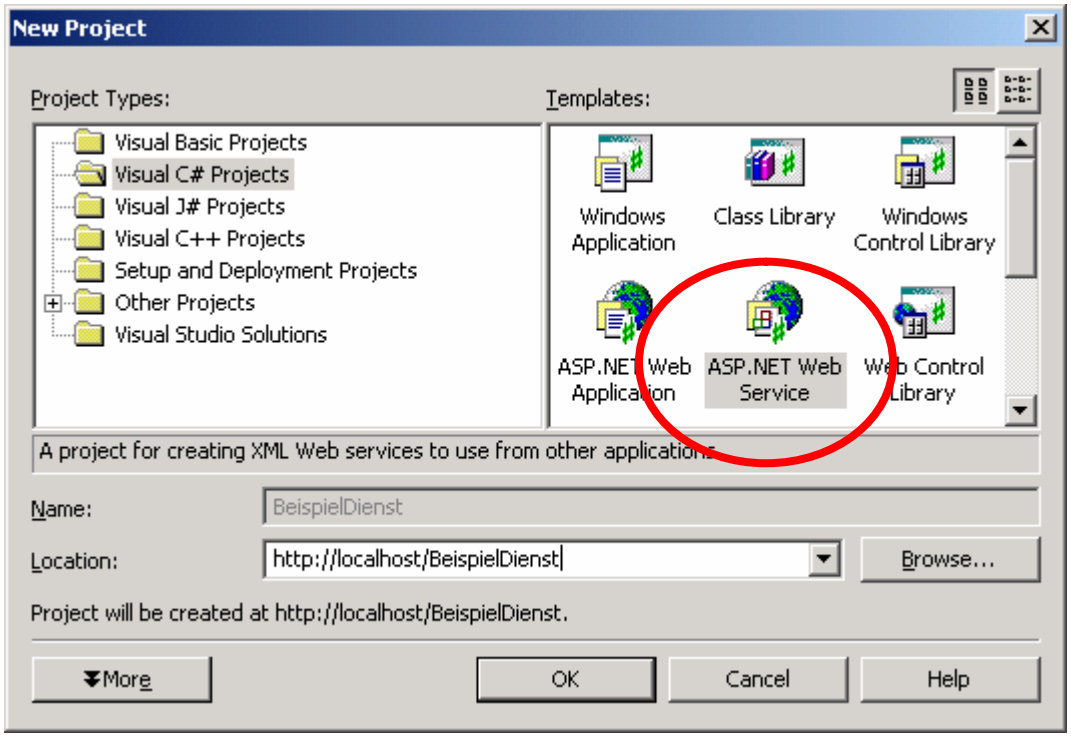
- Logisch zentral
- White-Pages:
  - Informationen zu den Anbietern (URL, Adresse, TelNr.)
- Yellow-Pages:
  - Branchenspezifische Suche nach Unternehmen (Branche, Land)
- Green-Pages:
  - Technische Beschreibung der Plattformen und Dienste eines Unternehmens

Weitere Informationen: http://www.uddi.org

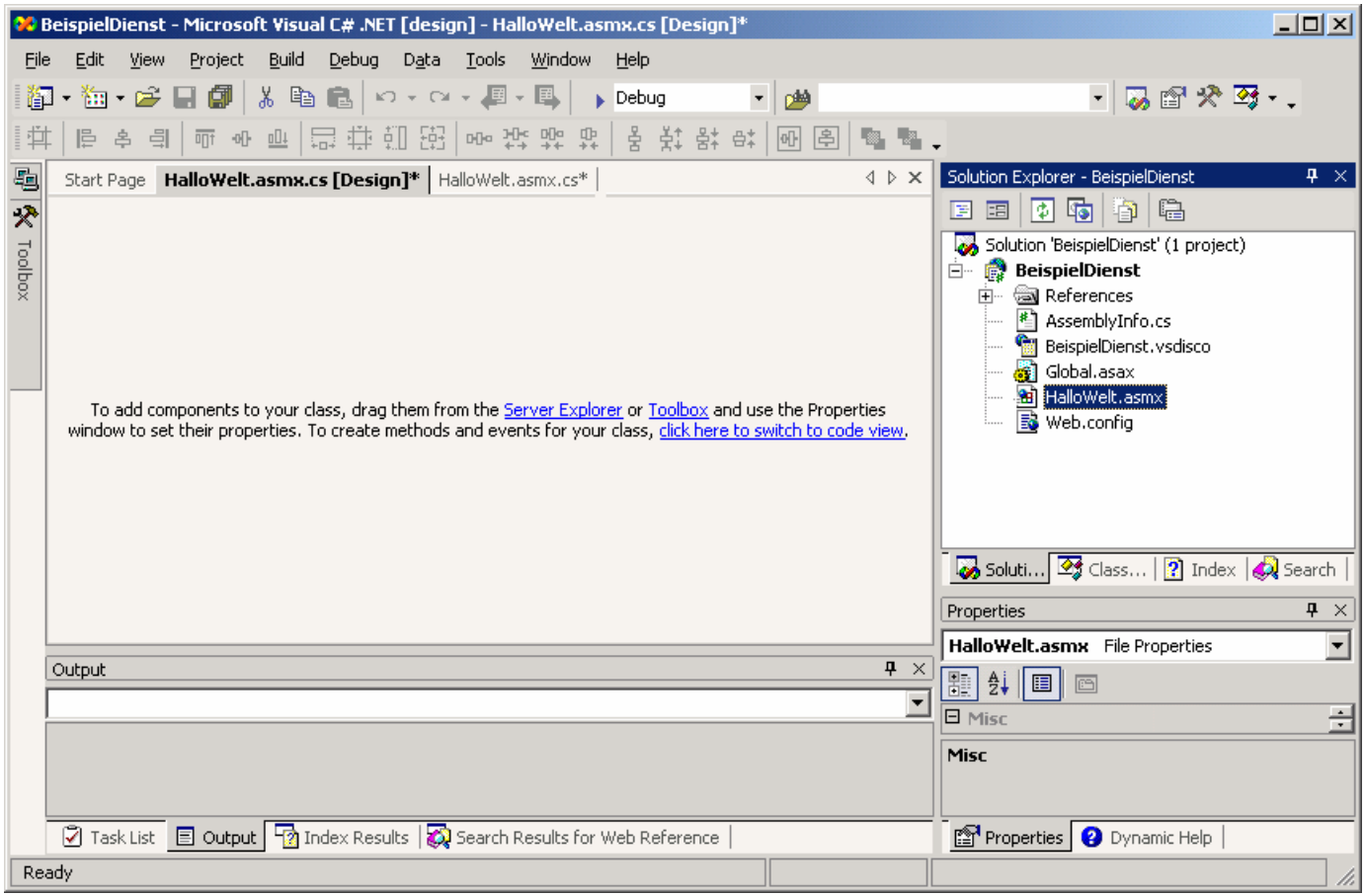
Webservices in .NET basieren auf offenen Standards

- Extensible Markup Language (XML)
  - grundlegender Datenbeschreibungsstandard auf der Präsentationsschicht
- Hypertext Transfer Protocol (HTTP)
  - Transportschicht-Standard
- Simple Object Access Protocol (SOAP)
  - Standardprotokoll zur Kommunikation zwischen Applikationen über das Web
- Web Services Description Language (WSDL)
  - Beschreibungssprache für Web Services

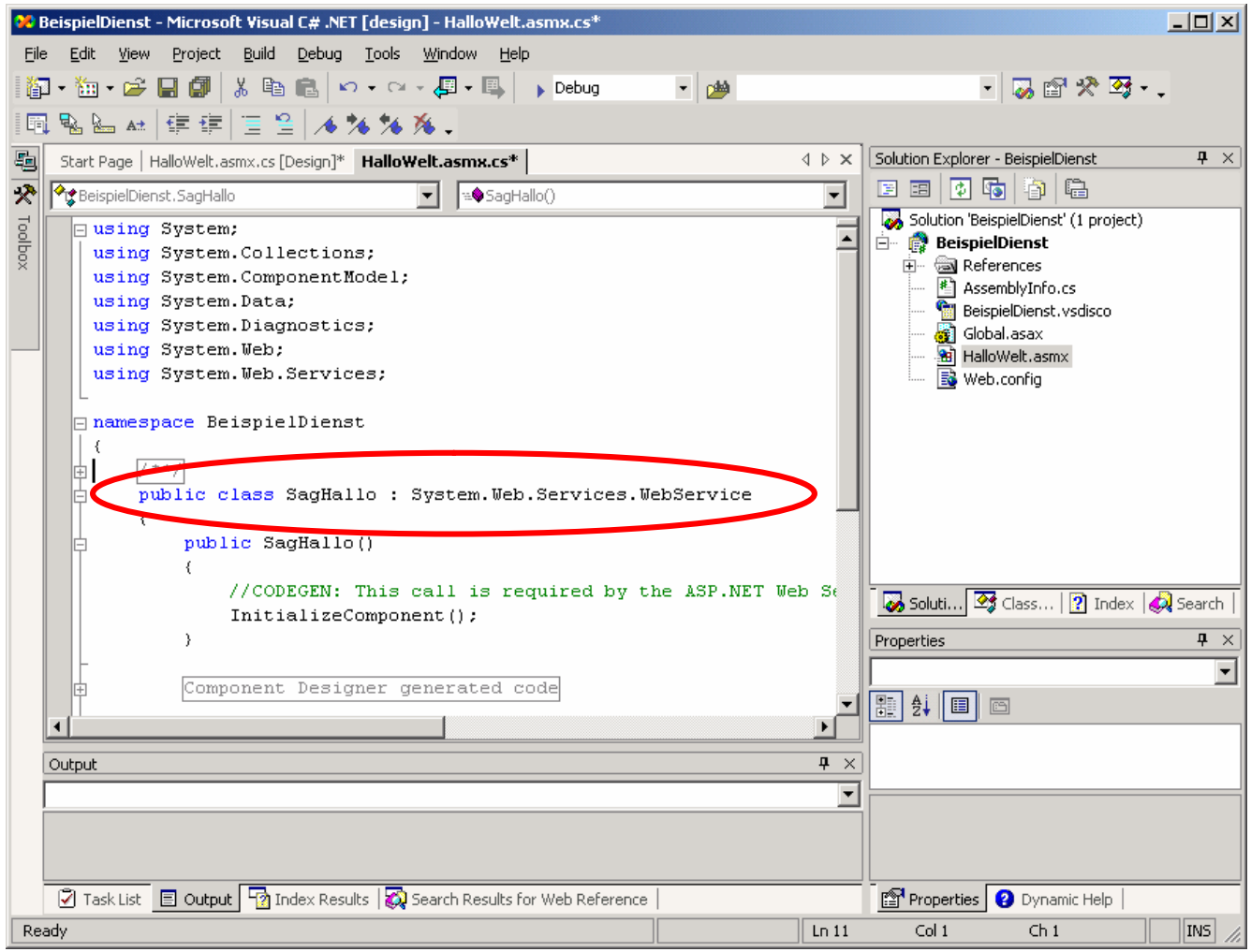
Erstellung eines einfachen WebService mit Visual Studio.NET



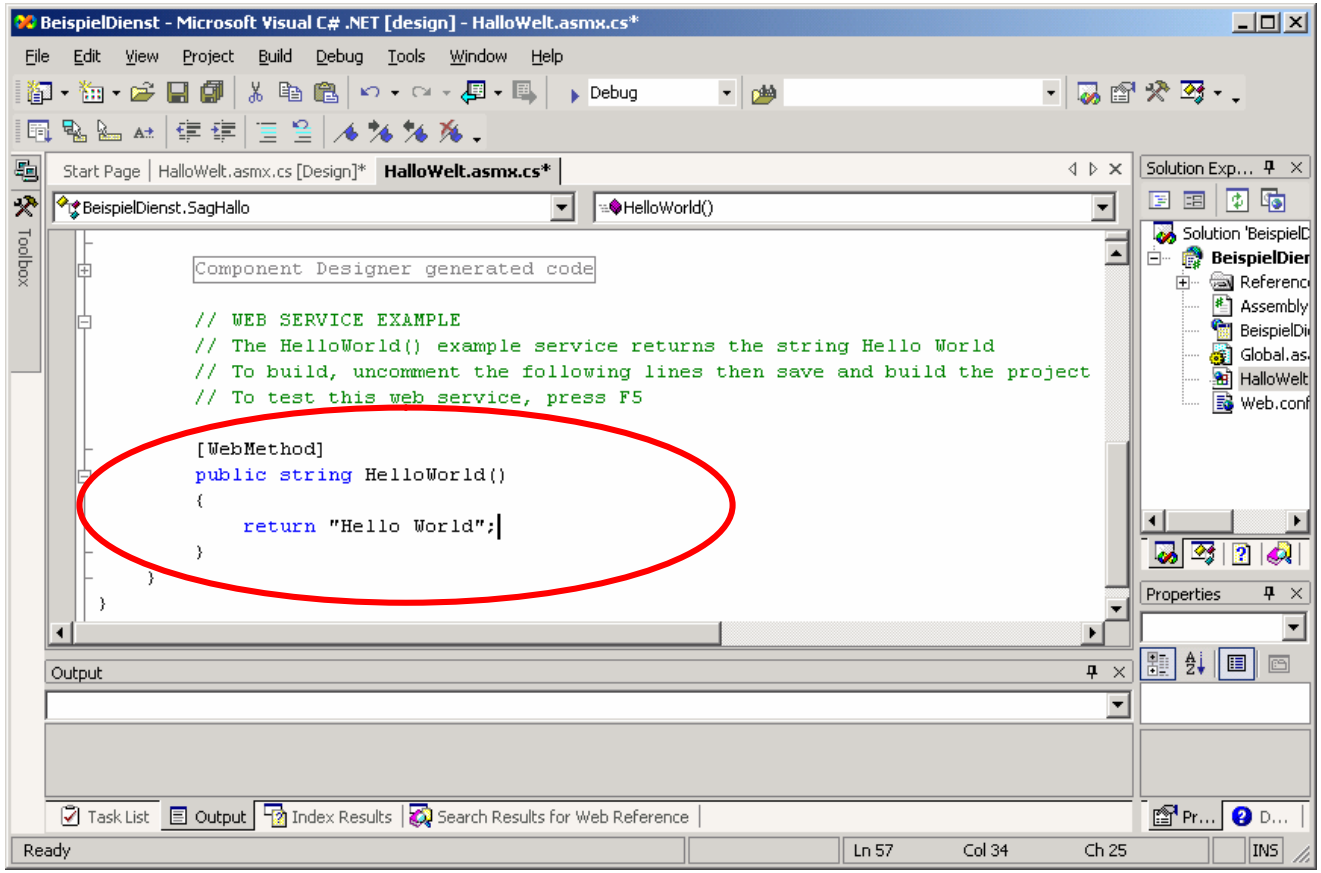
WebService Applikation :



WebService Applikation :

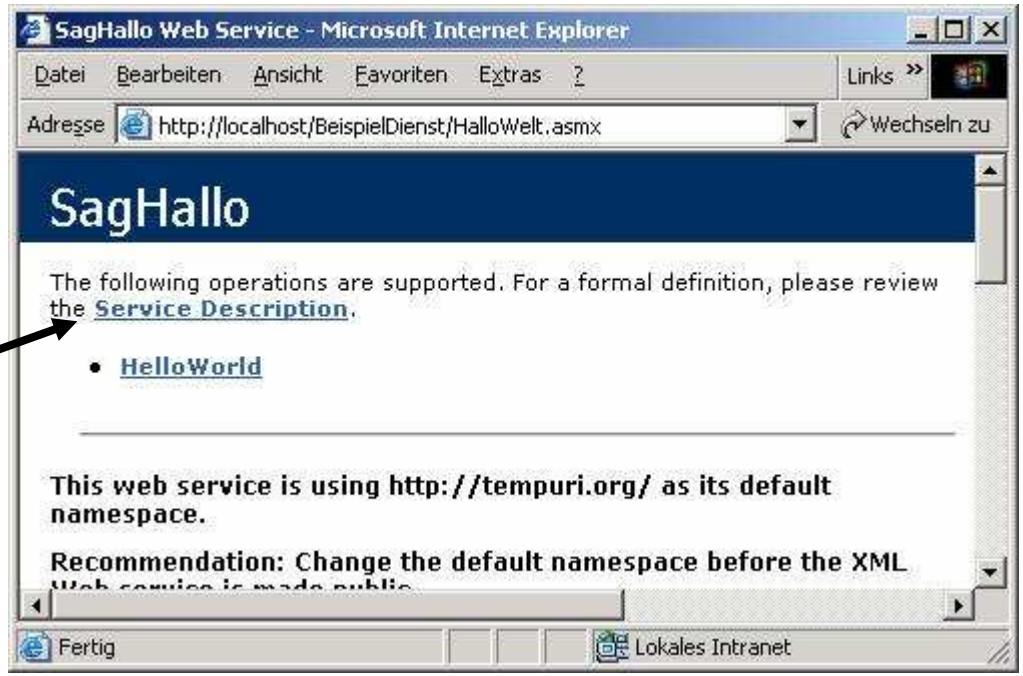


WebService Applikation :

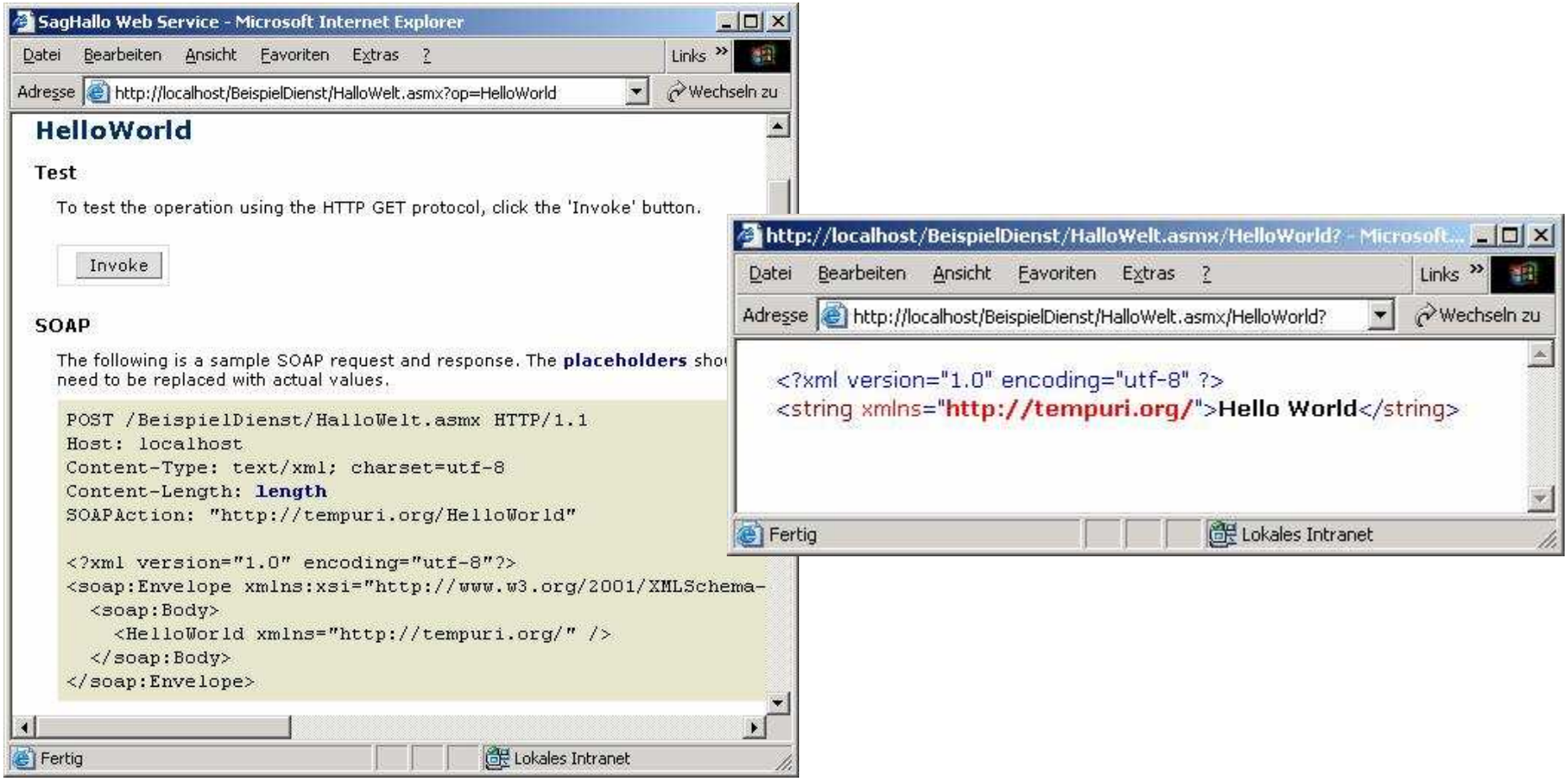


Lokaler Test

WSDL - Datei



Lokaler Test

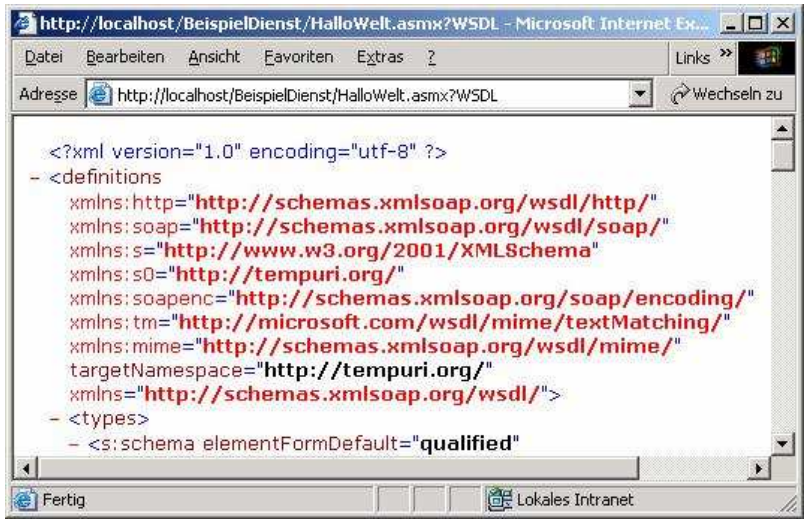


1. Schritt : Erstellung einer Proxyklasse unter Zuhilfenahme des .NET Tools „wsdl.exe“

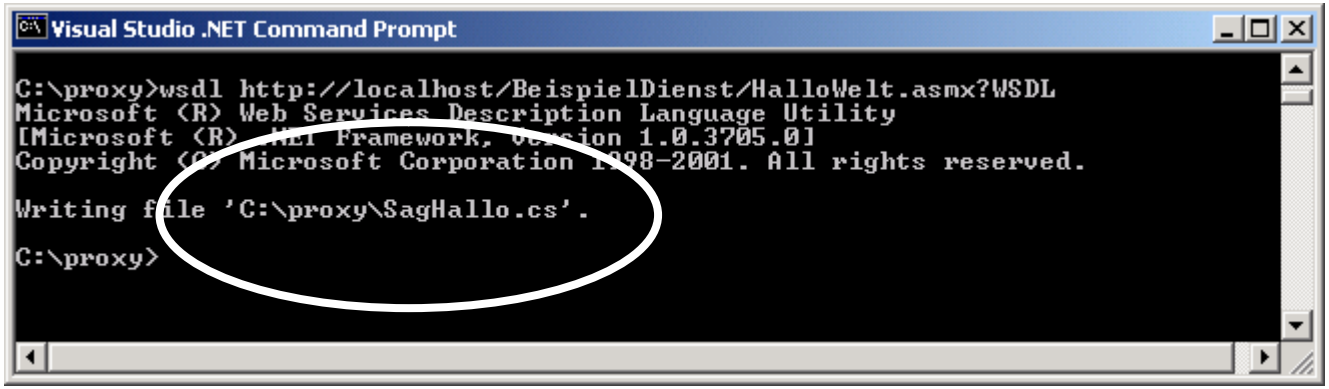
- **Beschreibung des zu nutzenden Webdienstes ist erreichbar unter:**

http://localhost/BeispielDienst/HalloWelt.asmx?WSDL

URL des Webdienstes

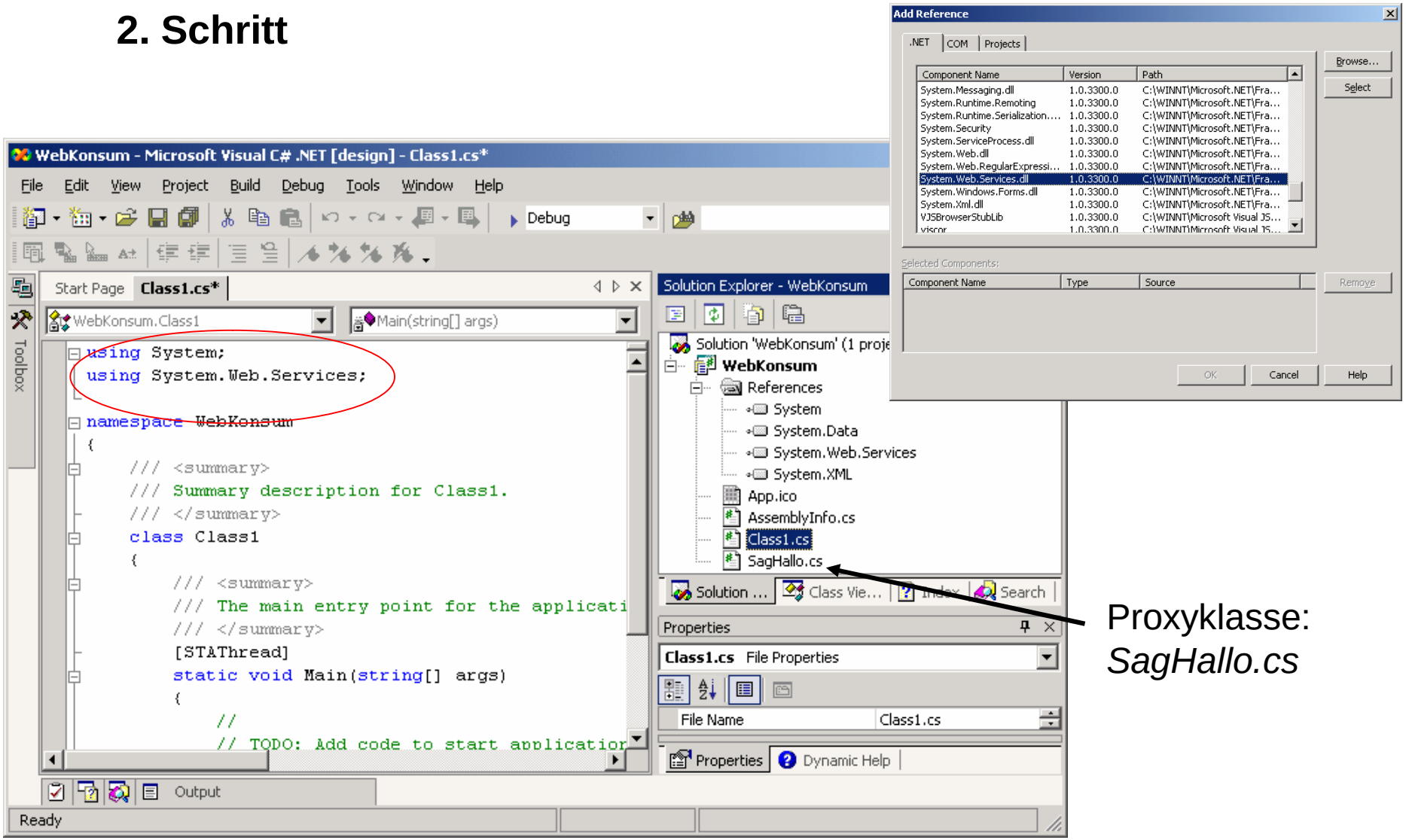


1. Schritt (Fortsetzung):



Übernahme dieser „Proxyklasse“ in ein aktuelles Projekt (C# Console Application)

2. Schritt



Proxyklasse:
SagHallo.cs

MICRO CONSULT

3. Schritt : Nutzung des Webdienstes