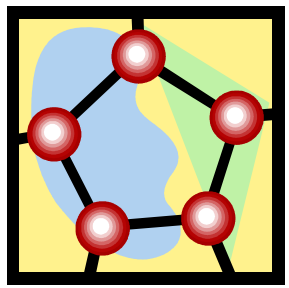
XML Web Services



Dietrich Birngruber

Software Architect
TechTalk
www.techtalk.at

Why XML Web Services?

TECH TALK
it works, we care

What is an XML Web Service?

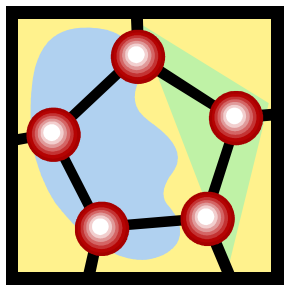
TECH TALK
it works, we care

XML Web Service =

a “web” of loosely coupled software services

+ programming language independent (client / server)

+ platform independent (runtime, OS)



→ .NET is one possible solution platform!

Independence, because of …

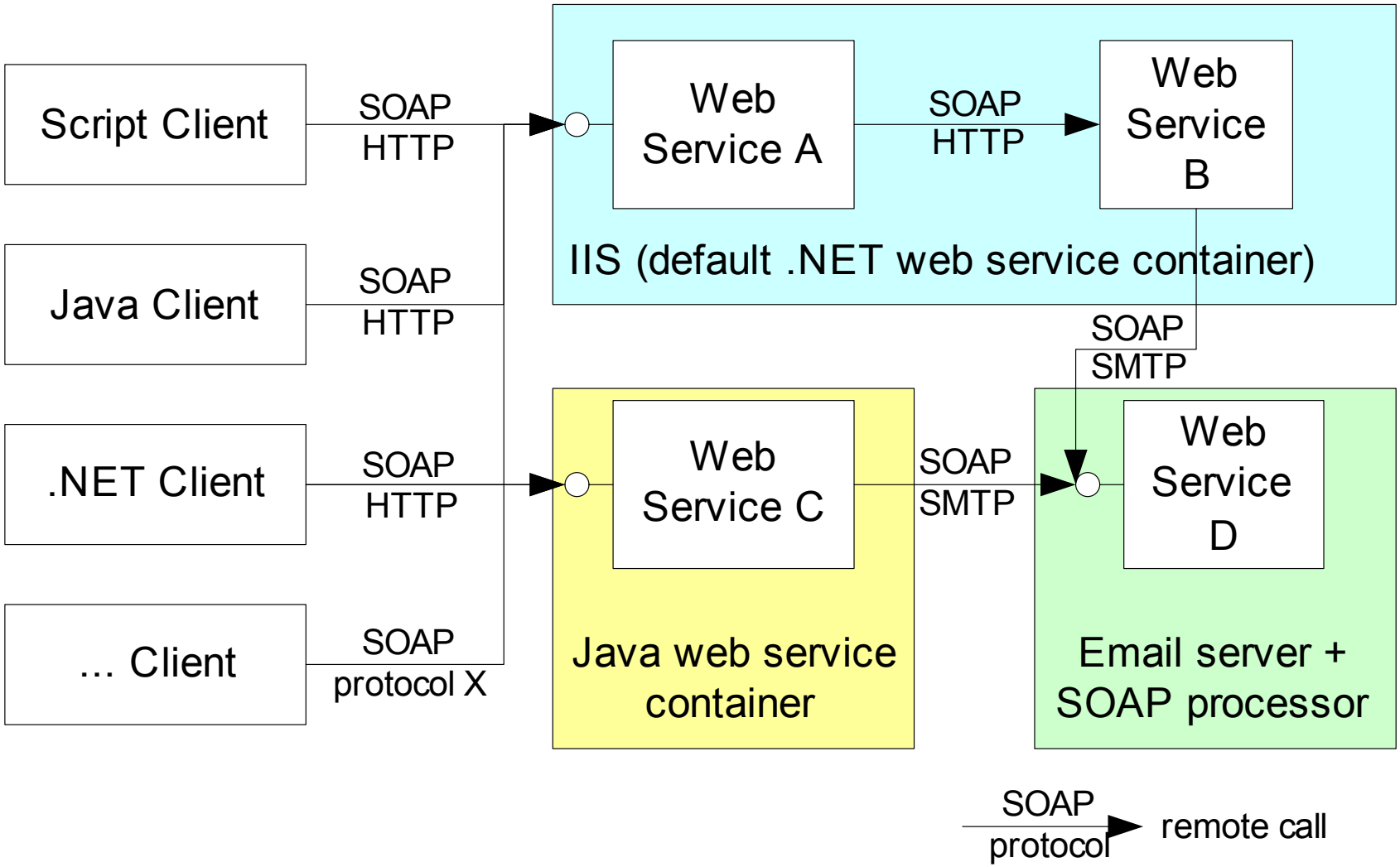
TECH TALK
it works, we care

- SOAP
  - "RPC" based on XML
  - Independent of transport protocol (e.g.: HTTP, SMTP, ...)
  - It's only text! Different implementations and locations:
    Java, .NET, Python, Delphi, ..., cell phone, PDA, workstation, ...
  - NO distributed object model
    (in contrast to CORBA, Java RMI, DCOM, ...)

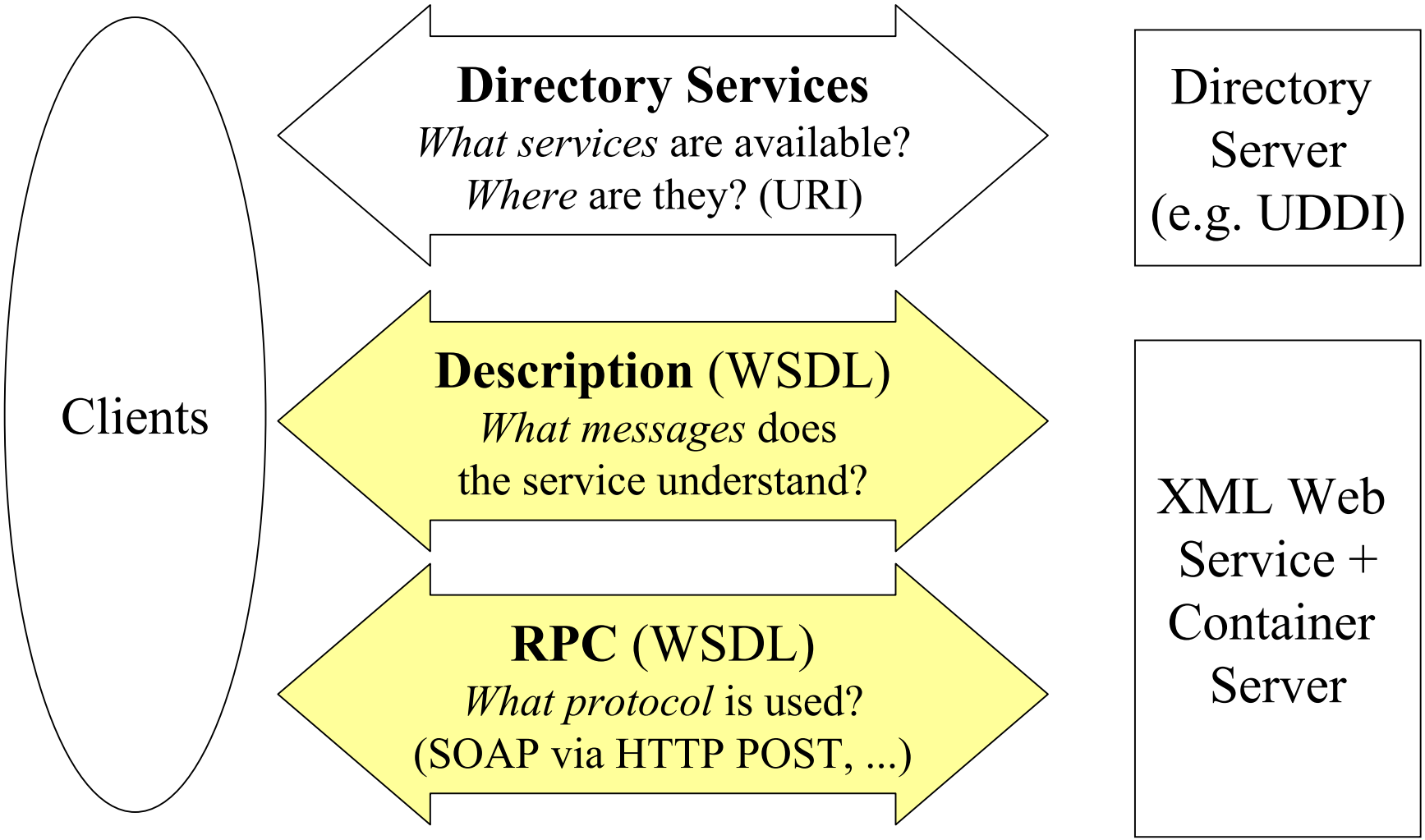
- Web Services Description Language - WSDL (1.1)
  - Service (or interface) description in XML

- Standardized (W3C)

Web Services Scenario



Common Web Service Infrastructure



Contents

TECH TALK
it works, we care

- **Example**
- SOAP
- WSDL
- Lookup of Web Services (Discovery)
- Resources

Example: TimeService

TECH TALK
it works, we care

MyFirstService.**asmx**

```
<%@ WebService Language="C#"
Class="TimeService" %>

using System.Web.Services;


public class TimeService : WebService {

  [WebMethod(Description="Returns the current time")]
  public string GetTime() {
    return System.DateTime.Now.ToLongTimeString();
  }

}
```

WebService directive

WebService namespace

Base class WebService

Attribute [WebMethod] declares a web service method that is callable via SOAP

Who compiles the .asmx file? Who executes our web service?

Example: Simple Client in C#

TECH TALK
it works, we care

- wsdl.exe generates proxy classes (*TimeClient*.TimeServie)

  wsdl.exe /namespace:*TimeClient* /out:TimeServiceProxy.cs
  http://localhost/netsem-ws/MyFirstService.asmx

```
using System;
using TimeClient; // namespace of the generated proxy

public class NetClient {
   public static void Main(string[] args) {
      TimeService service = new TimeService();
      Console.WriteLine("Server time: ");
      string time = service.GetTime();
      Console.WriteLine(time);
   }
}
```

Example: Simple Client in Java

TECH TALK
it works, we care

- Develop proxy by hand, or
- Use a tool and library
  - E.g.: GLUE
    wsdl2Java generates a Java interface (ITimeServiceSoap) and an implementation class (TimeServiceHelper)

```
import Kapitel7.GLUEProxy.*; // package of the generated proxy classes
/** Simple client in Java which uses GLUE for accessing web services */
public class JavaClientGLUE {
  public static void main(String[] args) {
    try {
      //use Java interface and proxy class – generated by wsdl2Java
      ITimeServiceSoap service = TimeServiceHelper.bind();
      String time = service.GetTime();
      System.out.println("Server time: \n" + time);
    } catch (Exception ex) {  ex.printStackTrace(); }
  }
}
```

Contents

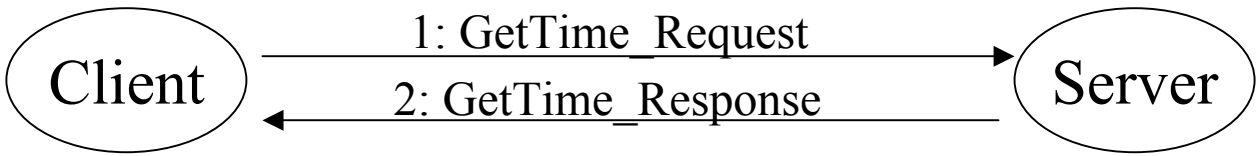
TECH TALK
it works, we care

- ☑ Example
- ☐ **SOAP**
- ☐ WSDL
- ☐ Lookup of Web Services (Discovery)
- ☐ Resources

SOAP

TECH TALK
it works, we care

- Independent of a physical transport protocol
  - E.g. HTTP, which is request / response oriented
- Asynchronous, "one-way" protocol in XML
- Least common denominator is ***one message***
  - Synchronous method call: combines two messages



- SOAP **does not** define a
  - Distributed object model
  - Distributed garbage collection mechanism
  - Distributed callback mechanism (no events)

XML Layout (simplified, v1.2)

TECH TALK
it works, we care

```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="...">
  <soap:Header> <!-- (optional and extendable)  -->
    <m:my xmlns:m="anURI" soap:mustUnderstand="true" soap:role="uri2" />
    ...
  </soap:Header>

  <soap:Body>
    data (depends on encoding and format) or Fault element
    <soap:Fault>
      <soap:Code>...who is responsible?... </Code>
      <soap:Reason>...textual description...</soap:Reason>
      <soap:Detail>...more error details...</soap:Detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Encoding of the Data in the <Body> Element

TECH TALK
it works, we care

- How does the XML structure look like?
  - document (based on an *XML schema*)
  - rpc (based on the *SOAP specification*)
- How are the actual parameters encoded? (e.g. array)
  - literal (based on an *XML schema*)
  - encoded (based on the SOAP or custom encoding rules)
- Typical combination:
  - *document/literal* (default value in .NET) or
  - *rpc/encoded* (default value for Java server)

SOAP-HTTP Binding

TECH TALK
it works, we care

- HTTP GET
  - Sent data is URL encoded and size is limited
  - Example: type in a URL in a web browser
    http://localhost/netsem-ws/MyFirstService.asmx/GetTime?
- HTTP POST
  - Sent data is URL encoded, but size of the data is not limited

- SOAP via HTTP POST
  - Data is SOAP encoded (document/literal, rpc/encoded, ...)
  - No restrictions from URL encoding!

Useful .NET Namespaces

TECH TALK
it works, we care

- ***System.Web.Services***
  - for creating web services (e.g. WebService, WebMethod)
- *System.Web.Services.Configuration*
  - for extending SOAP
- *System.Web.Services.Description*
  - for manipulating WSDL
- *System.Web.Services.Discovery*
  - for using DISCO
- *System.Web.Services.Protocols*
  - Implementation of communication protocols (e.g. SOAP-HTTP)
- *System.Xml.Serialization*

What Datatypes Are Supported?

TECH TALK
it works, we care

- classes, interfaces (e.g. DataSet)
- arrays
- structs
- enumerations
- primitive datatypes (int, double, ...)
- ...

- What ever you can define in an XML schema!
- Default and custom mapping of XML to runtime types in .NET

Example: TimeService2

- rpc/encoded
  - Attribute [SoapRpcService]
- WSDL of the service defines its own XML namespace
  - Attribute [WebService]
- Method *GetTimeDesc* has user defined datatype
  - Encode fields of class TimeDesc as XML *attributes* (not elements)

```
public struct TimeDesc {
   [SoapAttribute] public string TimeLong;
   [SoapAttribute] public string TimeShort;
   [SoapAttribute (AttributeName = "ZoneID")] public int TimeZone;
}
```


```
<types:TimeDesc id="id1" xsi:type="types:TimeDesc"
types:TimeLong="10:00:25" types:TimeShort="10:00" types:ZoneID="1" />
```

Example: TimeService2

TECH TALK
it works, we care

```
<%@ WebService Language="C#" Class="Kapitel7.TimeService" %>
using System;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml.Serialization;

namespace Kapitel7 {
  [SoapRpcService]
  [WebService(Namespace="http://dotnet.jku.at/time/",Description="Returns the time")]
  public class TimeService : WebService {
    // ... andere Methoden
    [WebMethod(Description="Returns the time description of the server")]
    public TimeDesc GetTimeDesc() {
      TimeDesc td = new TimeDesc();
      // ...
      return td;
    }
  }
  public struct TimeDesc { ...  }
}
```

Contents

TECH TALK
it works, we care

- ☑ Example
- ☑ SOAP
- ☐ **WSDL**
- ☐ Lookup of Web Services (Discovery)
- ☐ Resources

WSDL (1.1) Gives the Answers to ...

- What „methods“ does the service offer?
- What ports, protocols and messages does the service offer in order to call its methods?
- What name and parameters does a message consist of?
- What data types are used?

Simplified WSDL Layout

TECH TALK
it works, we care

<definitions>

<**types**/>	... describes the datatypes in XML
<**message**/>	... describes messages (name and parameter)
<**portType**> <**operation**> <**input**/> <**output**/>	... describes the callable messages for each protocol. The messages are grouped togoether to an operation. Thus, this elements describe these operations which can be reached via this "port".
<**binding**> <**operation**/>	... binds a protocol to a port and defines the used encoding for each operation (e.g. SOAP, rpc/encoded).
<**service**> <**port**/>	... describes the identification of a service: name, used bindings and the URI under which the service can be reached

</definitions>

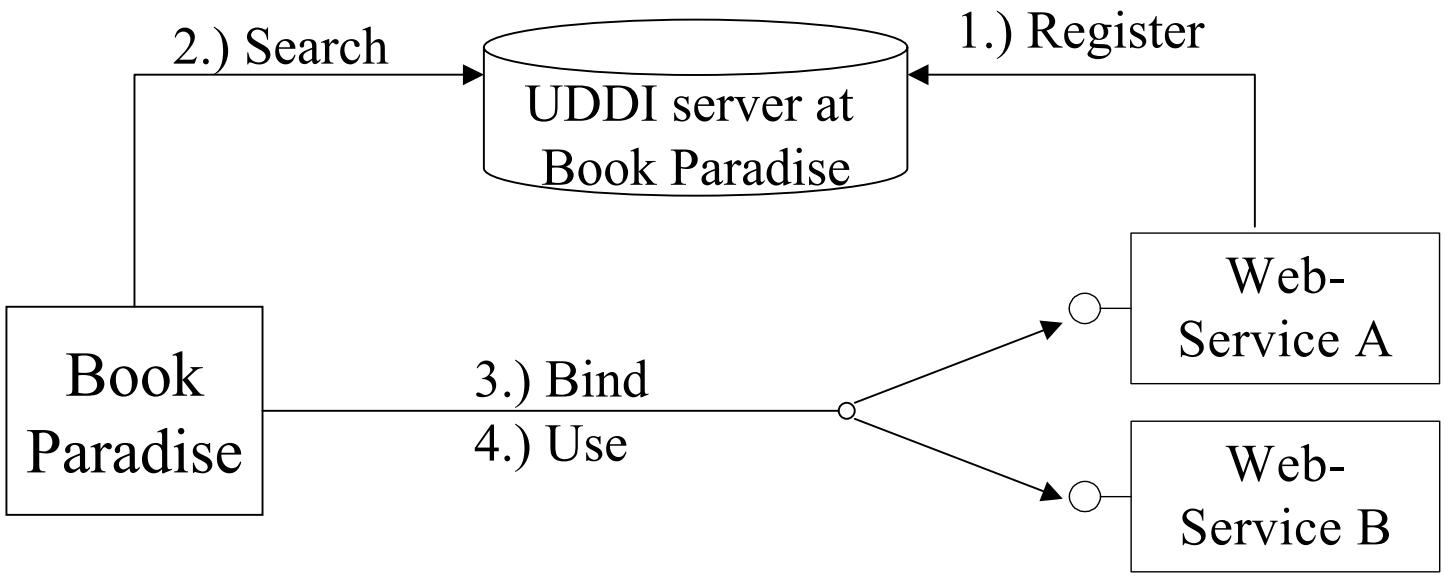
Web Service Discovery Infrastructure

TECH TALK
it works, we care

- Universal, Description, Discovery and Integration (UDDI)
  - "Yellow Pages": industry, companies, web services …
  - For software not humans
  - Scope:
    - Within a company (main successful scenario)
    - "Global Registry", everyone is invited (business case????)
  - Platform independent: Standardized, Web Service Facade
    - www.uddi.org
    - IBM, Microsoft, Ariba, Sun, Oracle, Intel, SAP, Boeing, ...
- DISCO
  - XML document containing URIs to a particular service and WSDL
  - Possible result of a UDDI query
- ...

Scenario: Online Book Store

- "Book Paradise":
  - Online shop, simple user interface
- Publisher A, Publisher B:
  - Publish their product catalog as a web service
  - **Precondition: business contract with Book Paradise!!!**



Resources (besides dotnet.jku.at ☺)

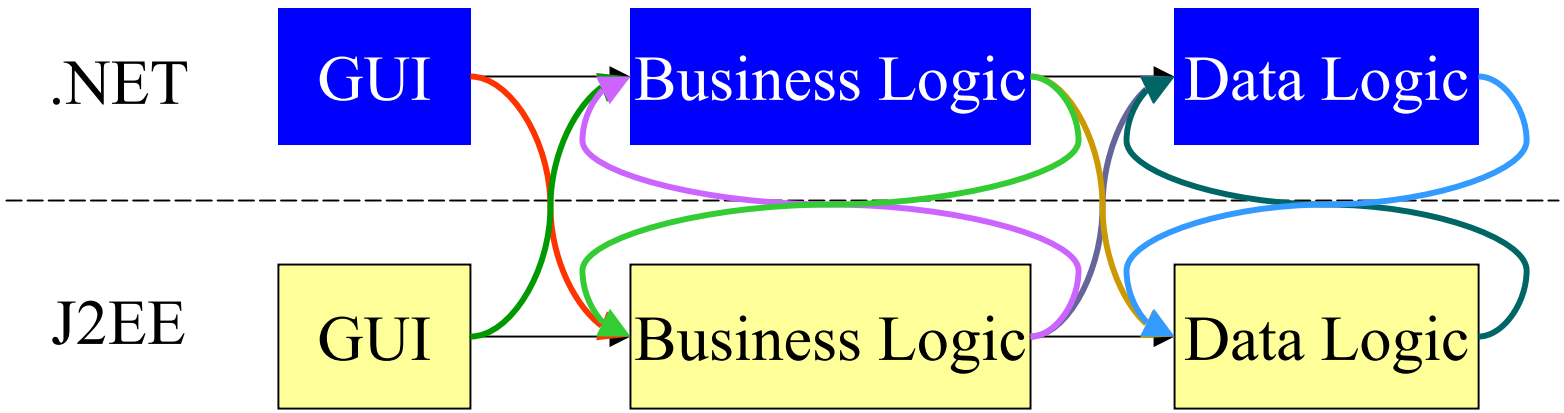
TECH TALK
it works, we care

- UDDI & Co
  - www.uddi.org,
    www-3.ibm.com/services/uddi,
    uddi.microsoft.com
  - www.xmethods.com
- Tutorials, Beispiele, Forum, ...
  - www.webservices.org
  - www.gotdotnet.com
  - groups.yahoo.com/group/soapbuilders
- Java-Implementierungen
  - xml.apache.org/axis/
  - www.themindelectric.com

Summary

- SOAP, WSDL are standardized
- Main usage scenario: EAI (Enterprise Application Integration)

  e.g.:



- New standards: security, transactions, …
- If you control client and server: **binary protocol!**

Questions + Contact

TECH TALK
it works, we care

http://dotnet.jku.at

http://www.ssw.uni-linz.ac.at

http://www.techtalk.at

birngruber@acm.org