



## Web Services

### Der Business Case



# Agenda

- Die Schlagzeilen
- Die Lügen
- Die Bugwelle – der Hyper Cycle (Gardner)
- Ein Beispiel
- Web Services als Business
- Web Services als Baustein in einer IT Architektur
- Web Services als Integrationskonzept
- Sicherheit und Web Services



## Die Schlagzeilen

- *„Programm-Häppchen setzen sich durch“*
  - „Dem Durchbruch steht nichts im Weg“  
Computerworld 23. Januar 2004
- *„Webservices werden Chefsache“*  
Computerworld 23. Januar 2004
- *„Webservices verbinden Java und Dotnet“*  
Computerworld 5 Dezember 2003
- *„Gartner Surveys Show Web Services Are Entering the Mainstream“*  
Gartner First Take 24 April 2003

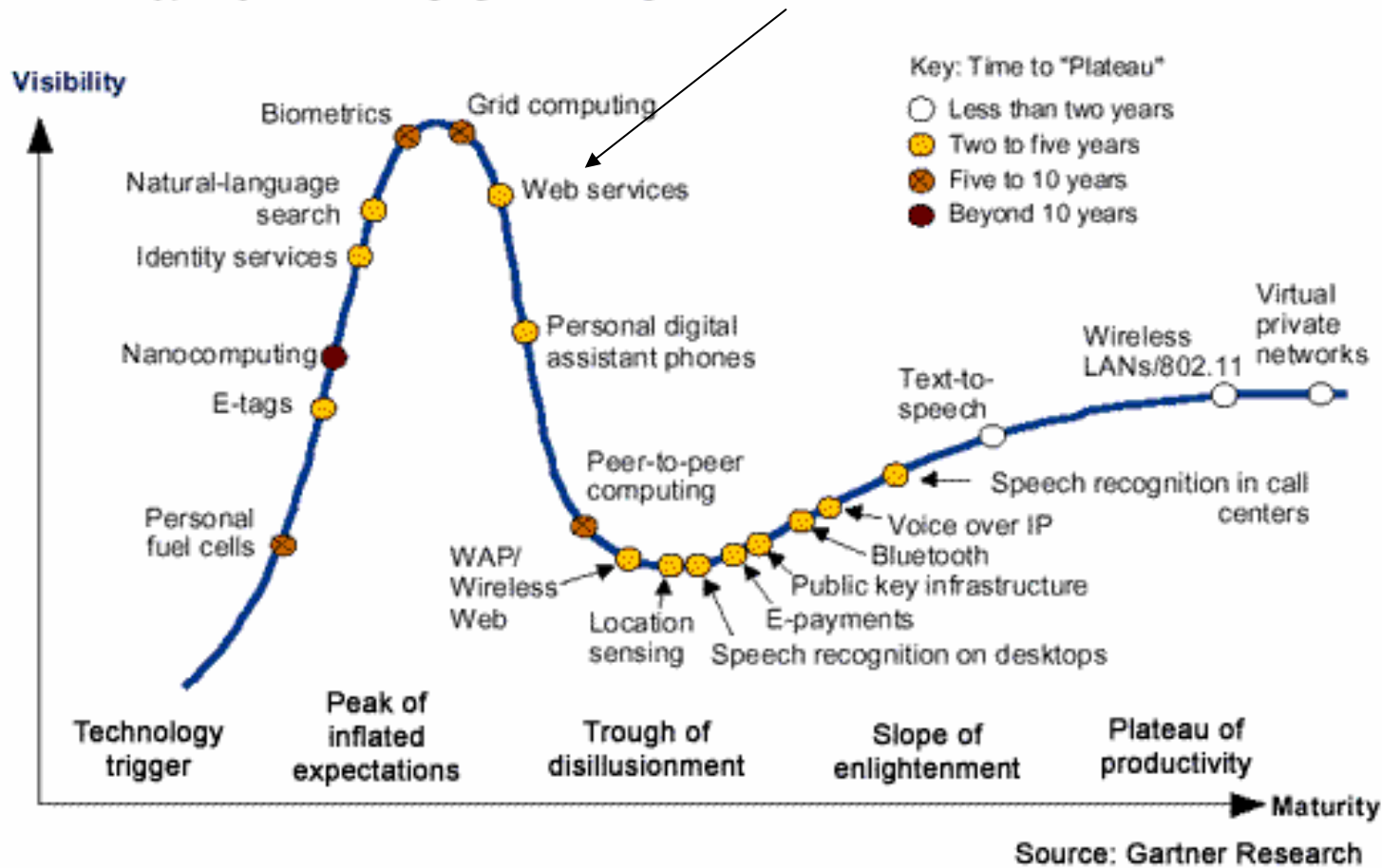


## Die Lügen

- Webservices sind nur für kleine Datenmengen sinnvoll
- Webservices sind unsicher
- Webservices sind langsam
- Es ist noch zu früh, Webservices einzusetzen

# Die Bugwelle – 2002!

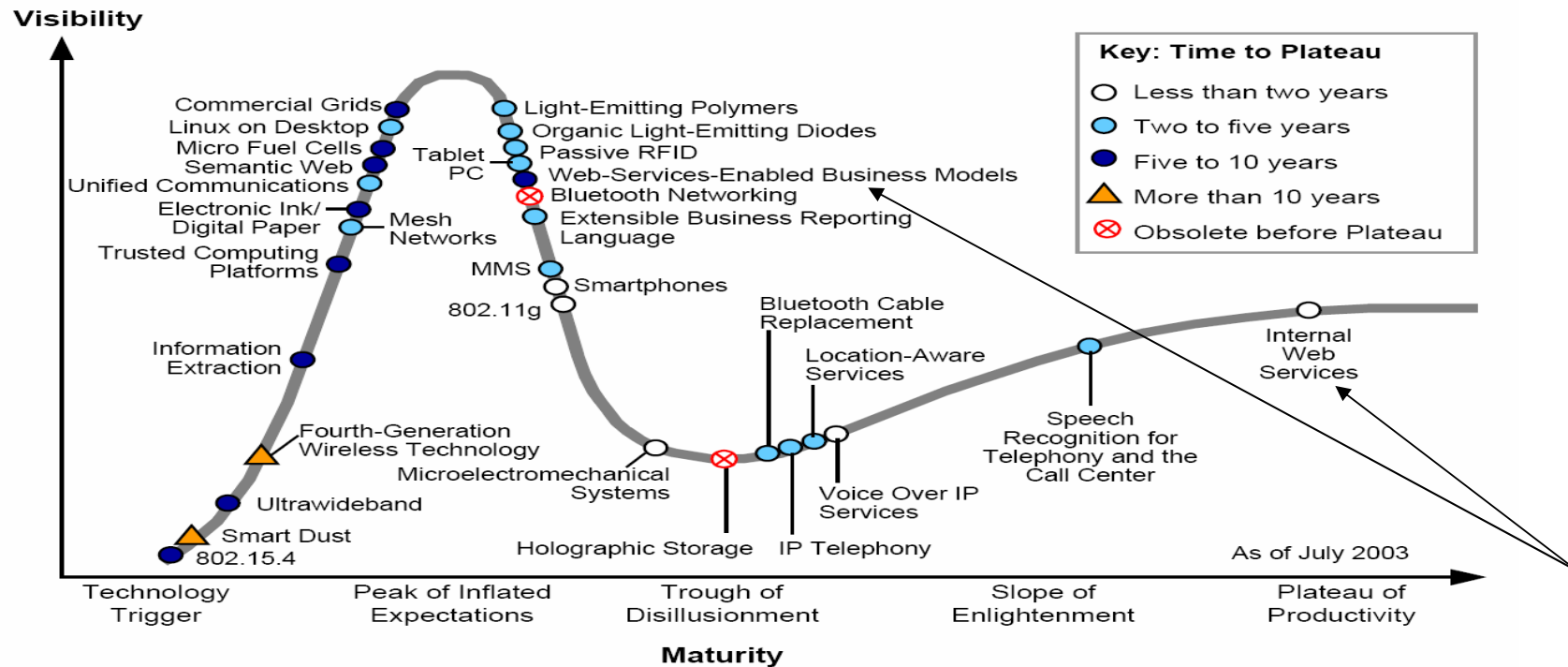
The 2002 Hype Cycle of Emerging Technologies



# Die Bugwelle – 2003!

## Hype Cycle for Emerging Technologies, 2003

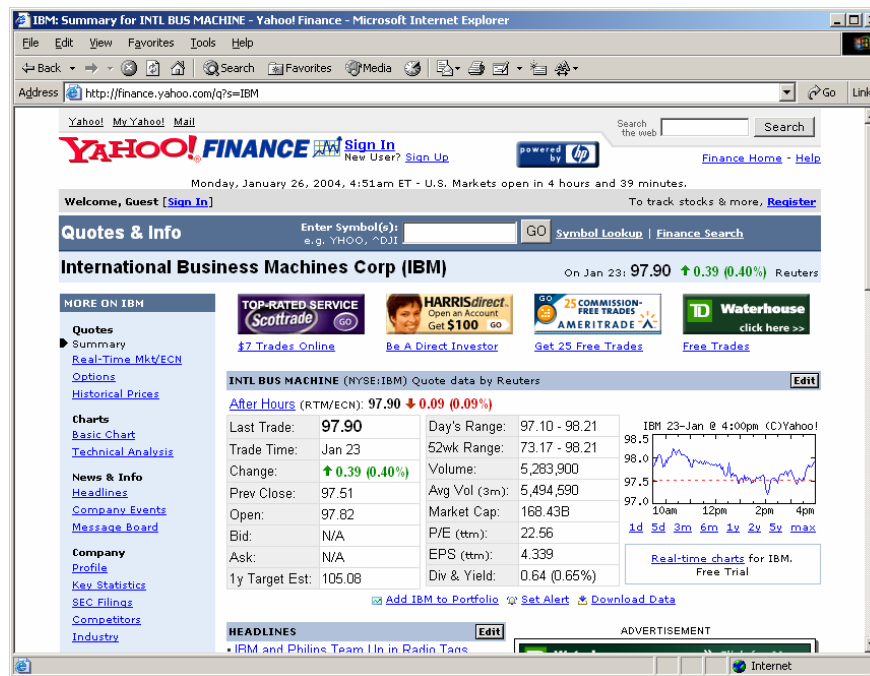
### 1.0 The Hype Cycle



Source: Gartner Research (July 2003)

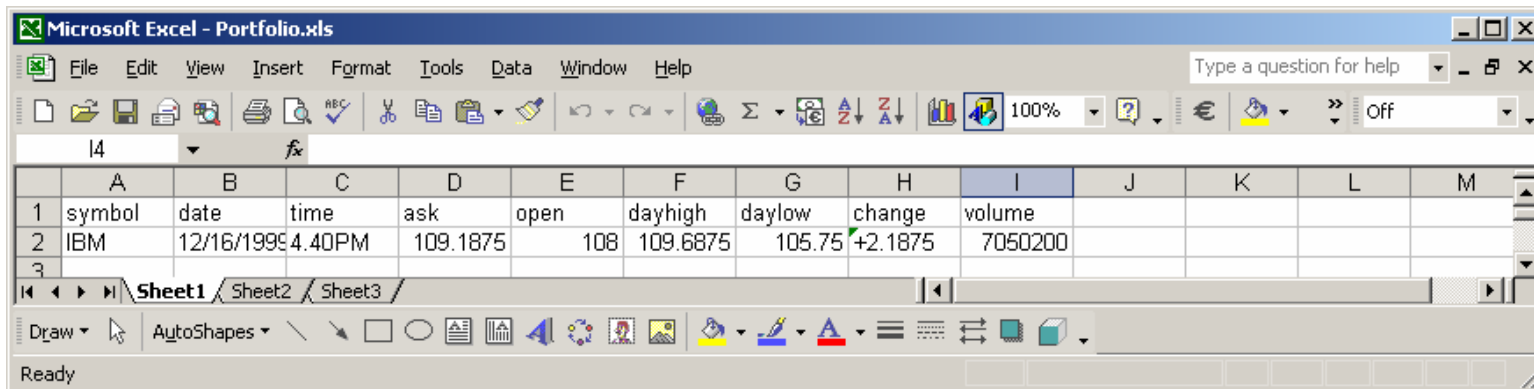
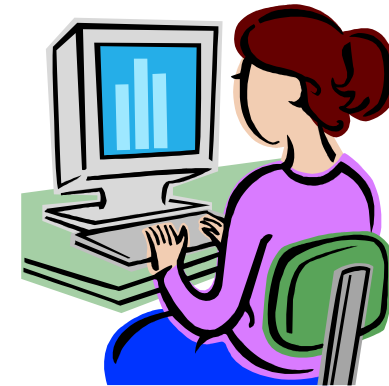
# Ein einfaches Beispiel – externe Information

- Phase 1
  - Frau Schweizer arbeitet mit ihrem Geld.
  - Sie kaufte einige Aktien und verfolgt die Börsenkurse täglich am Bildschirm



# Ein einfaches Beispiel – externe Information

- Phase 2
  - Frau Schweizer arbeitet *intensiver* mit ihrem Geld.
    - Sie verwaltet ihr Aktien-Portfolio mit Excel damit sie dieses jederzeit bewerten und einfache Szenarien durchrechnen kann
    - *Diese Lösung ist preiswert, schnell, einfach, voll unter Kontrolle*
    - *Frau Schweizer tippt die aktuellen Daten von Hand in ihr Spreadsheet ein*



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	symbol	date	time	ask	open	dayhigh	daylow	change	volume				
2	IBM	12/16/1999	4.40PM	109.1875	108	109.6875	105.75	+2.1875	7050200				



## Ein einfaches Beispiel – externe Information

- Phase 3
  - Frau Schweizer arbeitet *nur noch* mit ihrem Geld.
    - *Das Eintippen der Kursinformationen Hand in ihr Spreadsheet wird zu umständlich und zeitaufwendig*
  - Frau Schweizer wird eine Integration mithilfe eines *Webservices* angeboten
    - Die Kursinformation wird (mit 20 Minuten Verspätung) von einem *Provider* bezogen,  
  
der die Informationen als *XML Nachricht* liefert  
  
in *SOAP* verpackt  
  
über *http* transportiert  
  
mittels JDBC – ODBC  
wird diese lokal ins Excel geliefert



# Ein einfaches Beispiel – externe Information

- Phase 3
  - **Das XML Dokument**

```
<stock_quotes>
  <stock_quote>
    <symbol>IBM</symbol>
    <when>
      <date>12/16/1999</date>
      <time>4:40PM</time>
    </when>
    <price type="ask" value="109.1875"/>
    <price type="open" value="108"/>
    <price type="dayhigh" value="109.6875"/>
    <price type="daylow" value="105.75"/>
    <change>+2.1875</change>
    <volume>7050200</volume>
  </stock_quote>
</stock_quotes>
```



# Ein einfaches Beispiel – externe Information

- Phase 3
  - **Der SOAP Umschlag**

*Nachricht zum Service*

```
<soapenv:Envelope xmlns:soapenv=„...“>  
  <soapenv:Body>  
    <ns1:getQuote soapenv:encodingStyle=„...“>  
      <symbol xsi:type=„...“>UBS</symbol>  
      ...  
    </ns1:getQuote>  
  </soapenv:Body>  
</soapenv:Envelope>
```



# Ein einfaches Beispiel – externe Information

- Phase 3
  - **Technische Skizze**
    - Einbindung der externen Information in die internen Informationslogistik
    - In der Regel:
      - Kostenpflichtige Echtzeitinformationen
    - Eventuell:
      - „Proxy“ Server  
Ein Access Point auf die Echtzeitinformation  
Internes Verteilen der Information  
(„Just in Time“ Informationskonzept)

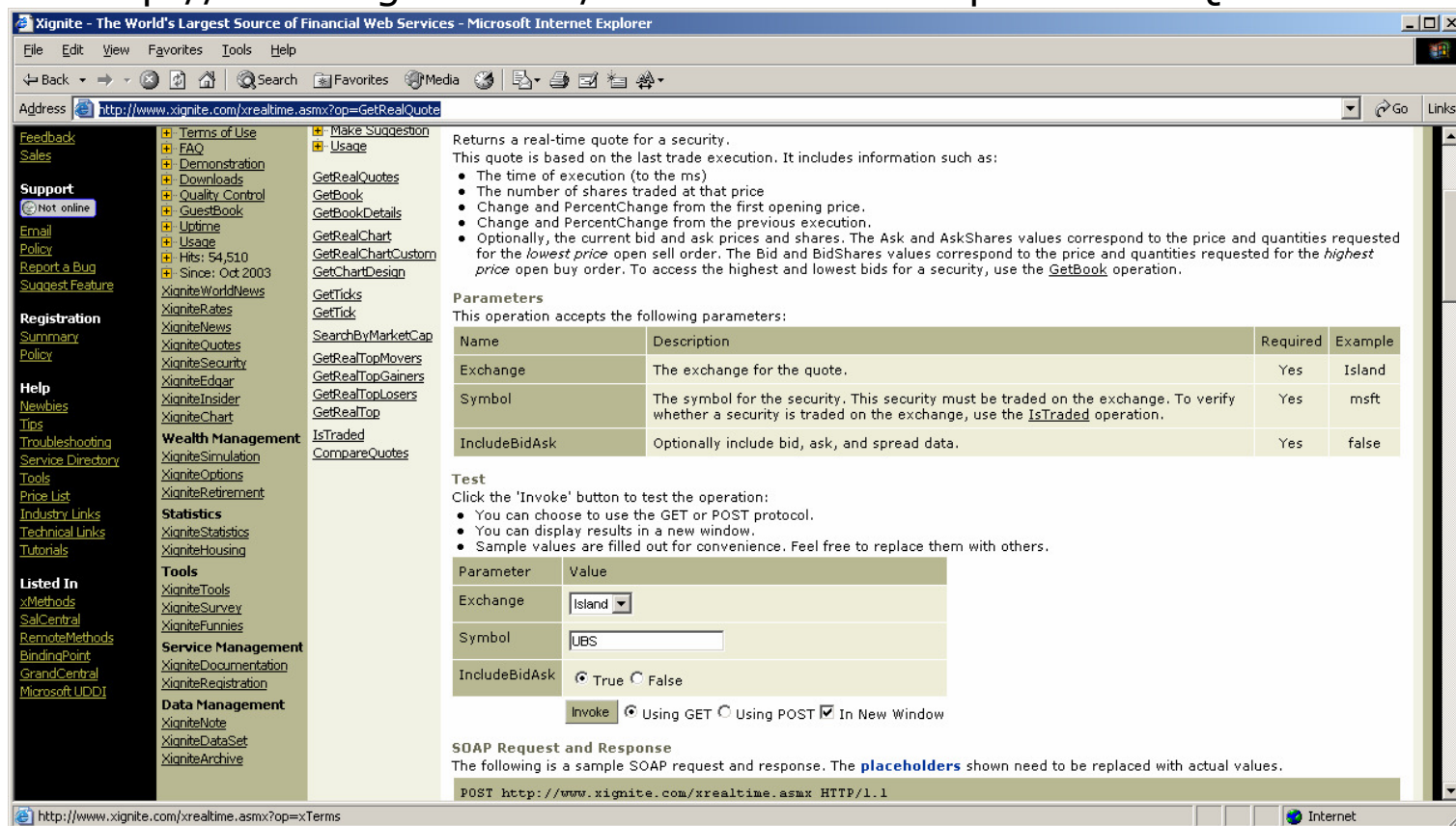


# Ein einfaches Beispiel – externe Information



- Phase 3
  - Demo

- <http://www.xignite.com/xrealtime.asmx?op=GetRealQuote>



**Xignite - The World's Largest Source of Financial Web Services - Microsoft Internet Explorer**

Address: <http://www.xignite.com/xrealtime.asmx?op=GetRealQuote>

Feedback  
Sales  
Support  
Not online  
Email  
Policy  
Report a Bug  
Suggest Feature  
Registration  
Summary  
Policy  
Help  
Newbies  
Tips  
Troubleshooting  
Service Directory  
Tools  
Price List  
Industry Links  
Technical Links  
Tutorials  
Listed In  
xMethods  
SalCentral  
XigniteFunnies  
RemoteMethods  
BindingPoint  
GrandCentral  
Microsoft UDDI

Terms of Use  
FAQ  
Demonstration  
Downloads  
Quality Control  
GuestBook  
Uptime  
Usage  
Hits: 94,510  
Since: Oct 2003  
XigniteWorldNews  
XigniteRates  
XigniteNews  
XigniteQuotes  
XigniteSecurity  
XigniteEdgar  
XigniteInsider  
XigniteChart  
Wealth Management  
XigniteSimulation  
XigniteOptions  
XigniteRetirement  
Statistics  
XigniteStatistics  
XigniteHousing  
Tools  
XigniteTools  
XigniteSurvey  
XigniteFunnies  
Service Management  
XigniteDocumentation  
XigniteRegistration  
Data Management  
XigniteNote  
XigniteDataSet  
XigniteArchive

Make Suggestion  
Usage  
GetRealQuotes  
GetBook  
GetBookDetails  
GetRealChart  
GetRealChartCustom  
GetChartDesign  
GetTicks  
GetTick  
SearchByMarketCap  
GetRealTopMovers  
GetRealTopGainers  
GetRealTopLosers  
GetRealTop  
IsTraded  
CompareQuotes

Returns a real-time quote for a security.  
This quote is based on the last trade execution. It includes information such as:

- The time of execution (to the ms)
- The number of shares traded at that price
- Change and PercentChange from the first opening price.
- Change and PercentChange from the previous execution.
- Optionally, the current bid and ask prices and shares. The Ask and AskShares values correspond to the price and quantities requested for the *lowest price* open sell order. The Bid and BidShares values correspond to the price and quantities requested for the *highest price* open buy order. To access the highest and lowest bids for a security, use the [GetBook](#) operation.

Parameters  
This operation accepts the following parameters:

Name	Description	Required	Example
Exchange	The exchange for the quote.	Yes	Island
Symbol	The symbol for the security. This security must be traded on the exchange. To verify whether a security is traded on the exchange, use the <a href="#">IsTraded</a> operation.	Yes	msft
IncludeBidAsk	Optionally include bid, ask, and spread data.	Yes	false

Test  
Click the 'Invoke' button to test the operation:

- You can choose to use the GET or POST protocol.
- You can display results in a new window.
- Sample values are filled out for convenience. Feel free to replace them with others.

Parameter	Value
Exchange	Island
Symbol	UBS
IncludeBidAsk	<input checked="" type="radio"/> True <input type="radio"/> False

Invoke |  Using GET  Using POST  In New Window

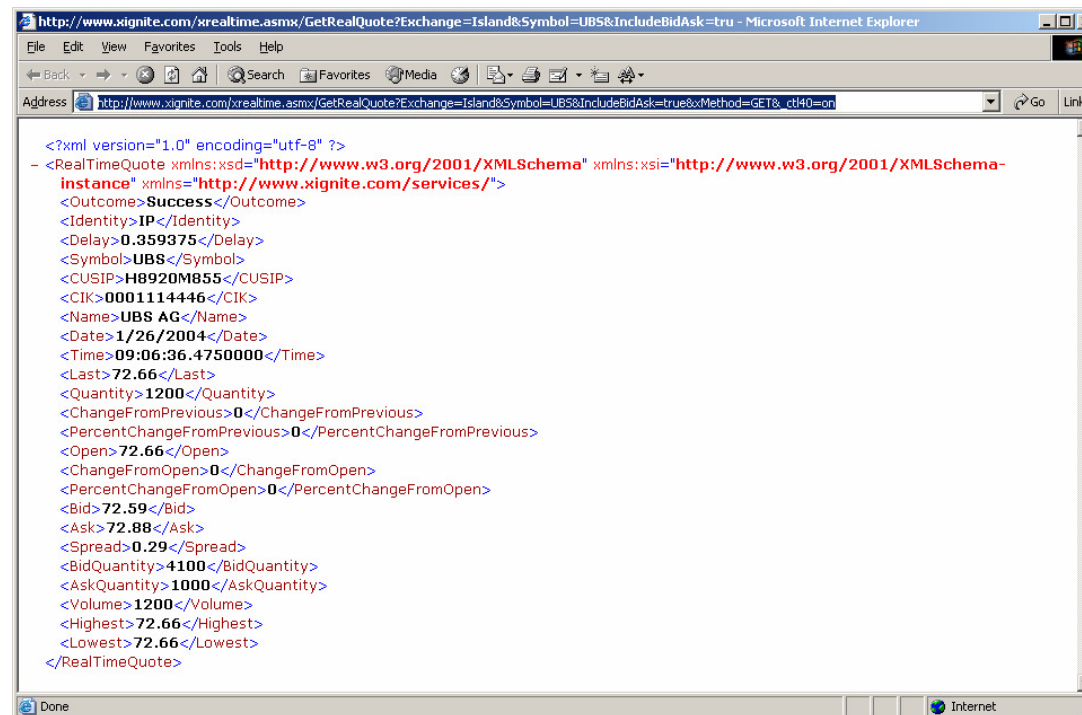
SOAP Request and Response  
The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST http://www.xignite.com/xrealtime.asmx HTTP/1.1
```

# Ein einfaches Beispiel – externe Information



- Phase 3
  - Demo
    - [http://www.xignite.com/xrealtime.asmx/GetRealQuote?Exchange=Island&Symbol=UBS&IncludeBidAsk=true&xMethod=GET&\\_ctl40=on](http://www.xignite.com/xrealtime.asmx/GetRealQuote?Exchange=Island&Symbol=UBS&IncludeBidAsk=true&xMethod=GET&_ctl40=on)



```
<?xml version="1.0" encoding="utf-8" ?>
- <RealTimeQuote xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://www.xignite.com/services/">
  <Outcome>Success</Outcome>
  <Identity>IP</Identity>
  <Delay>0.359375</Delay>
  <Symbol>UBS</Symbol>
  <CUSIP>H8920M855</CUSIP>
  <CIK>0001114446</CIK>
  <Name>UBS AG</Name>
  <Date>1/26/2004</Date>
  <Time>09:06:36.4750000</Time>
  <Last>72.66</Last>
  <Quantity>1200</Quantity>
  <ChangeFromPrevious>0</ChangeFromPrevious>
  <PercentChangeFromPrevious>0</PercentChangeFromPrevious>
  <Open>72.66</Open>
  <ChangeFromOpen>0</ChangeFromOpen>
  <PercentChangeFromOpen>0</PercentChangeFromOpen>
  <Bid>72.59</Bid>
  <Ask>72.88</Ask>
  <Spread>0.29</Spread>
  <BidQuantity>4100</BidQuantity>
  <AskQuantity>1000</AskQuantity>
  <Volume>1200</Volume>
  <Highest>72.66</Highest>
  <Lowest>72.66</Lowest>
</RealTimeQuote>
```

# Erste Hinweise

- Web Services
  - Technische Seite
    - ändern die Art und Weise, wie Verteilte Systeme geplant und realisiert werden.
    - Stellen einen Abstraktionslayer dar, analog zu CORBA, .NET Server, Messaging, ...
    - Können unabhängig vom Betriebssystem, der Hardware Plattform oder der Programmiersprache eingesetzt werden.
    - Sind dem Web angepasst (HTTP Protokoll)
    - Verwenden XML als Basistechnologie

# Erste Hinweise

- Web Services
  - Impact aus Business Sicht
    - Entwicklungskosten können tendenziell gesenkt werden.
    - Integration unterschiedlicher Technologien (MS dotNet, Java, ...) wird möglich (keine Glaubenskämpfe)
    - Getätigte Investitionen in Hardware / Software können genutzt werden.
    - Existierendes Know How kann eingesetzt werden, ohne teure Schulung / Umschulung





# Bausteine der Web Services

- XML – extensible markup language
  - Darauf bauen Web Services auf
  
- WSDL – Web Services Description Language
  - Abstrahieren Web Services
  - Beschreiben die Service Schnittstellen Web-seitig
  
- SOAP – Simple Object Access Protocol
  - Stellen Kommunikationsmechanismen zur Verfügung, mit deren Hilfe Web Services miteinander kommunizieren können
  
- UDDI – Universal Description, Discovery and Integration
  - Registry und Repository Service zum Speichern und bestimmen von Web Service Interfaces
  
- ebXML - electronic business XML
  - Architektur und Spezifikationen zur Automatisierung von Business-Prozessen zwischen Geschäftspartnern



# Key Players der Web Services Szene

- IBM
  - Integrations“figur“
    - Ausgefeilte Tools
    - Viele Technologien
- Microsoft
  - Zu Beginn die treibende Kraft
    - Wichtige Beiträge zur Standardisierung
    - Hervorragende Tools
- Sun Microsystems
  - Aufsteiger
    - Eher schlechte Tools
    - Zum Teil Tendenz zu proprietären Ansätzen
- Weitere Key Players
  - BEA
  - Oracle

# Web Services als Business

- Ansatz
  - Rahmenvertrag mit einem Informations-Provider
  - Belastung
    - Grundgebühr
    - Nutzungsgebühr
  - Modell
    - Telefonie
  - Problem
    - Micro-Payments
      - Kostengünstige Verrechnung kleinster Beträge
      - Wie kann der Einmal-Nutzer eingebunden werden
    - Möglicher Ansatz : Integration von SMS und Mobile
      - Der einmal Benutzer erhält seinen Zugriffscode als SMS auf sein Mobile und sein Mobile wird belastet.



# Einführung in Web Services

- *Web Services verändern alles!*
  - Vom Text und Grafik Web zum Service Web
    - IP ... TCP
    - WWW
    - Web Services
    - Grid
    - Semantic Web
    - X-Internet



# Einführung in Web Services

- *Der aktuelle Web unterstützt Software-orientierte Interaktionen ungenügend!*
  - WWW
    - Ist Repräsentation –orientiert
  - Handelspartner möchten Daten und Informationen so einfach austauschen, wie sie Web Seiten veröffentlichen können.
    - Programm – zu – Programm Kommunikation
    - Integration verteilter Programme und
    - Integration von Daten zur Weiterverarbeitung



# Einführung in Web Services

## Web Service Grundlagen

- *Web Services transformieren XML Dokumente in und aus IT Systemen!*
  - Web Services sind XML Applikationen in Form von Programmen, Objekten oder Datenbanken
    - Programme senden Anfragen an einen Web Service und empfangen deren Antworten in Form von XML-Dokumenten
  - Web Service Standards
    - definieren das *Format* der Messages,
    - spezifizieren das *Interface*, an das die Message gesandt wird
    - beschreiben Konventionen für das *Mapping* des Message Inhalts auf Programme
    - definieren Mechanismen zum Publizieren und finden der Web Service Interfaces



# Einführung in Web Services

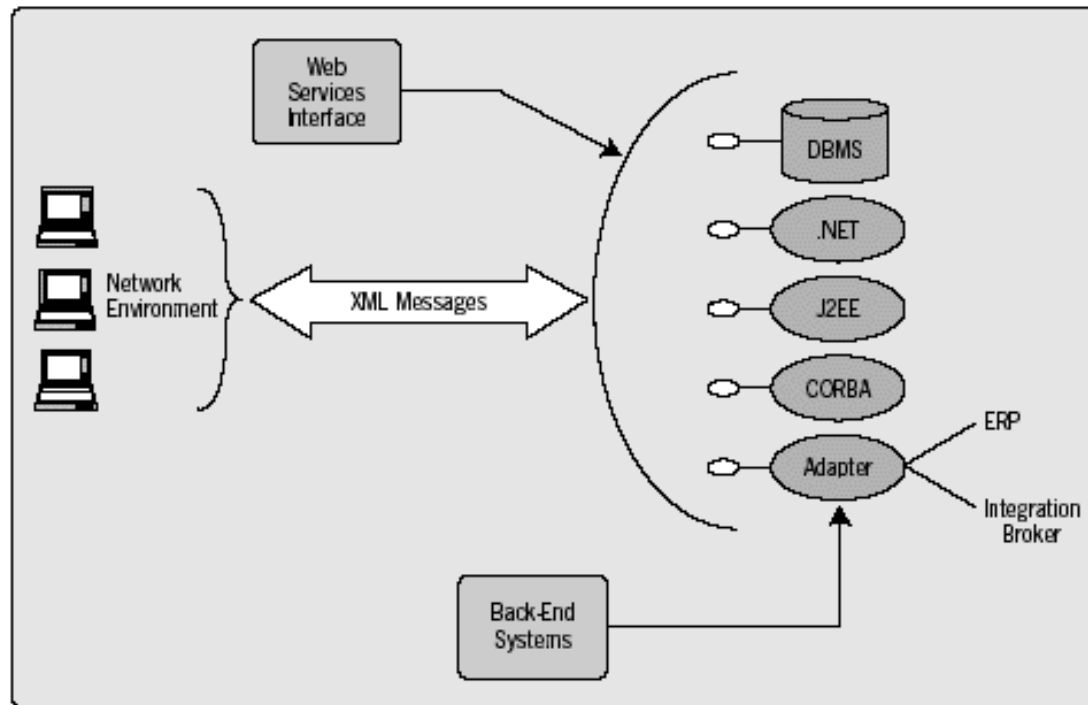
## Web Service Grundlagen

- *Web Services können in den unterschiedlichsten Anwendungen eingesetzt werden!*
  - Web Services
    - können lokal, auf dem Client, eingesetzt werden, um auf Internet Applikationen zuzugreifen.
    - Können in der Enterprise Application Integration (EAI) als Kit eingesetzt werden (Verbindung unterschiedlicher Technologien und Applikationen)

# Einführung in Web Services

## Web Service Grundlagen

- *Web Services können in den unterschiedlichsten Anwendungen eingesetzt werden!*
  - Web Services als *Wrapper* für Backend Systeme





# Einführung in Web Services

## XML

- *XML ist besser geeignet, Daten zu versenden*
  - *Die Daten lassen sich leicht strukturieren*
  - *Der Overhead ist in der Regel vertretbar!*
    - *Im Vergleich zu Edifact ist er äusserst gering*
    - *Im Vergleich zu einem reinen Text-Dokument ist XML viel leichter verarbeitbar*
  - **<SOAP-ENV:Body>**
    - <s:PO...>**
      - <Bestellung ...>**
        - <BestellPos>Verpackung</BestellPos>**
        - <BestellPos>Kabel 2m CH</BestellPos>**
        - <...>**
      - </Bestellung>**
      - </s:PO>**
    - </SOAP-ENV:Body>**

# Einführung in Web Services

## Ein Beispiel

- *Web Services verwenden XML Dokumente*
  - XML kennt zusammengesetzte Datentypen
    - `<Adresse>`
      - `<Name>...</Name>`
      - `<Vorname>...</Vorname>`
      - `<Strasse>...</Strasse>`
      - `<PLTZ>...</PLTZ>`
    - `</Adresse>`
  - XML Daten lassen sich einfach in andere Programme einlesen (weil sie sauber strukturiert sind)
  - XML Dokumente sind wohlgeformt und können validiert werden (sind die Daten korrekt?)



# Einführung in Web Services

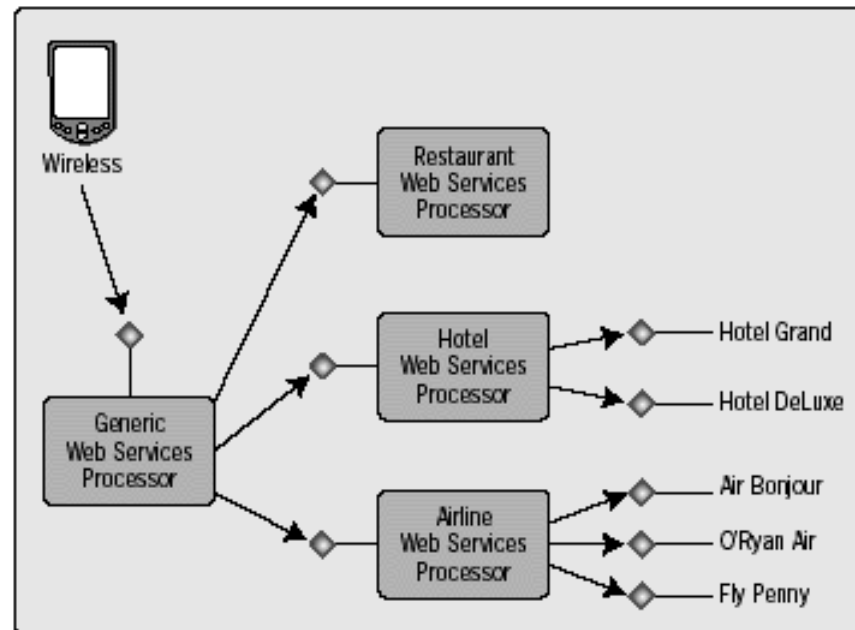
## Die nächste Web Generation

- *Die nächste Web Generation wird Software-orientierte Interaktion gestatten*
  - Software-orientierte Interaktion gestattet
    - Das Suchen und Kaufen von Ware und Diensten (und die Preisoptimierung : wer ist am günstigsten?)
    - Koordination unterschiedlicher Abläufe (Reservation, Buchung einer Veranstaltung und Bahn)
    - Die Vereinfachung von Geschäftsabläufen (Procurement : Einkauf, Verkauf, Nachlieferung, ...)

# Einführung in Web Services

## Die nächste Web Generation

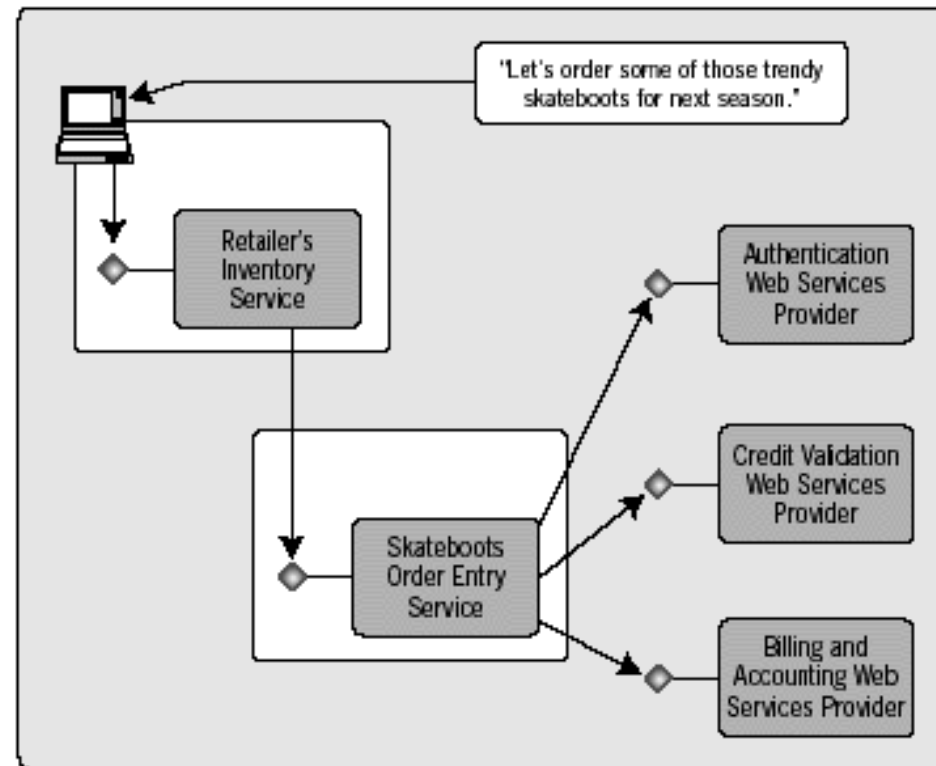
- *Web Services ermöglichen neue Interaktionen / Geschäfts-Typen*
  - Reservation des Restaurants  
(aus Tokyo für ein Mövenpick Bern)



# Einführung in Web Services

## Die nächste Web Generation

- *Web Services entdecken sich gegenseitig und interagieren miteinander – in Zukunft*
  - Kauf eines Artikels, der nachbestellt werden muss





# Einführung in Web Services

## Die nächste Web Generation

- *Web Services erlauben die Optimierung von Geschäftsprozessen*
  - Die Zusammenarbeit von Unternehmen wird vereinfacht  
(EAI: Enterprise Application Integration, sowohl Intraprise als auch Interprise Integration)
  - (Web) Services lassen sich zu Workflows kombinieren
    - Gemeinsam lösen mehrere Workflows ein oder mehrere Business Problem(e).

# Einführung in Web Services

## Interaktion mit Web Services

- *Web Services unterstützen unterschiedliche Messaging Paradigmen*
  - Web Services können von sich aus aktiv werden (wie ein Börsenticker)
  - Web Services können selektiv abonniert werden
    - Nur News zum Thema Sport und speziell Snowboarding
  - Web Services können gezielt zwischen zwei Partnern aufgebaut werden
    - Bestellungen übermitteln
    - Lagerinformationen abfragen (die Produktion schaut in die Vertriebslager)



# Einführung in Web Services

## Interaktion mit Web Services

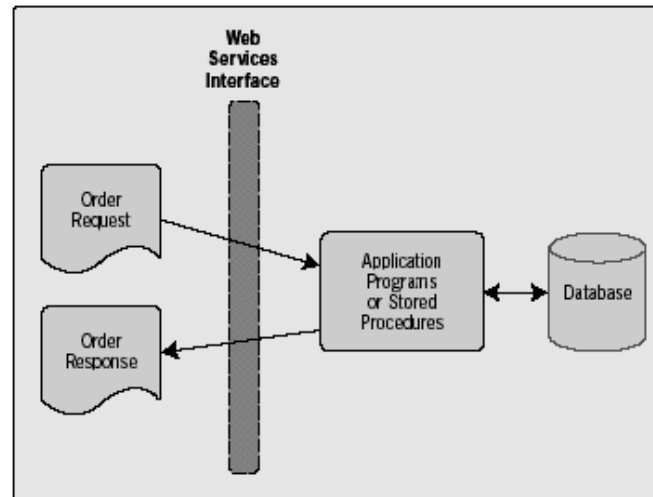
- *Web Services überbrücken die Unterschiede zwischen Dokument-orientierter Interaktion und Abfragen*
  - Web Services unterstützen beide Paradigmen
    - Abfrage einer Informationsquelle mit fixen Abfrageparametern
      - UBS Börsendaten
    - Übermitteln eines XML Dokuments an einen Business-Partner
      - Bestellformular
      - Lieferschein
      - Lagerlisten



# Einführung in Web Services

## RPC-orientierte Interaktion mit Web Services

- *Abfragen-orientierte Interaktionen sind sinnvoll, falls das Datenvolumen gering ist*
  - Schema des Web Services
    - Methode + Eingabe- oder Ausgabe-Parameter
      - `http://finance.yahoo.com/d/quotes.csv?f=s1d1t1c1ohgv&e=.csv&s=ubs`
      - Liefert  
"UBS",73.13,"2/6/2004","2:21pm",+1.48,72.70,73.39,72.59,372400



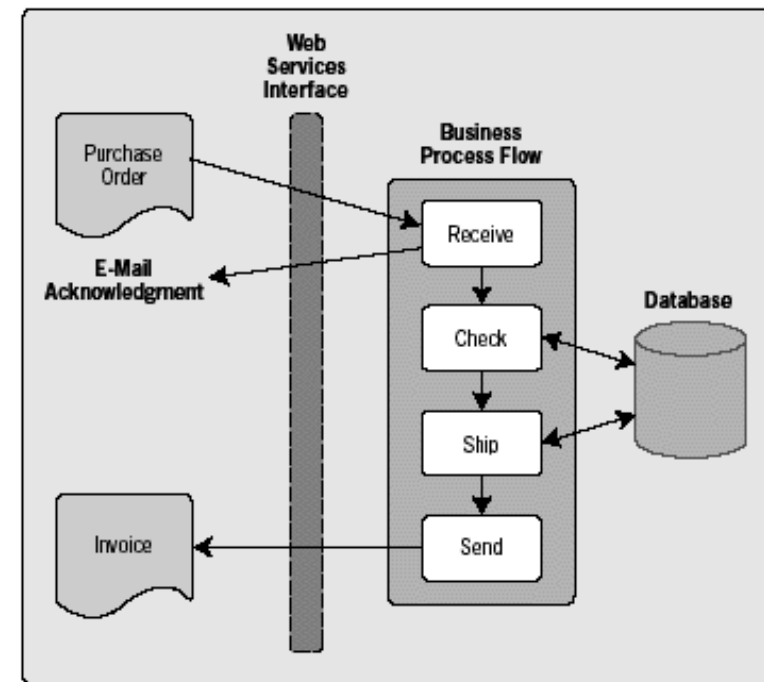
## Einführung in Web Services

### Dokument-orientierte Interaktion mit Web Services

- *Dokument-orientierte Interaktionen sind sinnvoll, falls das Datenvolumen hoch ist*
  - Dokument-basierter Web Service
    - XML Dokument (Einkaufsformular)

Geschäftspartner einigen sich auf das Austauschformat (Daten, Aufbau)

- receive
- check
- ship
- send





# Einführung in Web Services

## Die Technologie der Web Services

- *Standards definieren, wie Web Services beschrieben und entdeckt werden und wie Services miteinander kommunizieren.*
  - Parameter der Kommunikation könnten sein
    - Request / Reply oder komplexere Partnerbeziehung
    - Quality of Service
    - Sicherheit
    - Message oder Transaktions-orientierte Kommunikation

# Einführung in Web Services

## Die Technologie der Web Services

- *Web Services benutzen unterschiedliche XML-basierte Technologien.*
  - XML (Extensible Markup Language)
    - Web Services verwenden verschiedene Standards von W3C
  - WSDL (Web Services Description Language)
    - XML-basierte Technologie zur Definition der Web Service Schnittstellen, Datentypen, Interaktionsmuster, Protokolle
  - SOAP (Simple Object Access Protocol)
    - Definition eines Umschlages und einer Serialisierung von XML-Dokumenten sowie Konventionen für die RPC-basierte Kommunikation
  - UDDI (Universal Description, Discovery and Integration)
    - Mechanismus für Speicherung und Kategorisierung von Business Informationen und Verweisen auf dazu gehörige Web Service Interfaces

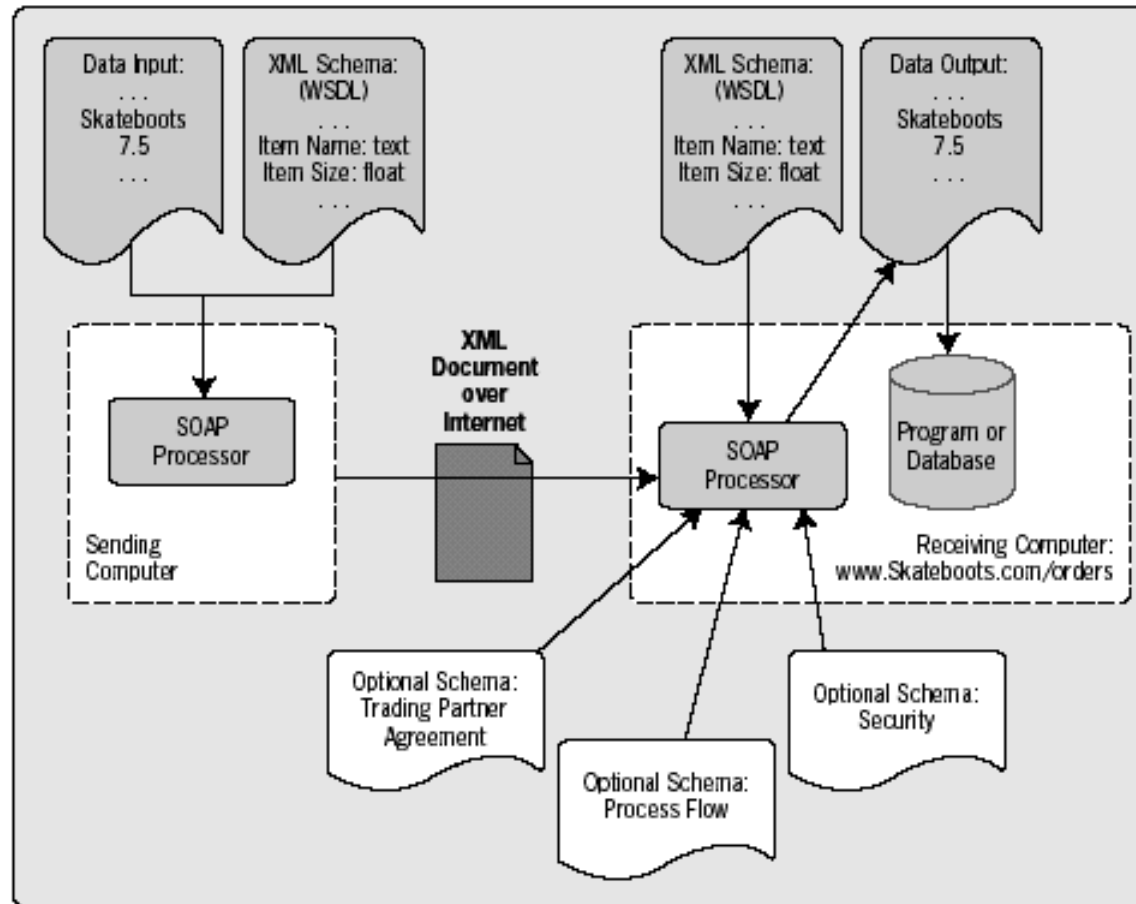


INTERNET  
EXPO

## Einführung in Web Services

### Die Technologie der Web Services - Einsatzbeispiel

- *Web Services Standards werden in der Regel gebündelt eingesetzt*





# Einführung in Web Services

## Die Technologie der Web

- *Web Services Description Files werden typischerweise als URL's angegeben*
  - Web Services sind eine spezielle Art von Web Ressourcen (URL : Uniform Resource Locator)
  - Web Services werden im Web mit den selben Techniken einsetzbar wie Web Seiten (downloaden und einsetzen), allerdings auf XML Basis.
  - „XML basiert“ impliziert, dass die Informationen validiert werden können.



## Einführung in Web Services

### Die Technologie der Web Services - Einsatzbeispiel

- *Web Services benutzen XML Schema für die Validation der XML Messages*
  - Nach dem Empfang eines XML Dokuments muss der Service das XML Dokument zuerst parsen und prüfen
    - Quality of Service
    - Sicherheit
    - Initialisieren und Ausführen von Business Prozessen
  - Web Services werden mittels WSDL Dateien identifiziert
    - Diese können vom sendenden Service herunter geladen und für die Generierung der Message eingesetzt werden.



## Einführung in Web Services

# Die Technologie der Web Services – XML als Basis

- *XML wird für verschiedene Aufgaben eingesetzt*
  - XML definiert das Message Format.
  - XML hat Auswirkungen auf die Art und Weise, wie Services und deren Wechselwirkung definiert wird.
  - XML ist die Basis der Web Service Beschreibung (WSDL – Web Service description Language)
  - XML ist die Basis von SOAP (Simple Object Access Protocol)





# Einführung in Web Services

## Die Technologie der Web Services – XML als Basis

- *XML gestattet es uns mehrere Elemente zu kombinieren*

```
- <?xml version="1.0"?>
<INVENTORY>
  <BOOK>
    <TITLE>The Adventures of Huckleberry Finn</TITLE>
    <AUTHOR>Mark Twain</AUTHOR>
    <BINDING>Taschenbuch</BINDING>
    <PAGES>336</PAGES>
    <PRICE>EUR 12,75</PRICE>
  </BOOK>
  <BOOK>
    <TITLE>The Legend of Sleepy Hollow</TITLE>
    <AUTHOR>Washington Irving</AUTHOR>
    <BINDING>Taschenbuch</BINDING>
    <PAGES>98</PAGES>
    <PRICE>EUR 4,95</PRICE>
  </BOOK>
</INVENTORY>
```



# Einführung in Web Services

## Die Technologie der Web Services – XML als Basis

- *XML Schema schränkt die Flexibilität ein*
  - Die hohe Flexibilität von XML kann zu Problemen führen.
  - Um die XML Dokumente gegenseitig interpretieren zu können, werden Regeln aufgestellt (XML Schema).
    - Die Definition der Daten und Elemente
    - Allfällige Beziehungsrestriktionen (Datenmodell)
    - Einschränkung der erlaubten Datentypen (Zeichen, Zahlen, komplexere selbst definierte Strukturen, wie Adressen, Formulare ...)



# Einführung in Web Services

## Die Technologie der Web Services – WSDL

- *WSDL ist das XML Format mit dem beschrieben wird, aus was die Web Services bestehen*
  - WSDL's sind XML Schema Formate, welche ein erweiterbares Framework (für W.S.) beschreiben.
    - WSDL wurde von MS und IBM entworfen und an W3C übergeben.
    - WSDL ist der Kern des Web Service Frameworks, im Einzelnen beschreibt man mit WSDL:
      - Darstellung unterschiedlicher Datentypen
      - Operationen auf Messages
      - Mapping von Messages auf den Netzwerk-Transport



# Einführung in Web Services

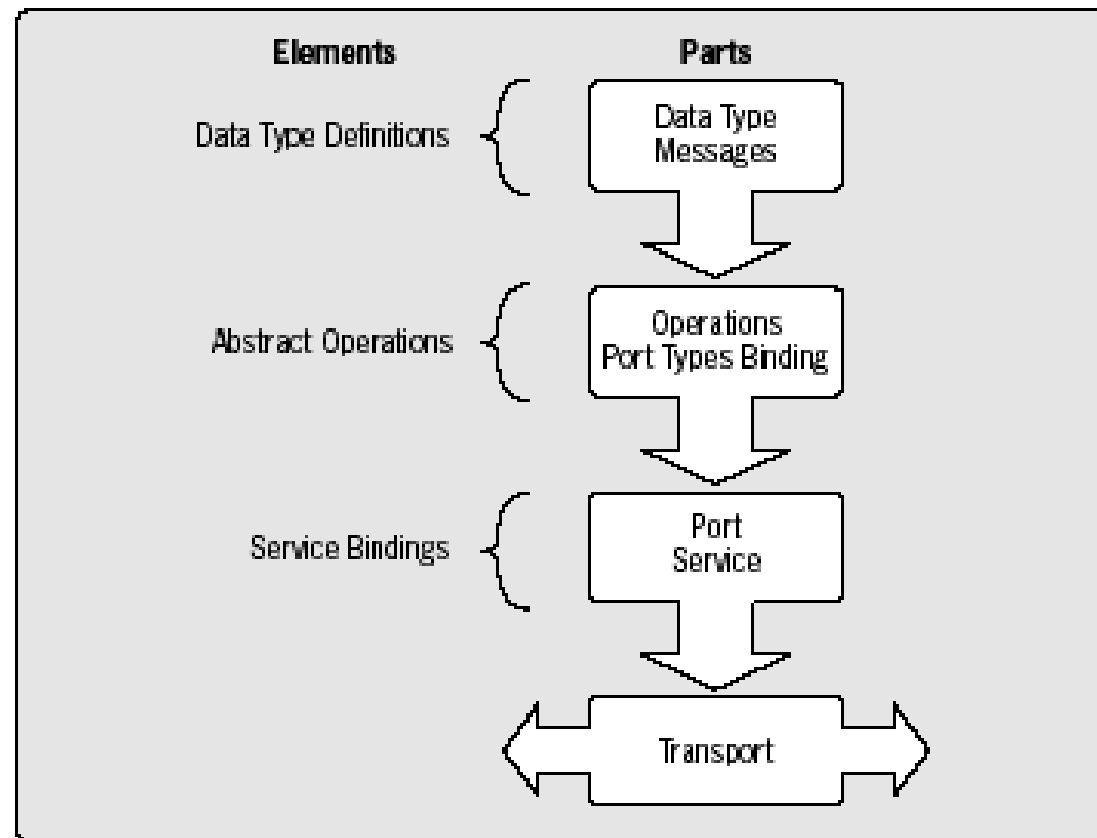
## Die Technologie der Web Services – WSDL

- *WSDL besitzt drei Hauptelemente / Abstraktionsebenen*
  1. Datentyp-Definition
    - Bestimmt die Struktur und den Inhalt der Message
  2. Abstrakte Operationen
    - Bestimmt die Operationen, welche auf dem Message Inhalt ausgeführt werden dürfen.
  3. Service Bindung
    - Bestimmt den Netzwerk-Transport, mit dem die Message übermittelt wird.

# Einführung in Web Services

## Die Technologie der Web Services – WSDL

- *WSDL Elemente können in separaten Dokumenten definiert werden*





# Einführung in Web Services

## Die Technologie der Web Services – WSDL

- *WSDL Interfaces sind technische Spezifikationen der Web Services*
  - Datentyp-Definitionen in WSDL sind ähnlich wie in anderen Middleware Technologien.
- *Web Service Datentypen basieren auf XML Elementen, können aber auch erweitert werden.*
  - Eigene Datentypen
- *Abstrakte Messages und Operationen werden auf spezifische Transportmechanismen abgebildet.*
  - In der Regel ist dies SOAP (XML Protokoll).
  - Auf HTTP Basis also über das Web

# Einführung in Web Services

## Die Technologie der Web Services – WSDL

- *Namespaces garantieren die Eindeutigkeit der Namen der WSDL Elemente.*
  - Falls Teile des WSDL separat entwickelt werden, ist die Eindeutigkeit nicht immer trivial.
  - Namespaces werden als Präfix der Elementenamen verwendet.
  - Falls unterschiedliche Geschäftspartner zusammen arbeiten wollen, müssen sie garantieren, dass nicht die selben Namen für unterschiedliche Konzepte / Inhalte verwendet werden
    - Adresse
      - Lieferant, Kunde : Business-Partner
      - Physikalische Adresse eines Informatik Dienstes
      - Uneinheitlicher Aufbau der Adresse der Business-Partner



# Einführung in Web Services

## Die Technologie der Web Services – SOAP

- *SOAP stellt den Kommunikationsmechanismus für die Verbindung zum Web Service zur Verfügung.*
  - SOAP definiert das Messaging Framework für den Austausch von XML Daten.
  - Das Framework ist simpel, einfach zu implementieren und neutral (Betriebssystem, Programmiersprache, Plattform).
  - SOAP bietet eine minimale Basis, auf der komplexere Mechanismen aufgebaut werden können.





# Einführung in Web Services

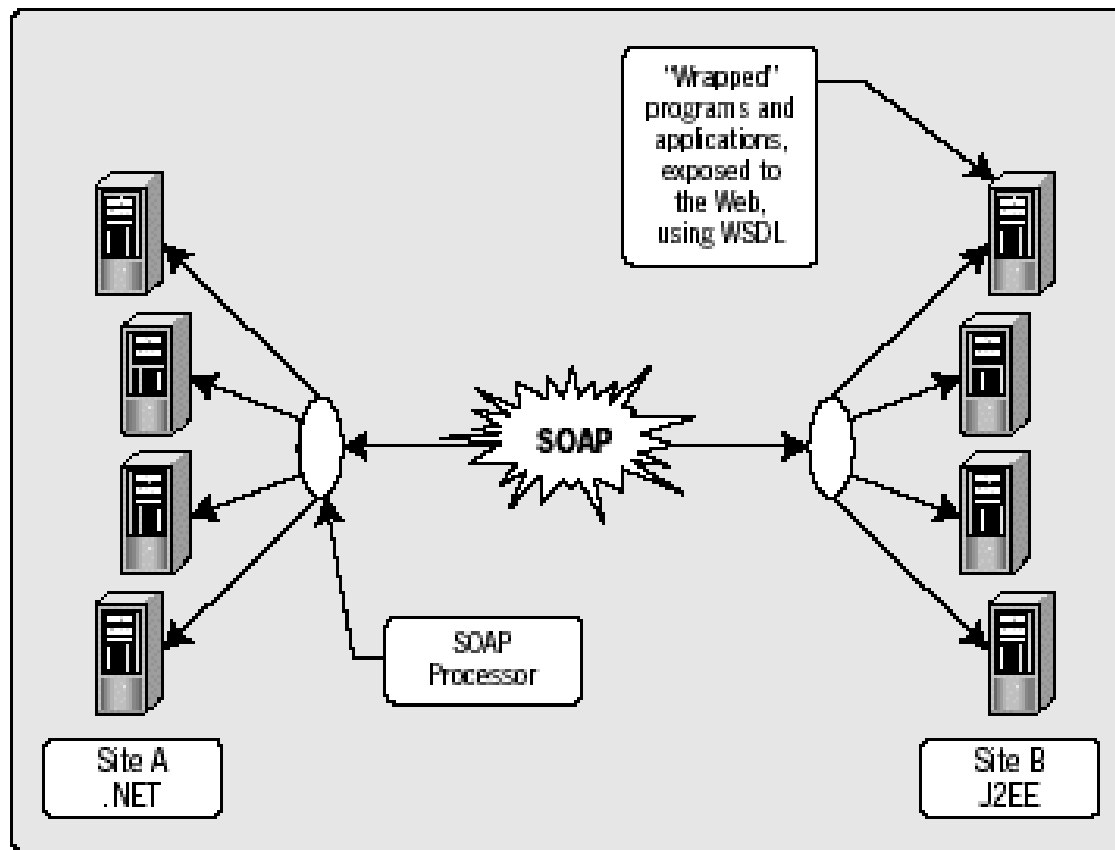
## Die Technologie der Web Services – SOAP

- *SOAP ist der Weg, wie man in XML Daten versendet, empfängt und beschreibt das Wie.*
  - SOAP umfasst im Wesentlichen eine Einweg-Kommunikation
    - Über PUT / (Send/Reply) Mechanismen kann ein RPC ähnliches System realisiert werden.
      - Sende eine Nachricht, ein Dokument an einen Empfänger
    - SOAP definiert auch mögliche Verschlüsselungskonzepte

# Einführung in Web Services

## Die Technologie der Web Services – SOAP

- *SOAP ist der Weg, wie man in XML Daten versendet, empfängt und beschreibt das Wie.*



# Einführung in Web Services

## Die Technologie der Web Services – SOAP

- *SOAP Messages enthalten (mindestens) einen Umschlag, einen Kopf und einen Rumpf.*
  - **Umschlag (Envelope) [Muss]**
    - Definiert Start und Ende der Message
  - **Kopf (Header)**
    - Enthält optionale Attribute, welche für die Verarbeitung der Message wichtig sind.
  - **Rumpf (Body) [Muss]**
    - Enthält die in XML beschriebene Message
  - **Anhänge (Attachment)**
    - Optionale Dokumente
  - **RPC Interaktion**
    - Definiert, wie RPC-ähnlich mit SOAP kommuniziert werden kann.
  - **Verschlüsselung (Encoding)**
    - Definiert, wie einfache und komplexe Daten in der Message übermittelt werden können.



# Einführung in Web Services

## Die Technologie der Web Services – UDDI

- *UDDI registriert und publiziert Web Service Definitionen.*
  - Das UDDI Framework definiert ein Datenmodell in XML und SOAP API's für die Registrierung und die
  - Suche (*Discovery*) der Web Services.
    - UDDI wurde von MS, IBM, Ariba, ... definiert und ist nicht in W3C.
    - UDDI ist ähnlich wie DNS in TCP/IP (replizierte Datenbanken).
    - Öffentliche UDDI Datenbanken
      - IBM, Microsoft, SAP

# Einführung in Web Services

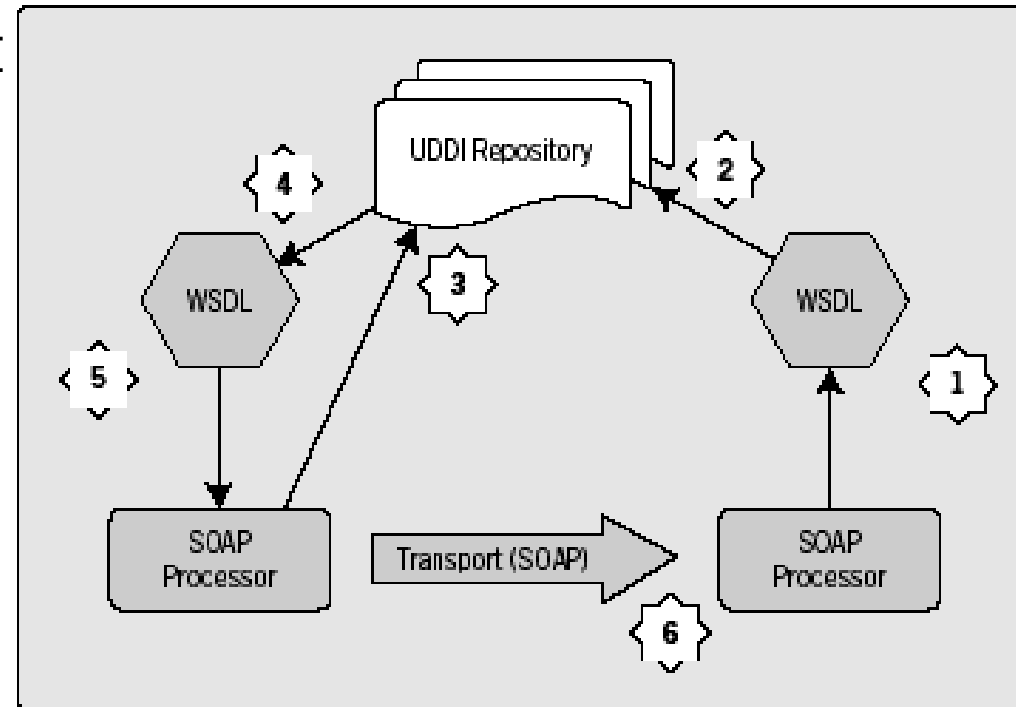
## Die Technologie der Web Services – UDDI

- *UDDI ist ein Verzeichnis von Web Services.*

1. WSDL Beschreibung erstellen
2. Registrierung mittels UDDI API

3. **Client:**  
Suchen eines Dienstes mittels UDDI API

4. Bestimmen der WSDL des Dienstes
5. Generieren der Client Message
6. Versenden der Message zum Dienst





# Einführung in Web Services

## Die Technologie der Web Services – ebXML

- *Die Spezifikation von ebXML geht über die Definition von Web Services hinaus.*
  - Das ebXML Konsortium definierte typische Einsatz-Muster für den XML Dokumente-Austausch.
  - Die Spezifikation basiert auf SOAP.
  - WSDL wird *nicht* verwendet.
  - Zusätzlich werden
    - Quality of Service
    - Sicherheit
    - Garantiertes Messaging
    - ...  
spezifiziert.



## Einführung in Web Services

### Die Technologie der Web Services – ebXML

- *Die Spezifikation von ebXML definiert den Einsatz von XML für kooperierende Business-Prozesse.*
  - ebXML wurde durch UN/CEFACT unterstützt.
    - Die ebXML Architektur beginnt mit der Definition der Business-Prozesse und Informations-Modelle.
    - Bildet diese auf XML Schema ab
    - Und definiert Anforderungen, die erfüllt sein müssen, damit Business-Partner XML Dokumente austauschen können.



# Einführung in Web Services

## Die Technologie der Web Services – Vergleich mit anderen Technologien

- *Web Services unterscheiden sich von traditionellen Technologien.*
  - DCOM, CORBA, EJB, ... sind eigenständig
  - Web Services lehnen sich an HTML und HTTP an.
    - Einweg-Kommunikation (GET)
    - Asynchrone Kommunikation
    - Programmiersprachen-unabhängige Datenformate.
  - Web Services sind ähnlich wie Adapter.
  - Web Service Aufrufe sind analog zu traditionellen Methoden/Prozedur-Aufrufen.





# Einführung in Web Services

## Die Technologie der Web Services – Zusätzliche Technologien

- *Je nach Anforderungen werden neue Technologien noch hinzukommen..*
  - Typische Kandidaten
    - Sicherheit
      - SAML (Security Authorization Markup Language)
      - XKMS / XML Key Management
      - Digitale Signaturen
      - MS: WS-Security
    - Prozesse / Work Flow
      - Automatisieren der Geschäftsabläufe
    - Transaktionen
      - BTP : Business Transaction Protocol (OASIS)
    - Messaging
      - IBM, MS, ...: verbessertes Messaging plus Zusatzfunktionalitäten
        - » WS-Inspection für die Discovery
        - » WS-Referral, WS-Routing (bessere Lokalisierung der Services)



# Einführung in Web Services

## Die Technologie der Web Services – Zusätzliche Technologien

- *Je nach Anforderungen werden neue Technologien noch hinzukommen..*
  - Typische Kandidaten
    - BEEP (Block Extensible Exchange Protocol) [IETF]
      - Definiert ein verbindungsorientiertes Internetprotokoll
      - Ein SOAP Mapping wurde definiert.
      - Mehrere Messages können als zusammengehörig gekennzeichnet werden.
  - Akteure
    - OASIS
      - » Kümmt sich um ebXML, BTP und SAML
    - RosettaNet
    - Userland
      - » XML-RPC
    - OAGI (Open Applications Group Inc.)
      - » Definiert Business Formate in XML



# Einführung in Web Services

## Die Technologie der Web Services – Anbieterspezifische Ansätze

- *Ansätze.*
  - Schnittstellen zu DB Systemen.
  - Schnittstellen zu Applikations-Servern.
  - Schnittstellen zu Integrations-Brokern.
  - Integration unterschiedlicher Technologien.
  - Als Architektur für Software-Komponenten.



# Web Services als Baustein in einer IT Architektur

- Vom Monolith zu einer Service orientierten IT Architektur
  - Monolithische Systeme
    - SAP R2 (Mainframe)
      - Lösen Business Probleme in der Regel auf eine und nur eine Art und Weise
      - Sind schwer wartbar (Release-Wechsel, Wartungskosten)
      - Sind kaum ersetzbar (sehr customized)
      - Sind kaum an neue Geschäftsabläufe anpassbar
    - Funktionieren
    - Leisten einen entscheidenden Beitrag zum Geschäftserfolg



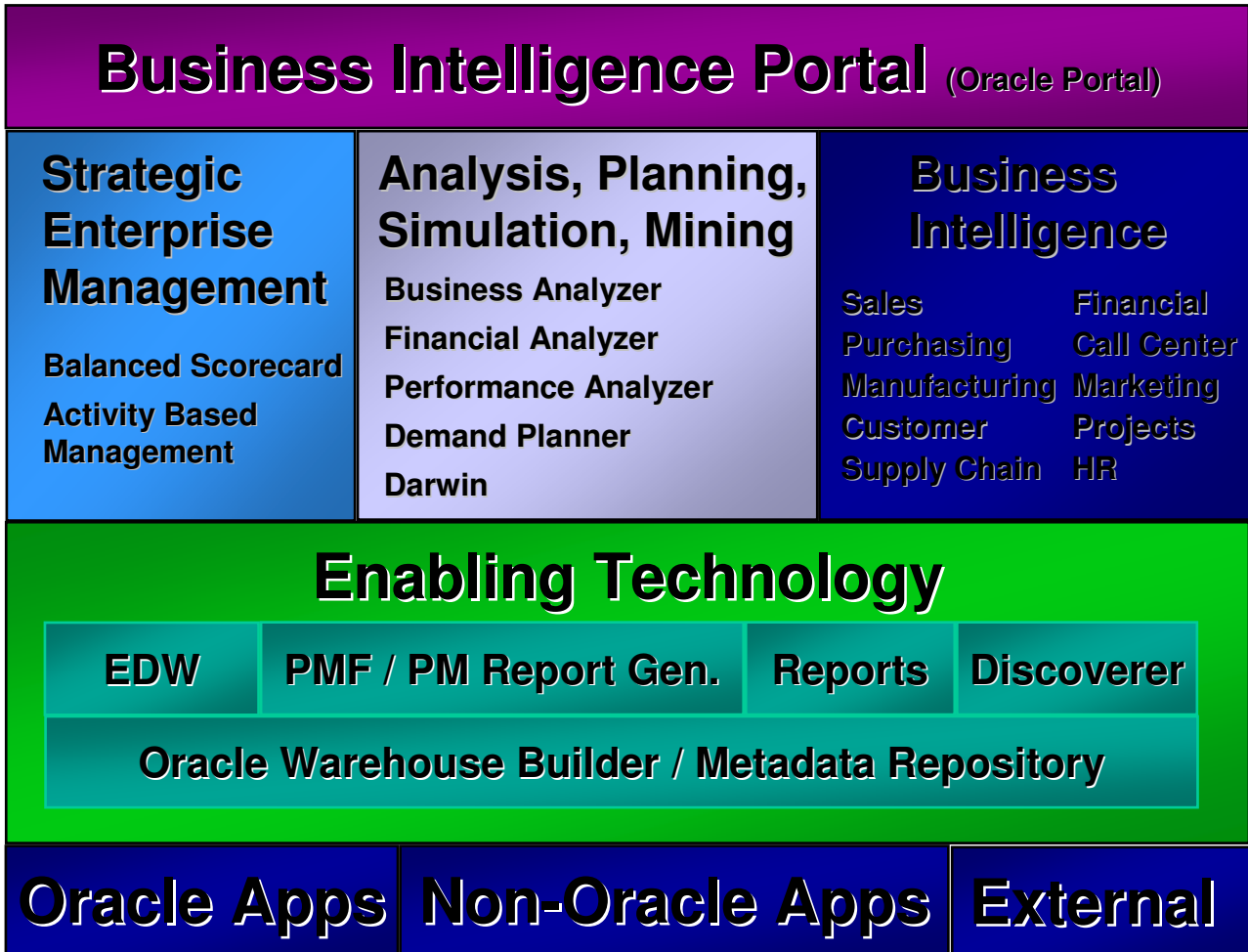
# Monolith Oracle E-business Intelligence Suite 11i

**Personalized View**

**Analysis  
Align Strategy...**

**Accessibility  
Availability  
Accuracy**

**Multiple Sources**

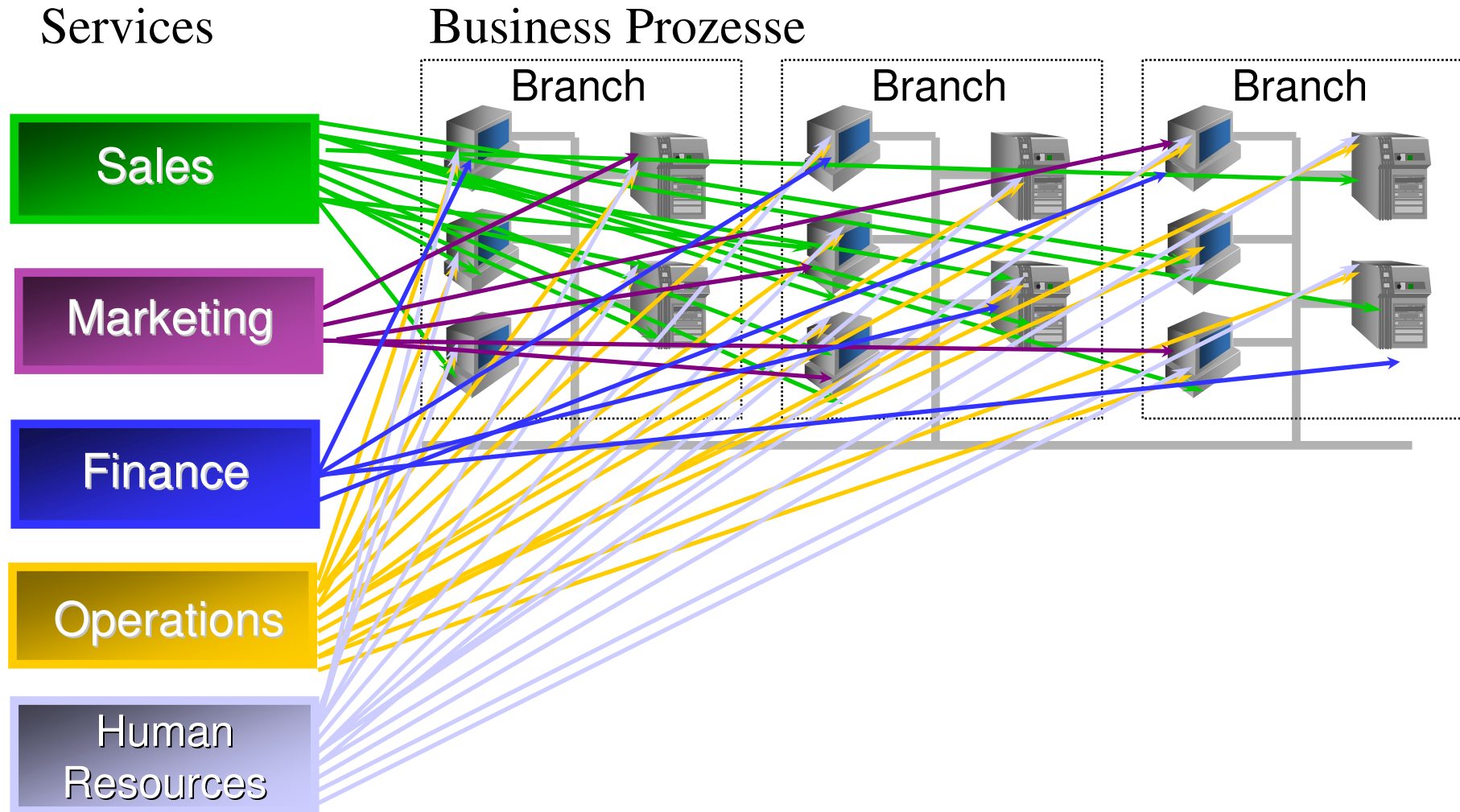




# Web Services als Baustein in einer IT Architektur

- Vom Monolith zu einer Service orientierten IT Architektur
  - Service orientierte Systeme
    - Lösen Business Teil-Probleme in der Regel auf eine und nur eine Art und Weise
    - Lassen sich zu komplexeren Business Prozessen kombinieren (Workflows, komplexere Use Cases)
    - Sind leicht anpassbar
    - Lassen sich leicht ersetzen
    - Lassen sich durch neue Dienste ergänzen
  - Funktionieren heute in einem leichtgewichtigen Umfeld
  - Leisten einen entscheidenden Beitrag zum Geschäftserfolg

# Services und deren Nutzung

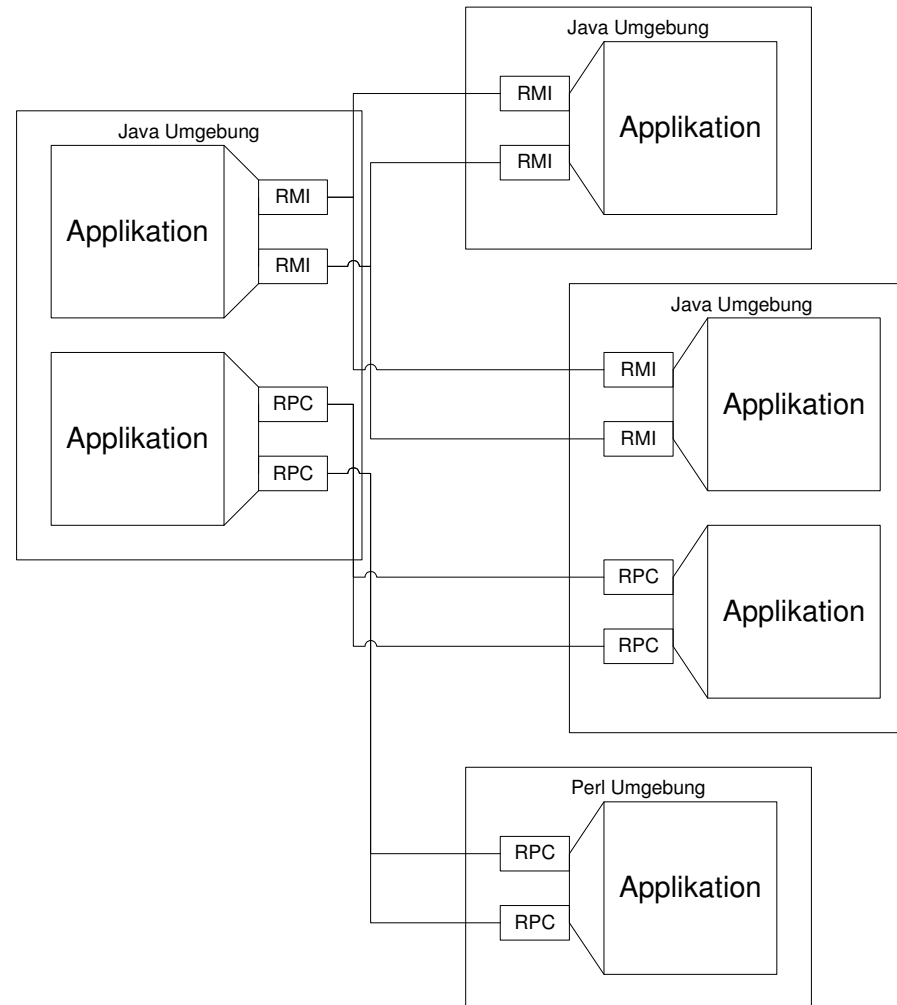


# Web Services als Integrationskonzept

- Die erfreuliche Nachricht
  - Sie brauchen sich nicht an den Positionskämpfen zu beteiligen (Microsoft, IBM, Sun, ... Arbeiten zusammen)
  - Microsoft
    - Hat fundamentale Beiträge geleistet (und leistet weiter wesentliche Beiträge)
    - Die Tools gehören zu den Besten
  - Sun
    - War eher ein Nachzügler
    - Die Software und die Tools sind nicht gerade benutzerfreundlich
  - IBM
    - Besitzt integrierende Wirkung auf Microsoft und Sun



# Web Services als Integrationskonzept



# Web Services als Integrationskonzept

- Heterogenität
  - Web Services können als Kit / Leim für unterschiedliche Plattformen oder Technologien eingesetzt werden
  - Web Services stehen (heute) nicht in Konkurrenz zu klassischen Monolithen
  - Web Services können mehrere Monolithen oder Monolithen mit ergänzenden Speziallösungen verknüpfen und dem Client / Benutzer einheitliche Dienste zur Verfügung stellen

## Welche Schritte sind sinnvoll?

- *Starten Sie mit einem überschaubaren Pilotprojekt*
  - Auswahlkriterium:
    - Welche Integration schafft einen grossen Business Value?
  - Zeitumfang:
    - Einige (wenige) Monate
  - Budget:
    - Einige Tausend bis einige Zehntausend
  - Technologie:
    - Abhängig vom bestehenden Umfeld (HP, IBM, ...)

## Projekt (-Ideen)

- *Integration Inhouse (EAI)*
  - Integration zweier Applikationen, welche strukturierte Daten strukturiert austauschen können.
  - Der Austausch der Daten gehorcht einem der Webservices Muster:
    - RPC oder Document
  - Phase 1
    - Der Webservice stellt vereinfacht gesagt eine Art Batch Verbindung zwischen den Applikationen her
  - Phase 2
    - Der Webservice wird stärker integriert



## Projekt (-Ideen)

- *Integration Inhouse (EAI)*
  - Praktische Erfahrung
    - Wir haben für unterschiedlichste Anwender EAI Projekte realisiert
      - Produktionsfirmen
      - Vertriebsfirmen
      - Handelsfirmen
    - Integration zwischen Vertrieb und Produktion  
zwischen Zentrale und Aussenstellen

## Projekt (-Ideen)

- *Informatik Migration*
  - Sie wollen Daten aus einer bestehenden in eine neue Applikation übernehmen
  - Webservices bieten sich als „Wegwerflösung“ an
    - Sie können schnell erstellt werden
    - Die Nutzung ist einfach
    - Die Realisierung ist kostengünstig
  - Beispiel aus der Praxis
    - Integration der Produktion (im Ausland) mit dem Headquarter führte bei der Neueinführung eines integrierten Informatiksystems zu personellen Engpässen.
    - Ein Webservice reduzierte das Abtippen der Business-Dokumente auf ein Minimum

## Projekt (-Ideen)

- *Anbindung von Aussenstellen*
  - Sie wollen Daten aus einer bestehenden aber nicht lokalen Anwendung in eine lokale Applikation übernehmen
  - Webservices bieten sich als „EDI des armen Mannes“ an
    - Sie können schnell erstellt werden
    - Die Nutzung ist einfach
    - Die Realisierung ist kostengünstig
  - Beispiel aus der Praxis
    - Anbindung der Filial-Shop-in-Shops an das Logistikzentrum in einem Lebensmittel Produktionsbetrieb.



## Wo stehen wir heute?

- Zur Zeit werden folgende Themen in Webservices noch genauer untersucht
  - Transaktionen
  - Business-Workflows
  - Security
  - Performance





## Ihr nächster Schritt

- ... Hoffentlich ein Pilotprojekt



## Fragen und Anregungen

- Wir sprechen gerne mit Ihnen über Ihre Probleme und helfen Ihnen diese zu lösen
- Wir sind nicht an Projekten interessiert, welche unendlich lange dauern und uns blockieren
- uns interessiert eher: wie geht's weiter?
  - Kontakt:
    - Dr Josef M. Joller  
joller@joller-voss.ch