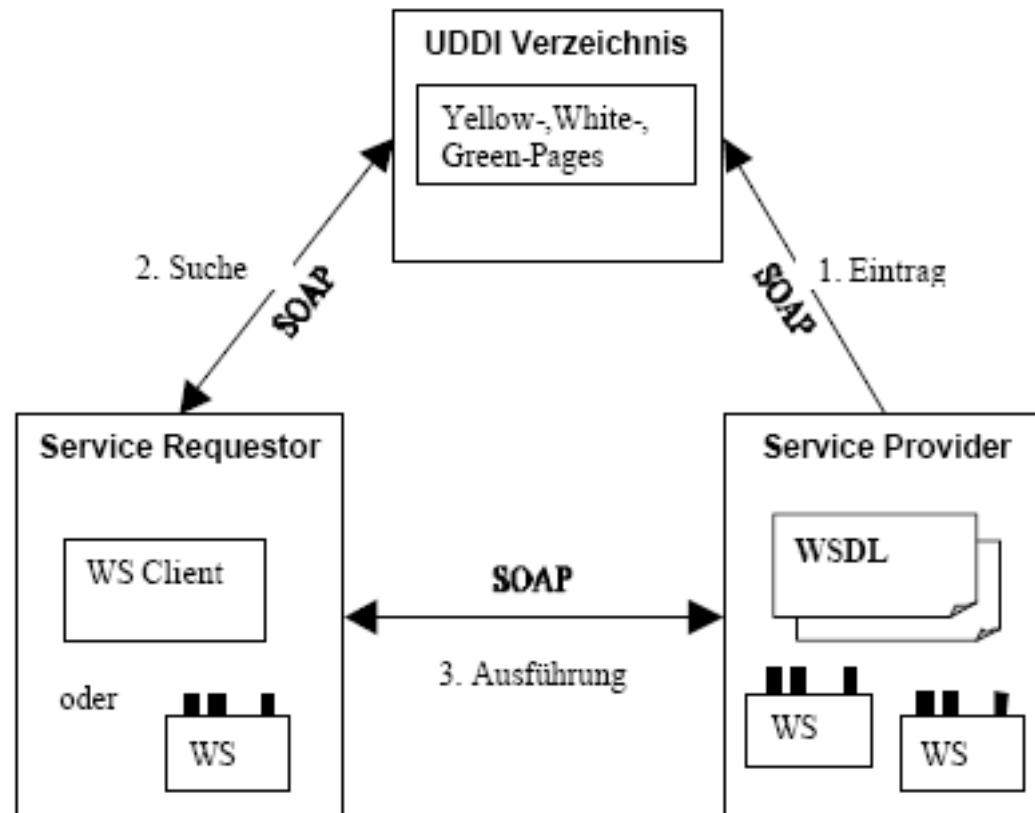
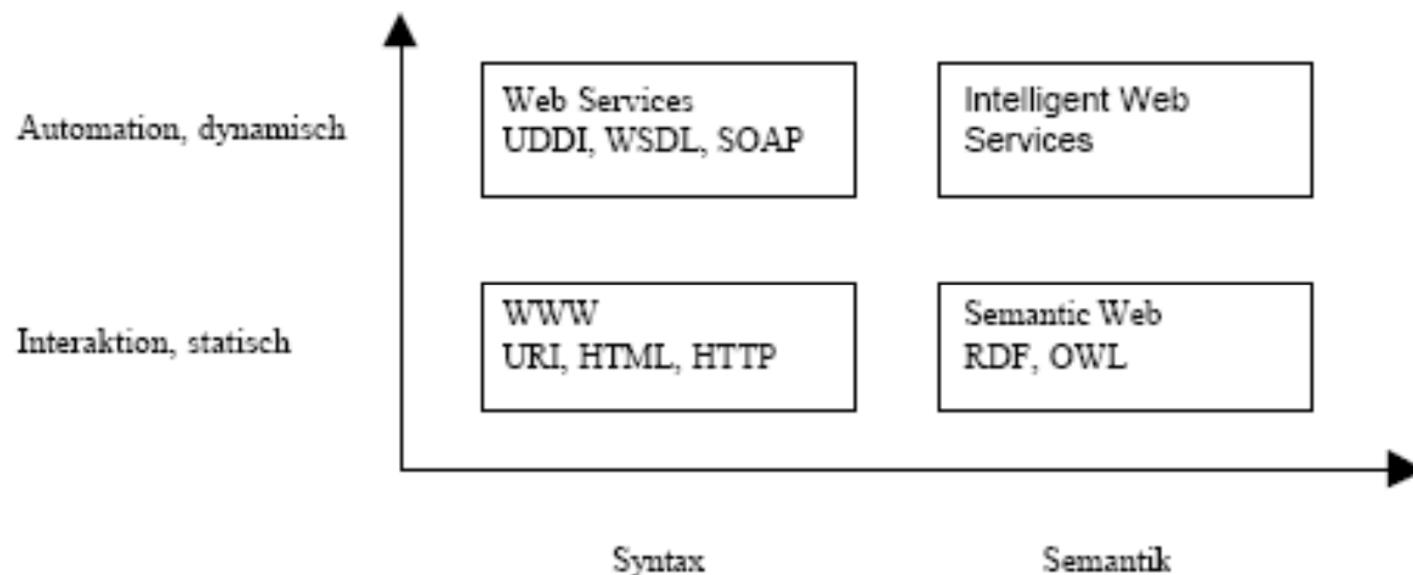


Kapitel 6: Workflows im Internet

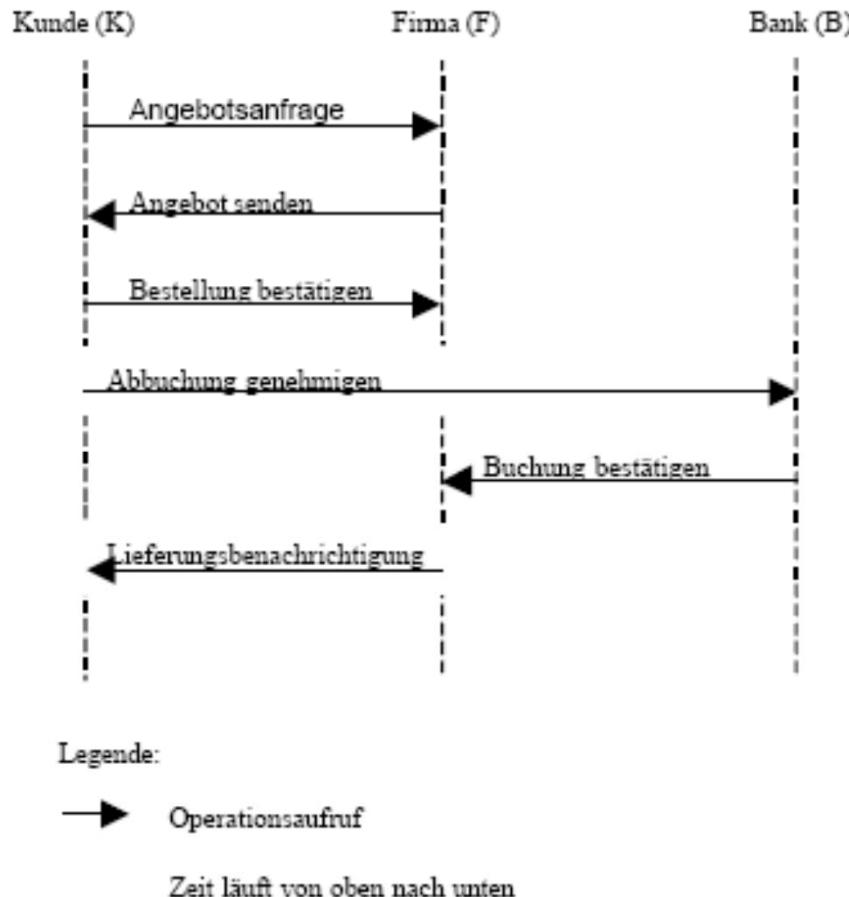
Service-orientierte Architektur (SOA) mit Web-Services



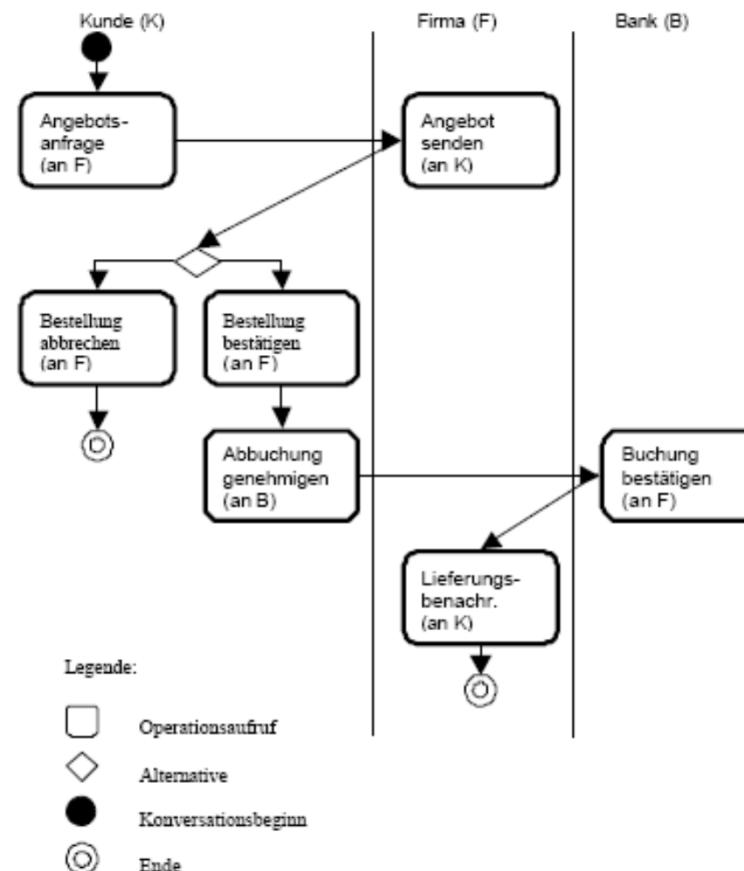
Einordnung der Nutzung von Web-Services im WEB



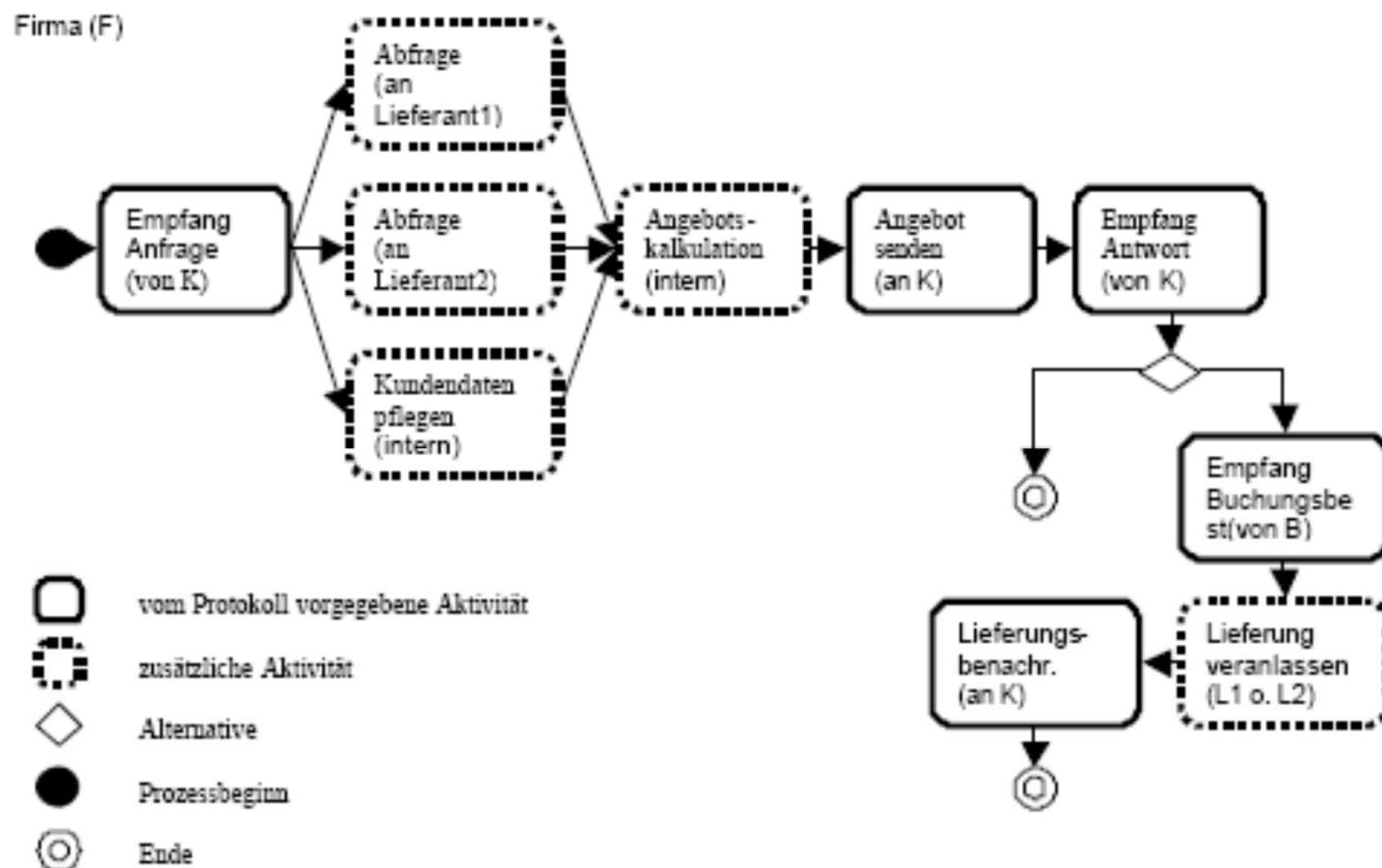
Konversation – Nachrichtenabfolge mit drei Partnern



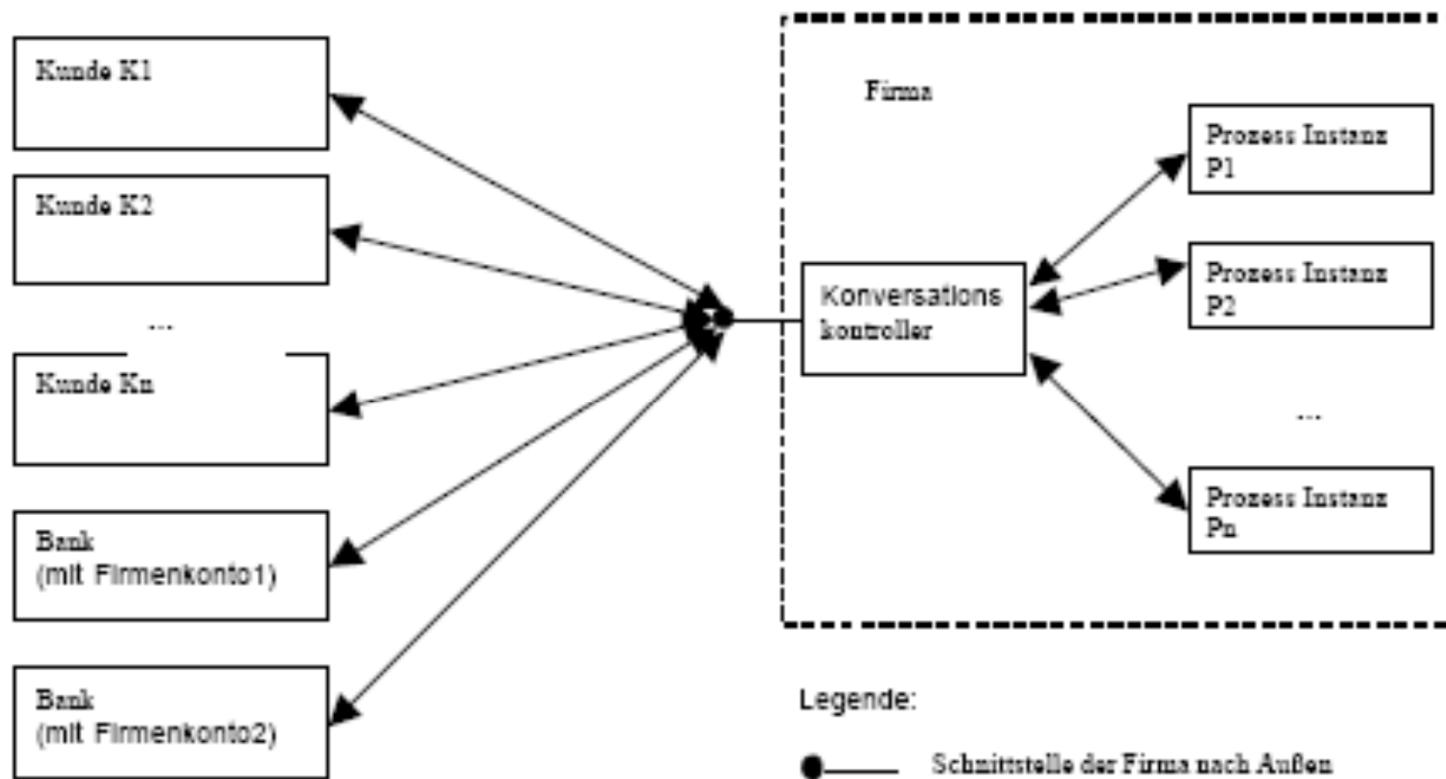
Choreographie - Koordinationsprotokoll



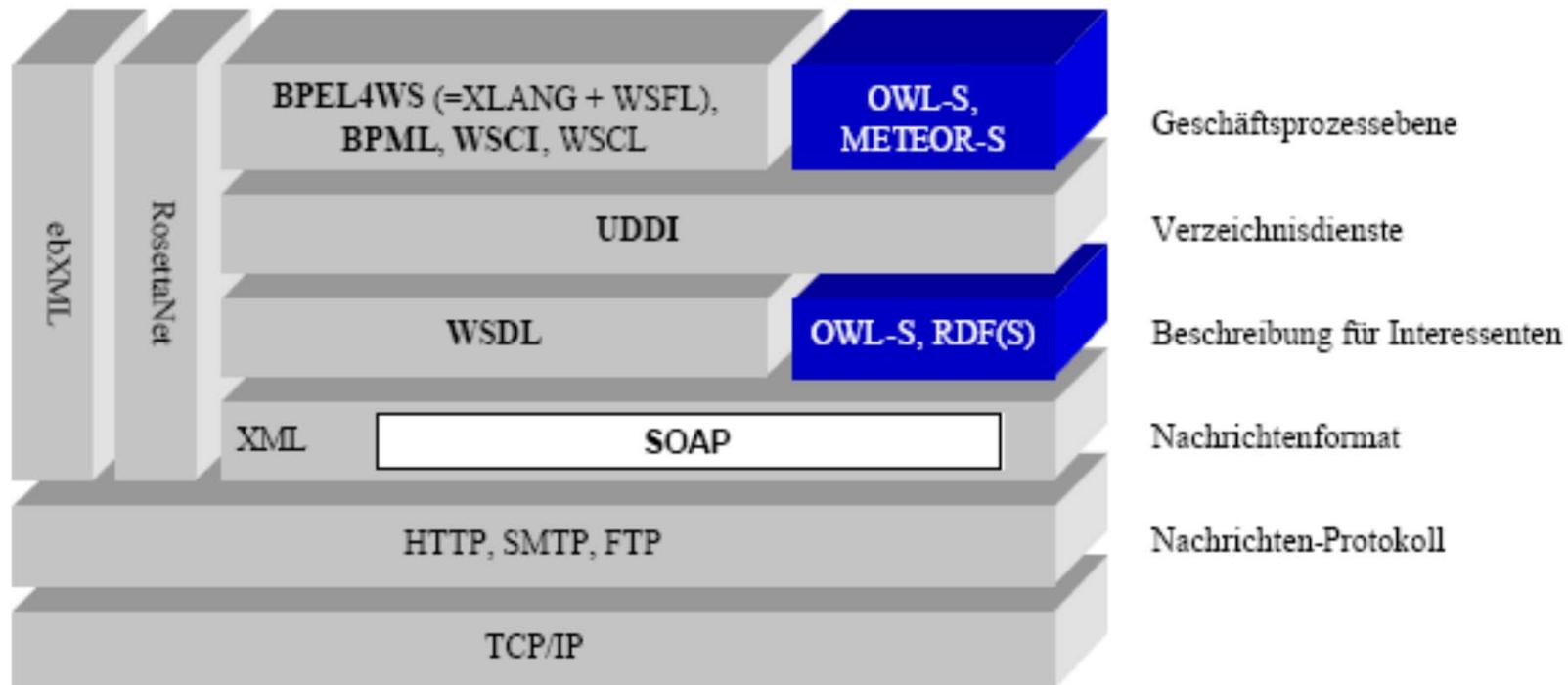
Orchestrierung von Web-Services



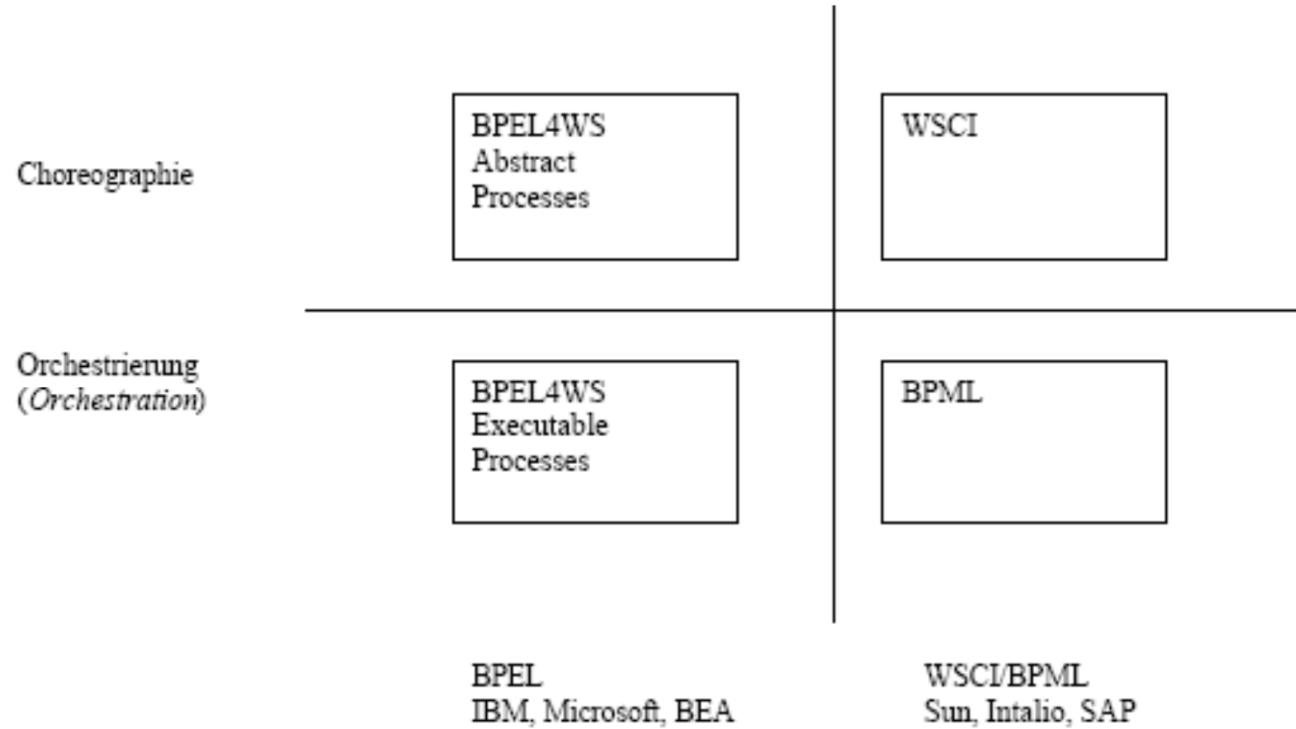
Konversationsunterstützung



Web Service Stack mit Semantikunterstützung



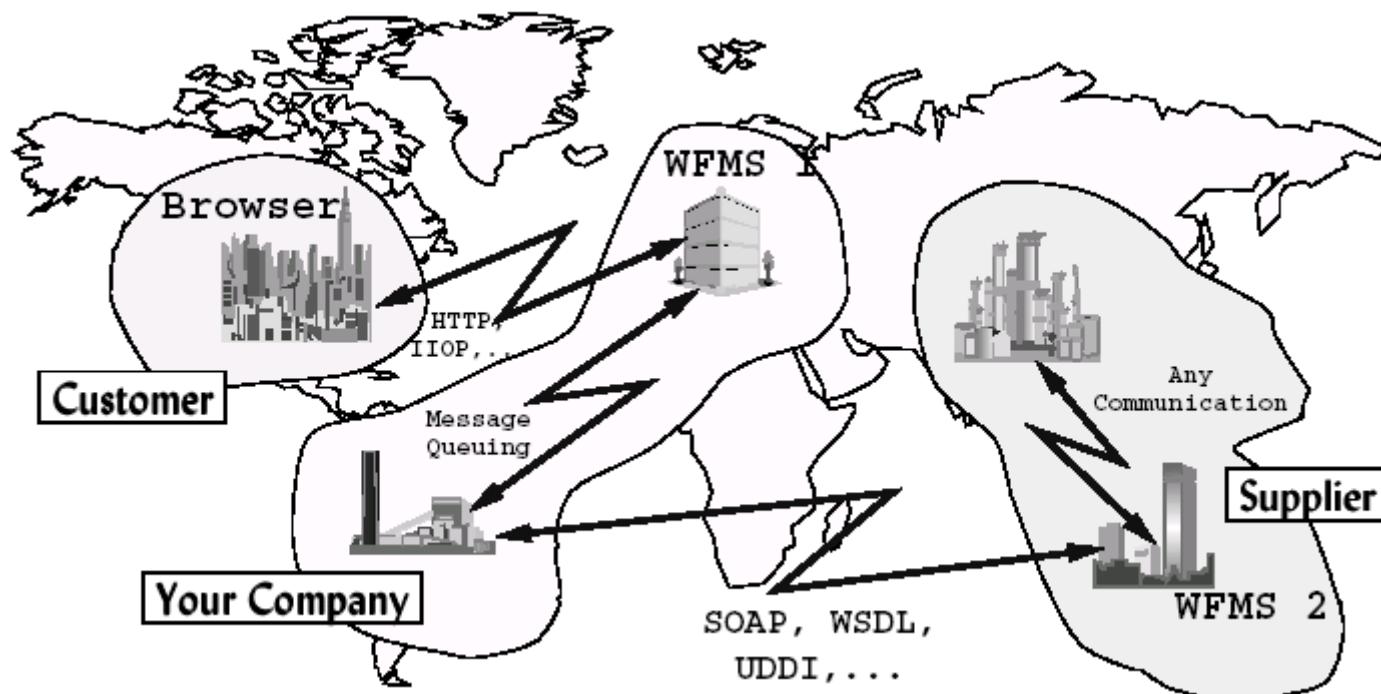
Standards für Workflowsprachen im Web



Standards für Workflowsprachen im Web

Choreographie	WS-BPEL Abstract Processes	WSCI	CDL BPMN
Orchestrierung	WS-BPEL Execut. Processes	BPML BPMN 1.0	BPMN 2.0 BPEL 2.0
Austausch- formate			XPDL (Prozesse) WSDL (WS- Interaktion)
	BPEL IBM, Microsoft, BEA	WSCI/BPML Sun, Intalio, SAP	Aktuelle Standards

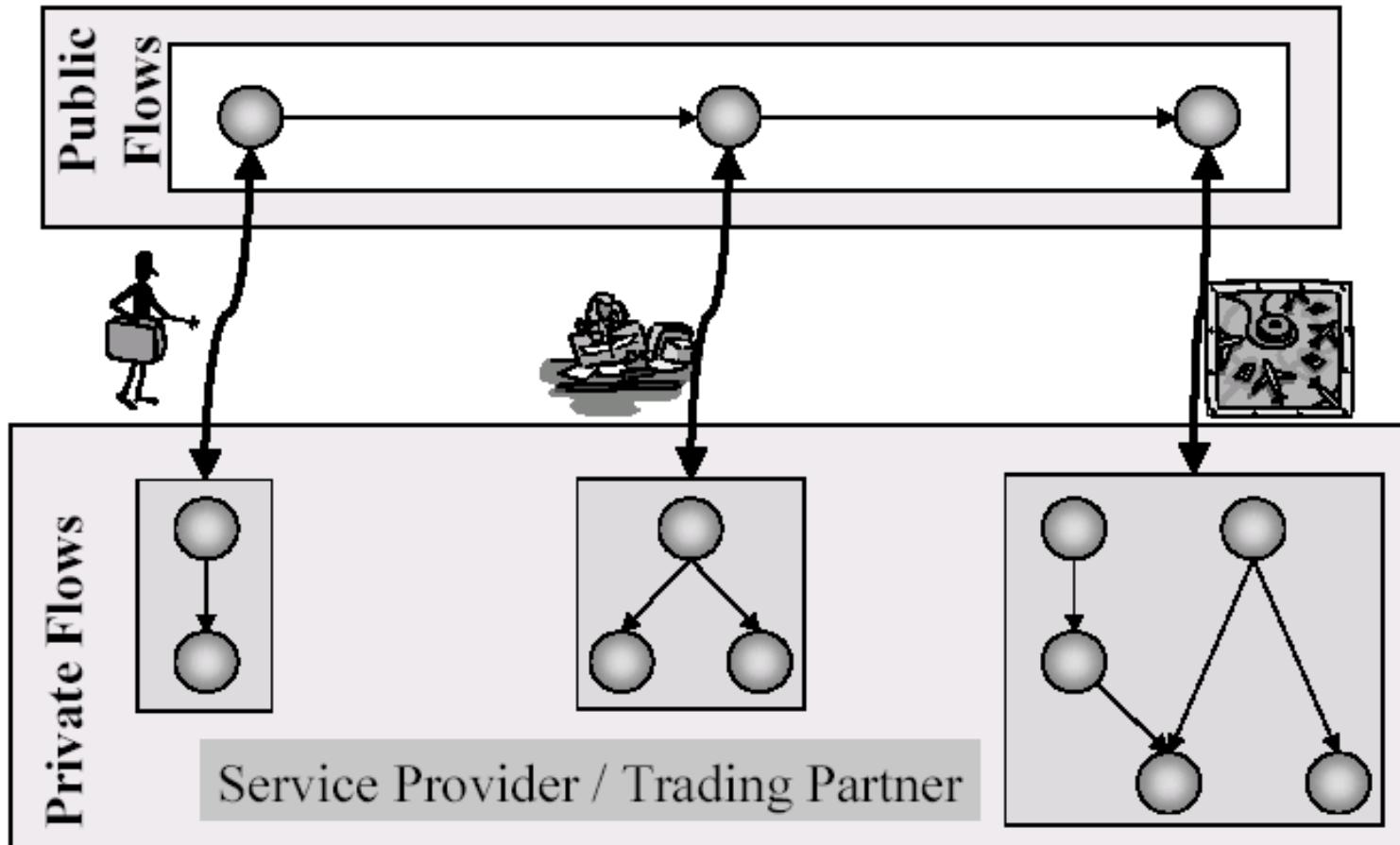
Virtual Enterprise: Scenario



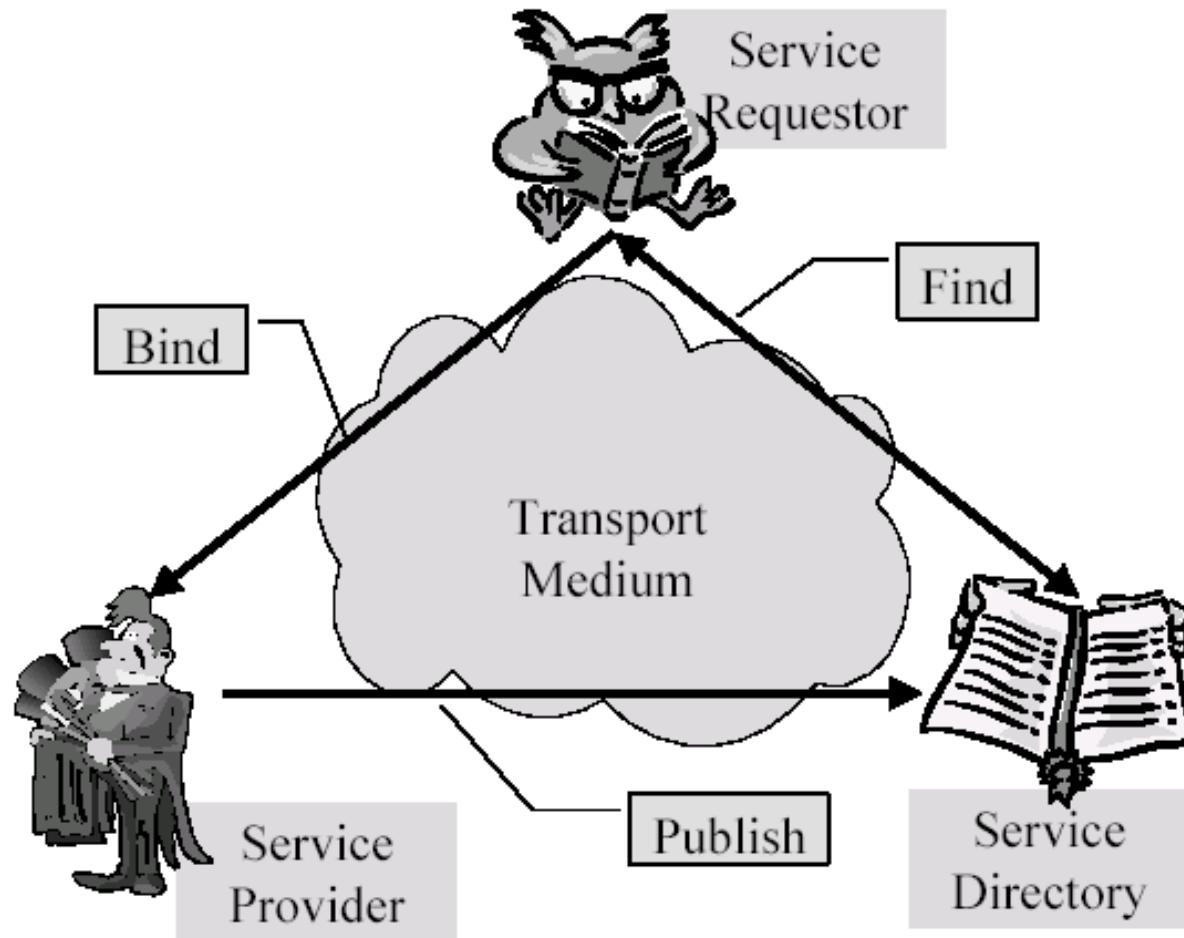
Workflows zwischen Partnern

- ◆ Es ist aufwändig, manuell die Verarbeitung bei verschiedenen Partnern zu überwachen und zu protokollieren und auf Ausnahmen zu reagieren.
- ◆ Workflow-Technologie kann helfen!
 - Modelliere reguläre Partner-Interaktionen
 - „Conversations“ sind unterschiedlich zum nicht verteilten Fall (!!!)
 - „Einzelner“ Nachrichtenaustausch zwischen zwei Partnern, um eine einzige Interaktion im Gesamtablauf zu vervollständigen
 - Spezifizierte die Ausnahmebehandlung
 - Time-out
 - nicht bearbeitete Anfragen
 - ...

Workflow-Typen



Überblick über beteiligte Stellen in einer SOA



Schlussfolgerungen

- ◆ ...Dienste (services) „outsourcen“ (z.B. was nicht direkt zu den Unternehmenskompetenzen zählt)
- ◆ ... Dienste anbieten (z.B. um Geld mit Funktionen & Prozessen zu verdienen, die man für andere durchführt)
- ◆ Diese Dienste umfassen
 - ...einfache, fein granulare Funktionen...
 - ...bis zu kompletten Geschäftsprozessen, d.h. grob granulare Funktionen

Workflow Technologie ist das Herz davon!

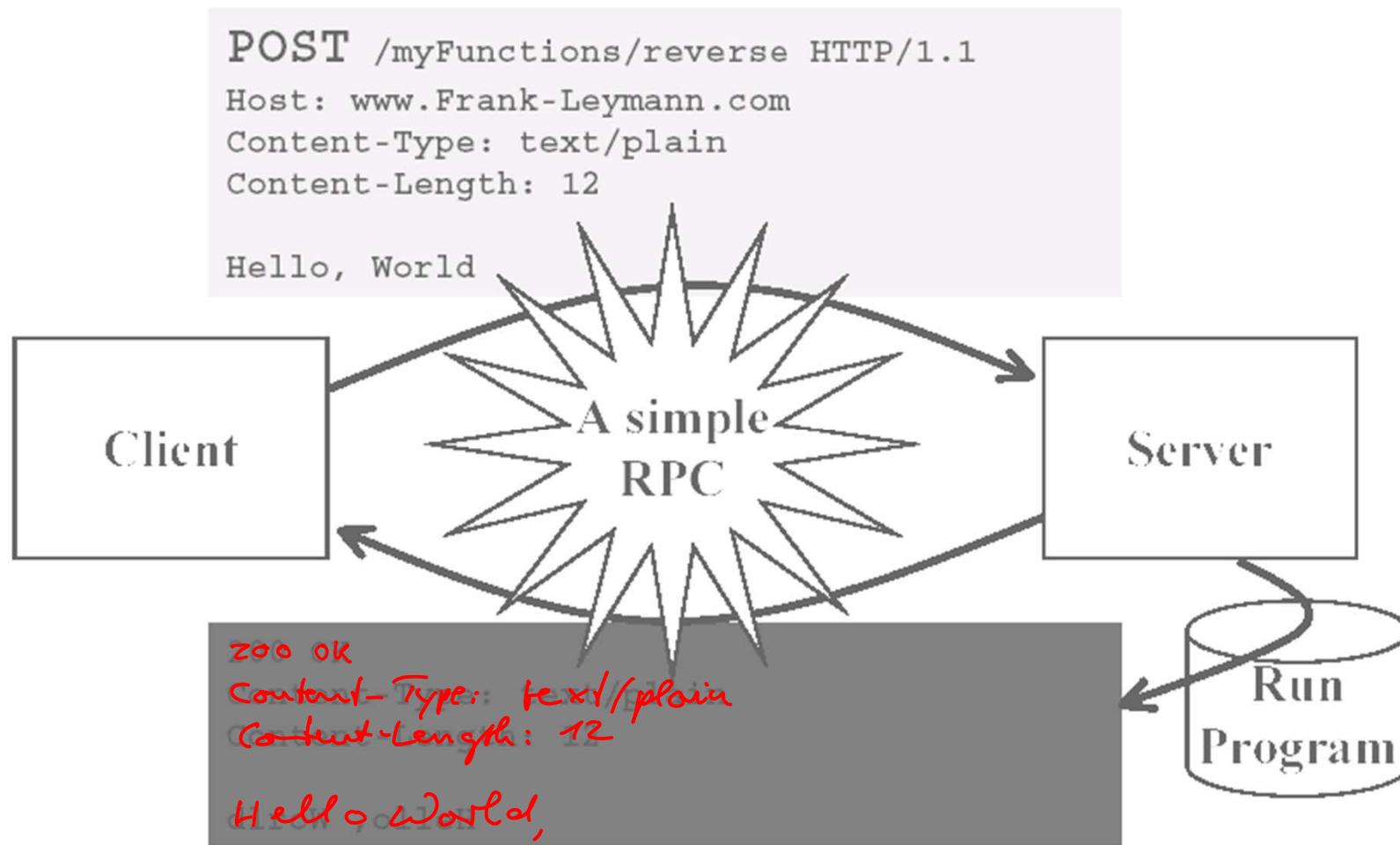
SOAP

Wie nutzt man Dienste (Web Services)?

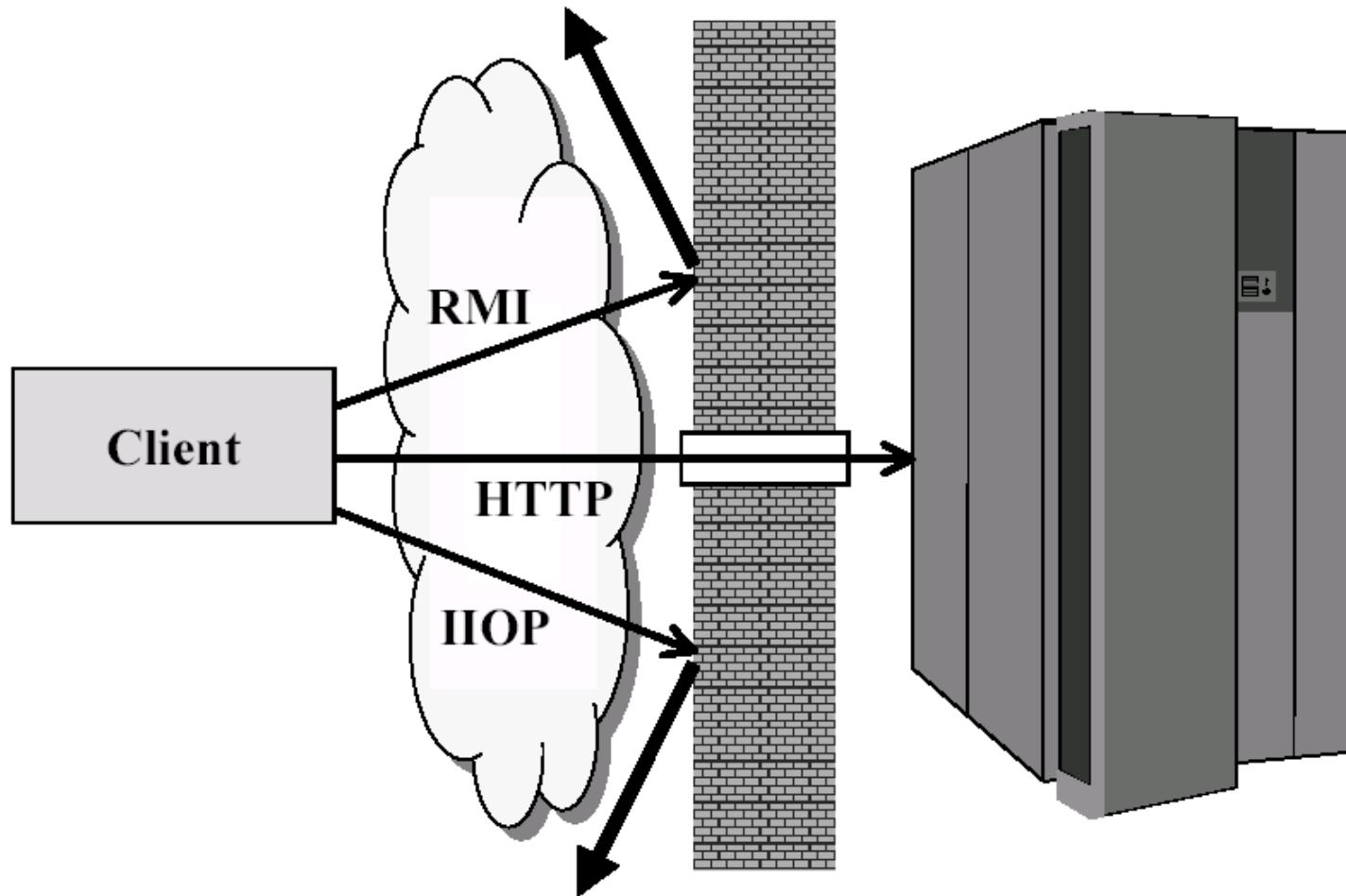
- ◆ Firewalls stellen Hindernisse dar für den unbeschränkten (ubiquitous!) Zugang zu Diensten
- ◆ Die Vielfalt und Heterogenität von Programmiersprachen, Programmierparadigmen, Server-Umgebungen, etc. ... machen die Orchestrierung zu einem Albtraum
- ◆ ...

Wir brauchen einen global verfügbaren Aufrufmechanismus!: **SOAP**

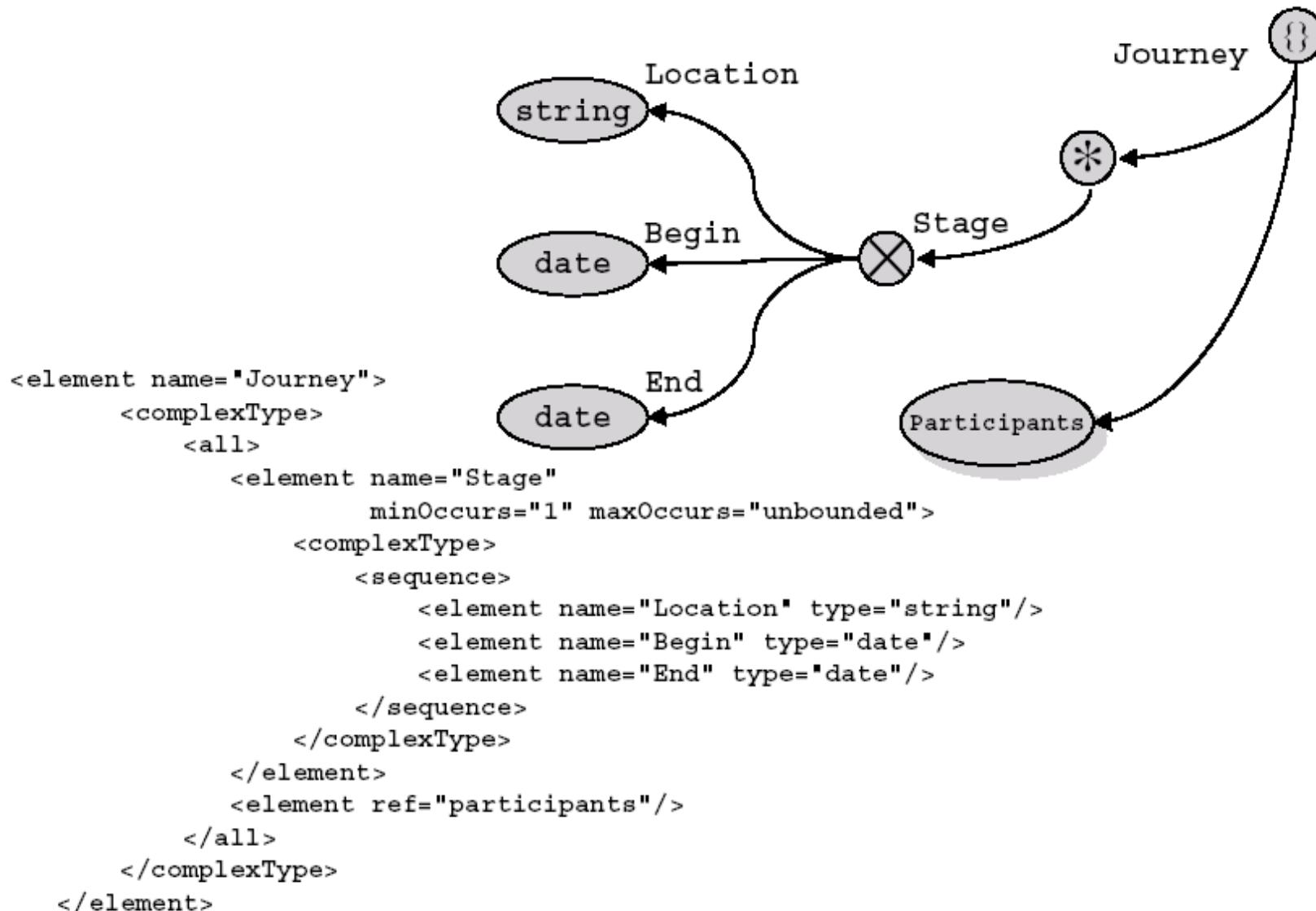
HTTP POST: Aufruf-Code



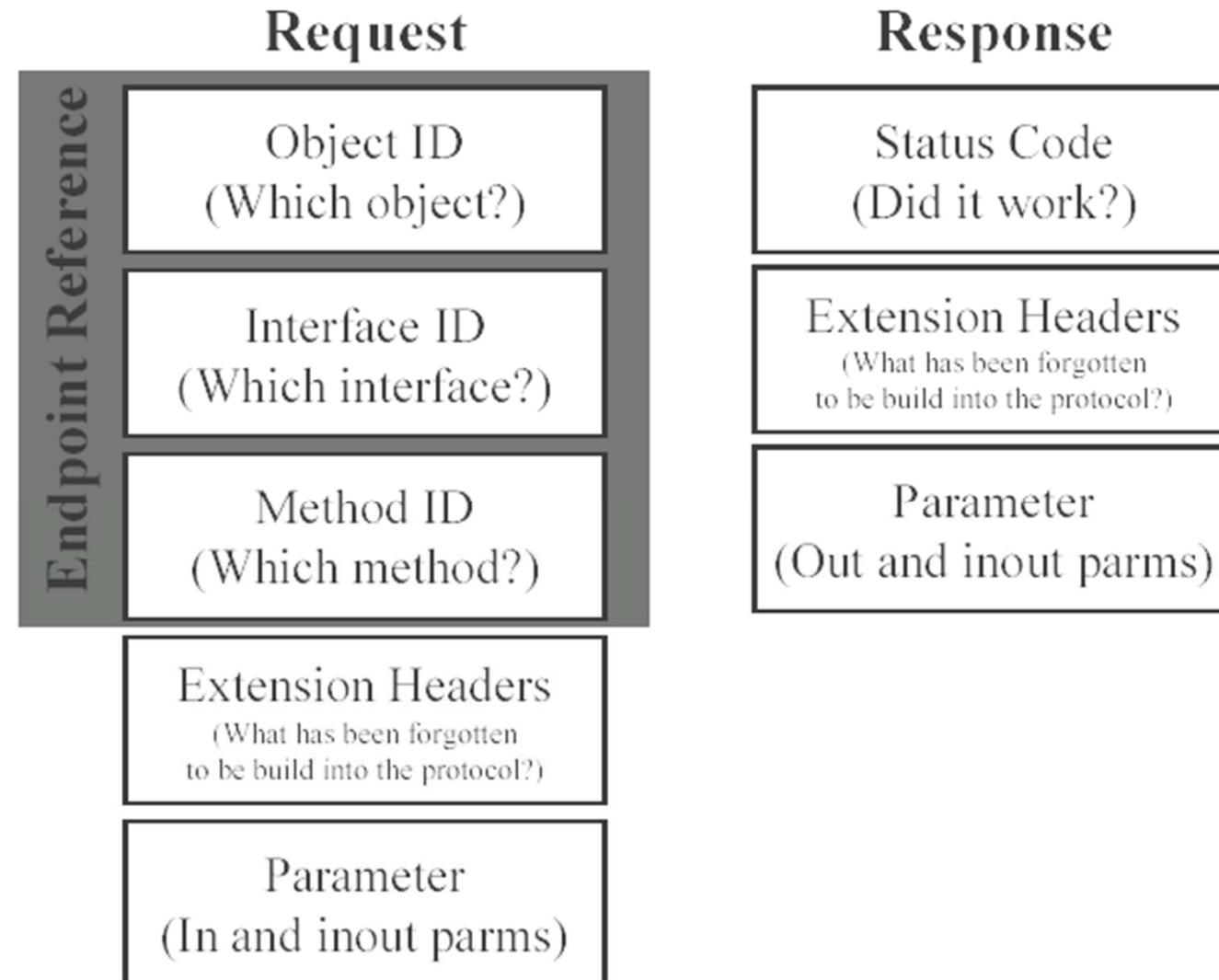
Die Firewall „durchtunneln“



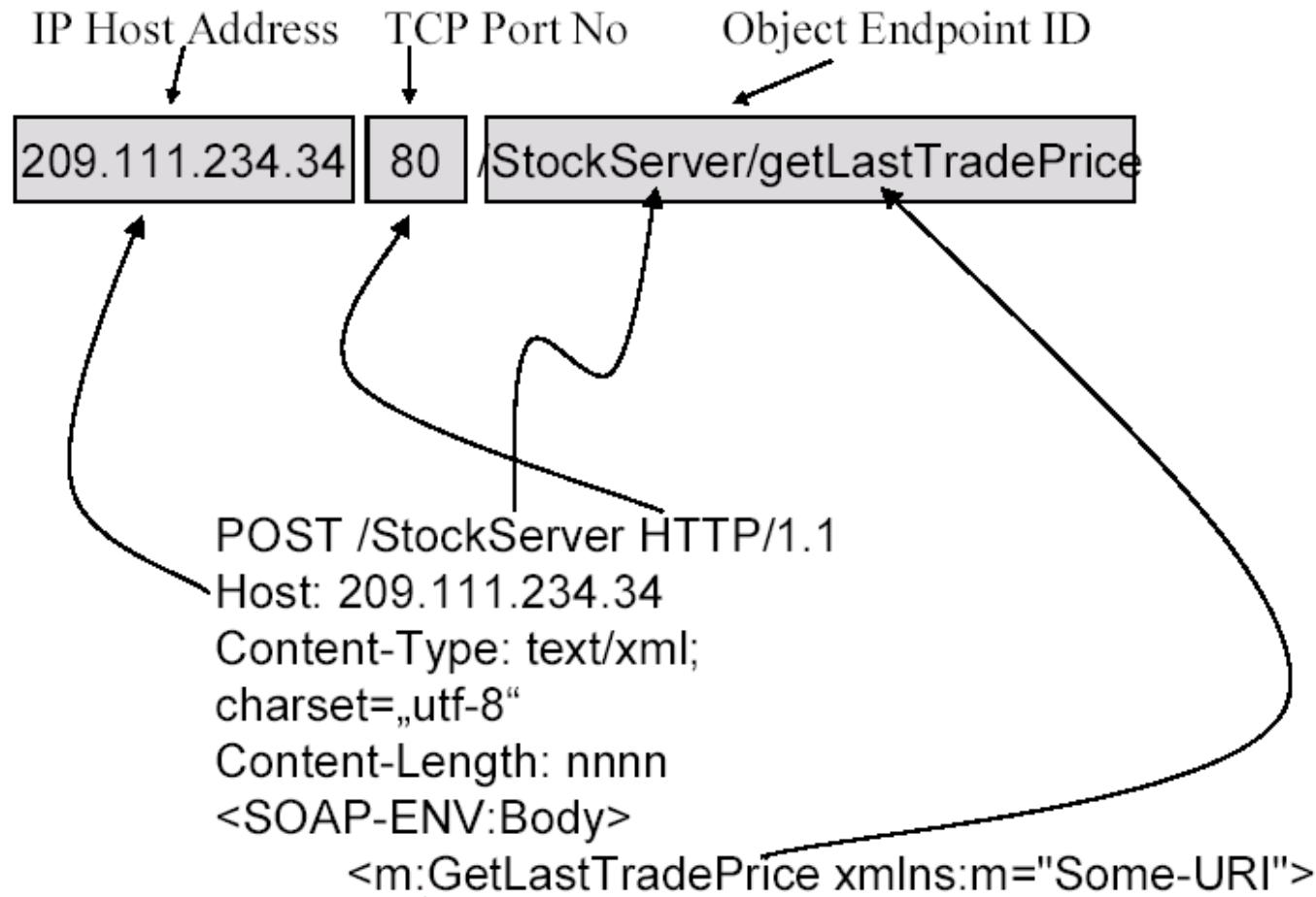
Ein XML-Schema-Beispiel



SOAP Request/Response Struktur



SOAP Endpunkt-Referenz



Ein einfacher SOAP-RPC

POST /StockQuote HTTP/1.1

Host: www.stockquoteserver.com

Content-Type: text/xml;

charset="utf-8",

Content-Length: nnnn

Method

Name

```
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/
    SOAPENV:
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
        <m:GetLastTradePrice xmlns:m="Some-URI">
            <symbol>DIS</symbol>
        </m:GetLastTradePrice>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Eine einfache SOAP-Antwort

HTTP/1.1 200 OK

Content-Type: text/xml;
charset="utf-8,"
Content-Length: nnnn

```
<SOAP-ENV:Envelope  
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/  
    SOAPENV:  
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
    <SOAP-ENV:Body>  
        <m:GetLastTradePriceResponse xmlns:m="Some-URI">  
            <Price>34.5</Price>  
        </m:GetLastTradePriceResponse>  
    </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

SOAP-Header

POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml;
charset="utf-8,"
Content-Length: nnnn
SOAPAction: „Some-URI“

```
<SOAP-ENV:Envelope xmlns:SOAPENV="  
http://schemas.xmlsoap.org/soap/envelope/"  
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
    <SOAP-ENV:Header>  
        <t:Transaction xmlns:t="some-URI", SOAP-ENV:mustUnderstand="1">  
            5  
        </t:Transaction>  
    </SOAP-ENV:Header>  
    <SOAP-ENV:Body>  
        <m:GetLastTradePrice xmlns:m="Some-URI">  
            <symbol>DEF</symbol>  
        </m:GetLastTradePrice>  
    </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

Protokoll Erweiterungen

Katalogdienste

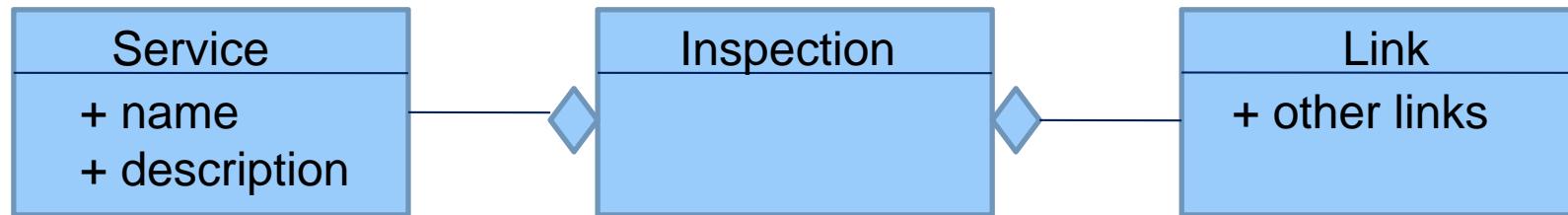
- ◆ WS-Inspection
- ◆ UDDI – Universal Description, Discovery & Integration
- ◆ WS-Discovery
 - Lokale Suche (im eigenen Netz) über Multicast
 - Einrichten von Discovery-Proxies möglich
 - Nutzbar für kleine Umgebungen mit nicht systematisch aufgebauten Verzeichnissen oder bei neuen Zusammenstellungen wie Adhoc-Netzen.
 - Für weitere Nutzung: Einsatz von z.B. SOAP und WSDL

WS-Inspection

- ◆ Einfache Schnittstelle.
- ◆ Kombinierbar mit Katalogdiensten, z.B. mit UDDI.
- ◆ Veröffentlichung der WS-Inspection Dokumente im Basisverzeichnis des Webservers des jeweiligen Unternehmens.
- ◆ Unterverzeichnisse / Hierarchien sind möglich
- ◆ Dynamische Definition der WS-Inspection-Hierarchie wird unterstützt, z.B. Generierung aus internen UDDI-Verzeichnissen oder anderen Verzeichnissen.

WS-Inspection-Datenmodell

- ◆ Inspection-Klasse als Wurzel des Katalogs.
- ◆ Daran geknüpft eine Menge von
 - Services (mit Name und Beschreibung) und
 - Links (z.B. zu Inspection-Unterverzeichnissen oder UDDI-Verzeichnissen)



Wie findet man Dienste?

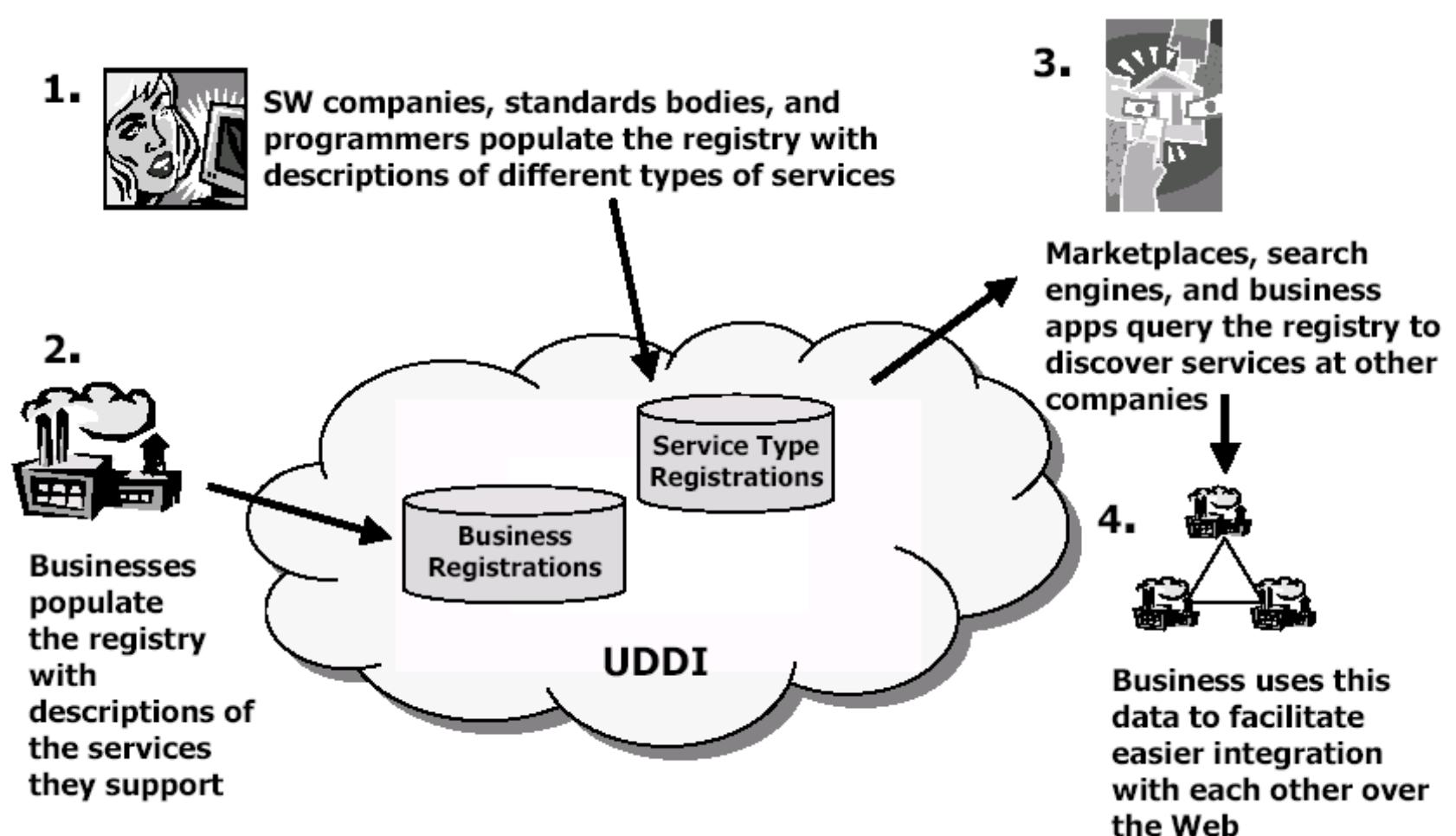
Wir brauchen ein (global verfügbares) Verzeichnis!

- ◆ ...um Dienste basierend auf „öffentlichen“ Anfragen von Service-Providern zu katalogisieren
- ◆ ...um Taxonomien zur Unterstützung der Suche von geeigneten Diensten zu unterstützen
- ◆ ...um Dienste durch menschliche Benutzer oder Programme suchen zu können.

UDDI

(Universal Description, Discovery & Integration)
bietet diese Funktionalität

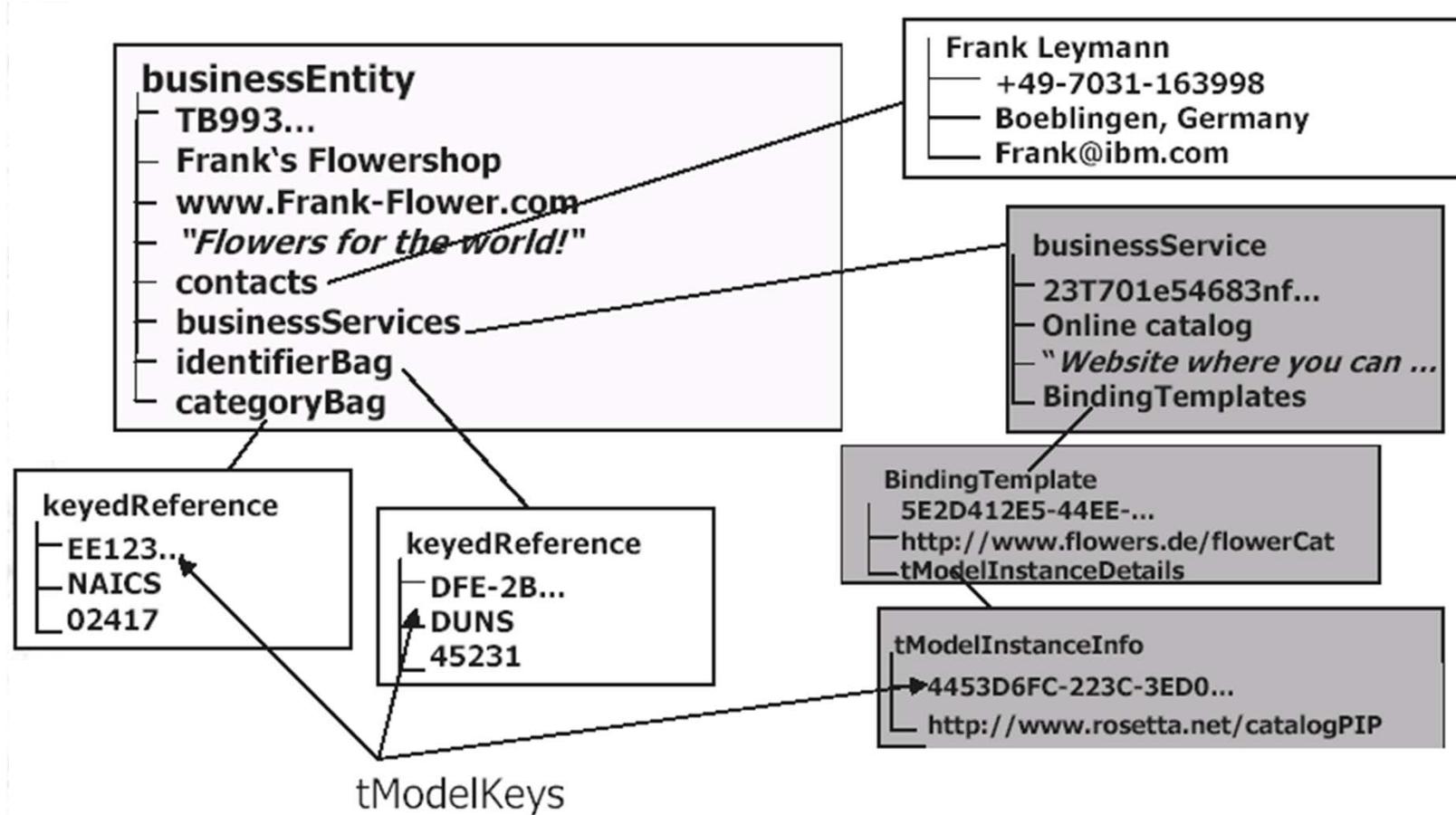
Wie arbeitet UDDI



Registry-Daten für UDDI

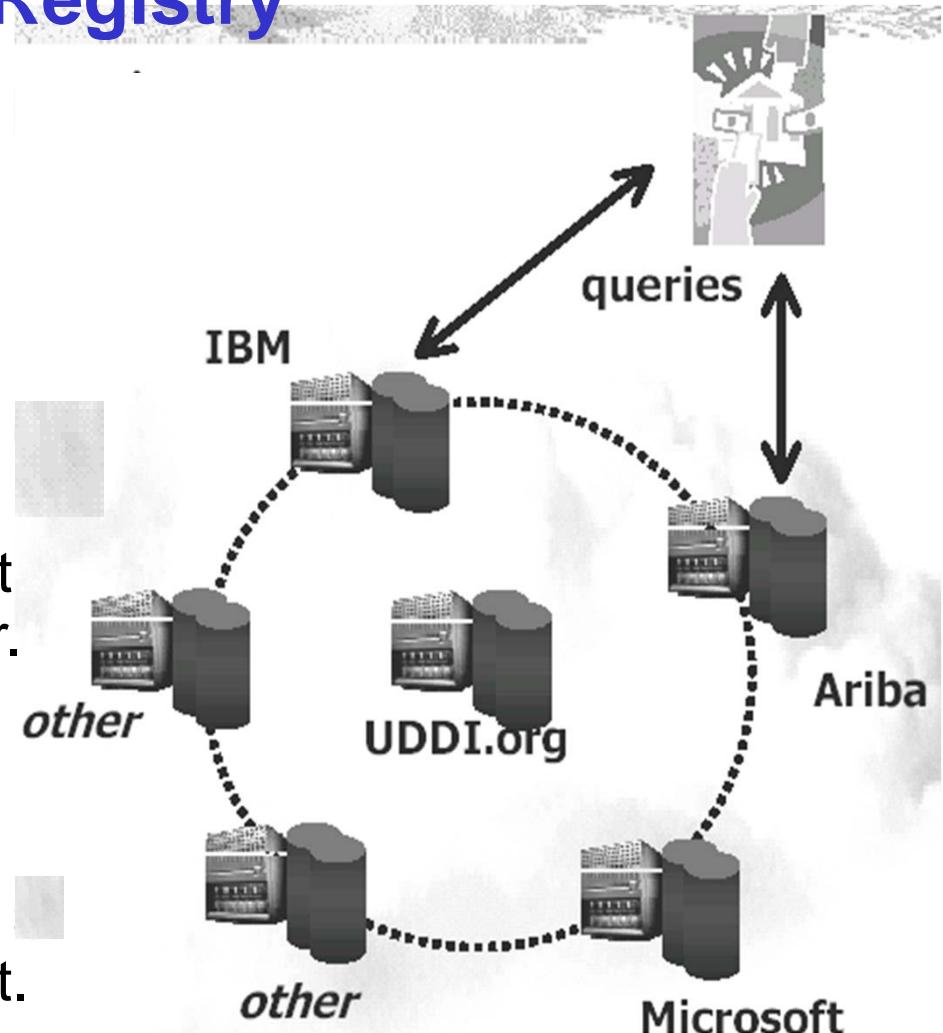
- ◆ Firmen registrieren öffentliche Informationen über sich: UBR „UDDI Business Registry
 - Öffentlicher Verzeichnisdienst
 - Betrieben von IBM, Microsoft, NTT-Communications und SAP
 - Eingestellt Januar 2006, da nur noch SAP als Knoten (Standardisierungsstelle) verblieb.
 - Nachweis der Machbarkeit bzgl. Funktionalität war erbracht
- ◆ Standardisierungsstellen, d.h. Programmierer, Firmen
 - registrieren Informationen
 - über ihre Service-Typen („tModels“).

Beispiel einer Registrierung



Registrierung – globales UBR „UDDI Business Process Registry“

- ◆ Peer-Knoten (Web Sites).
- ◆ Firmen über einen Knoten registrieren.
- ◆ Registrierungen werden täglich repliziert.
- ◆ Die komplette Menge an “registrierten” Einträgen ist auf allen Knoten verfügbar.
- ◆ Gemeinsame Menge von SOAP APIs wird auf allen Knoten unterstützt.
- ◆ Die Befolgung wird durch Geschäftsverträge erreicht.



WSDL

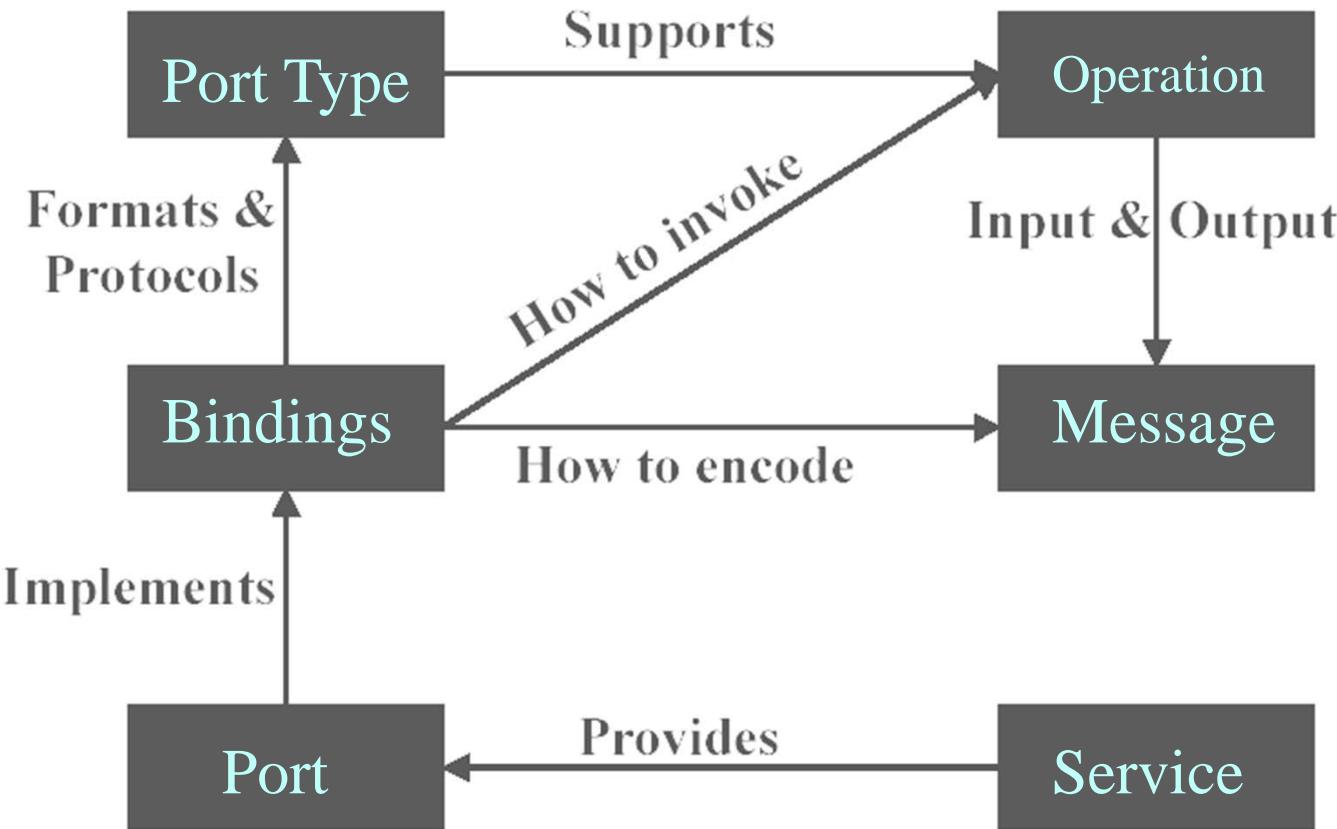
Wie definiert man neue Web Services?

- ◆ Es ist gut, Web Services zu nutzen, die andere definiert haben und über die man sich geeinigt hat (= tModels),
- ◆ ...aber: was, wenn man seine eigenen Web Services veröffentlichen will
 - z.B. einfache Funktionen, die auch für andere nützlich sind
 - oder komplette Geschäftsprozesse, die man anbieten kann.
- ◆ ...ohne einen Standard zu definieren!

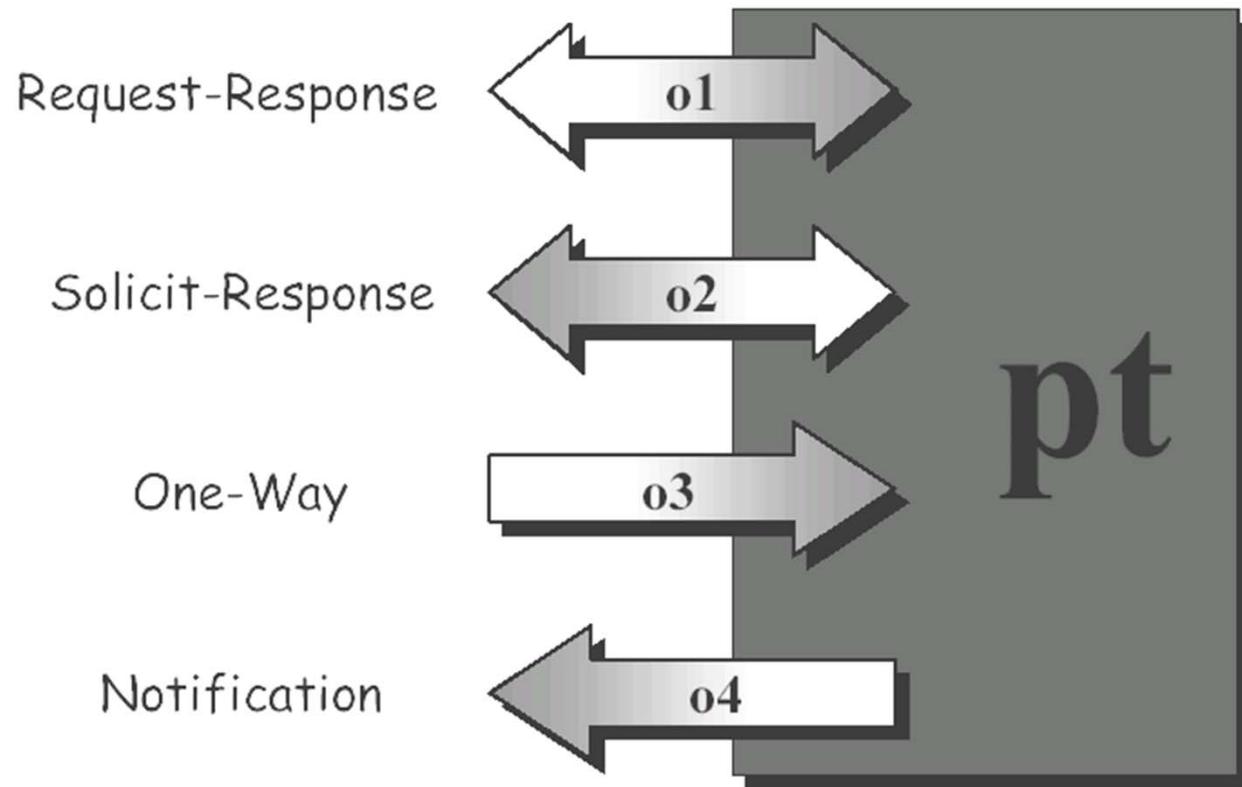
Benötigt wird ein Sprache zur *Spezifikation* von Diensten und zur Beschreibung, wie andere die Dienste koppeln (*Binding*):

WSDL (Web Services Description Language)

WSDL Bestandteile



Port Typen: graphische Darstellung



Beispiel - Porttypen

```
<xml version="1.0"?>
<definitions name = "StockQuoteService"
  xmlns:tns=http://muelle.com/stockquote.wsdl
  xmlns:xsd=http://muelle.com/stockquote.xsd
  xmlns:soap=http://schemas.xmlsoap.org/wsdl/soap/
  xmlns=http://schemas.xmlsoap.org/wsdl/"/>
    <message name="GetLastTradePriceRequest">
      <part name="tickerSymbol" element="xsd:string"/>
      <part name="time" element="xsd:dateTimeInstant"/>
    </message>
    <message name="GetLastTradePriceResponse">
      <part name="result" type="xsd:float"/>
    </message>
    <portType name="StockQuotePortType">
      <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceRequest"/>
        <output message="tns:GetLastTradePriceResponse"/>
      </operation>
    </portType> ...
```

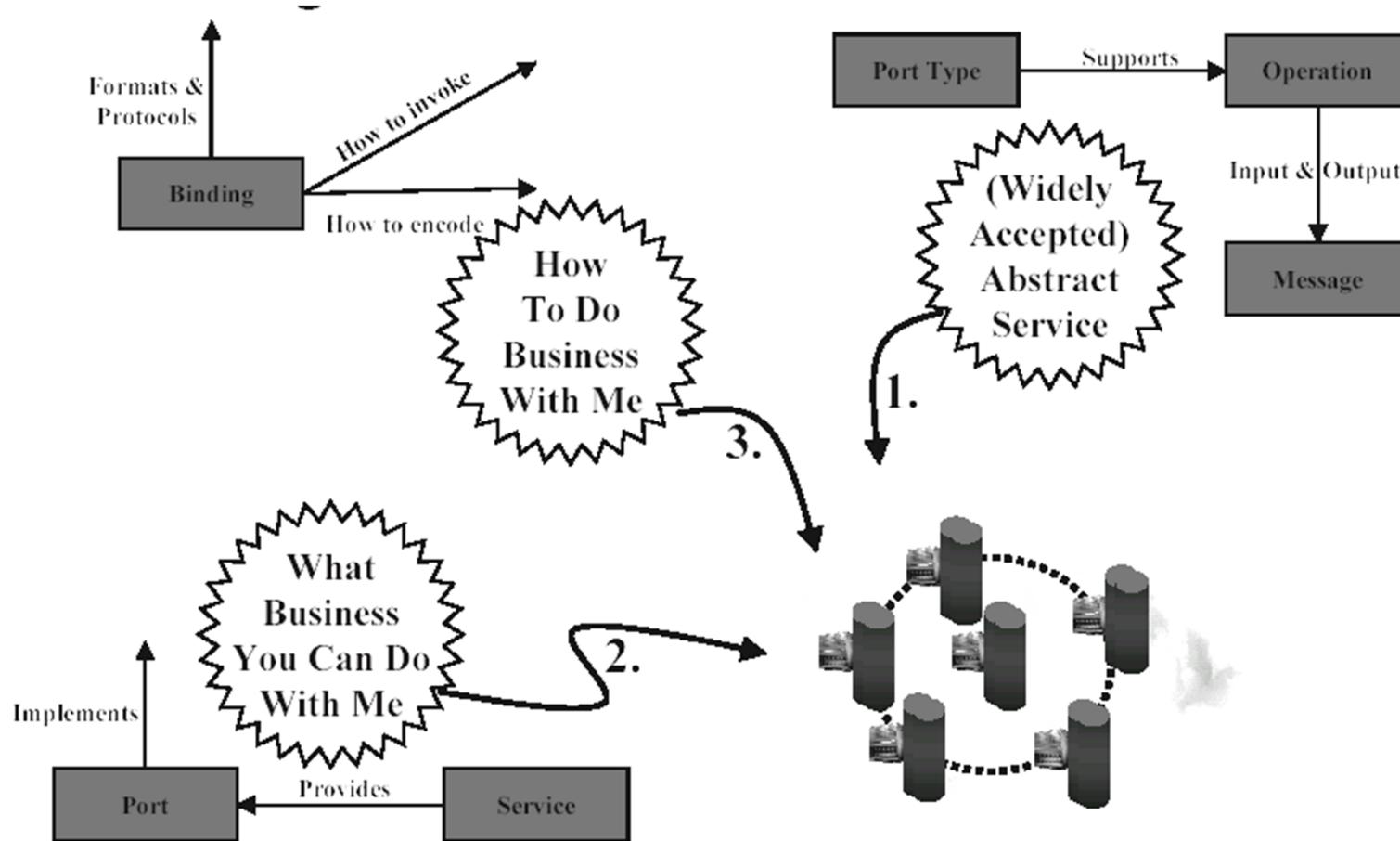
Beispiel - Porttypen

```
<binding name = "StockQuoteSoapBinding"
         type = "tns:StockQuotePortType">
<soap:binding
    style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http">
<operation name="GetLastTradePrice">
<soap:operation
    soapAction="http://muelle.com/GetLastTradePrice">
<input>
<soap:gody use="encoded"
    namespace="http://muelle.com/stickquote
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding">
</input>
<output>
<soap:body use="encoded"
    namespace="http://muelle.com/stockquote
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding">
</output> ...
```

Beispiel - Porttypen

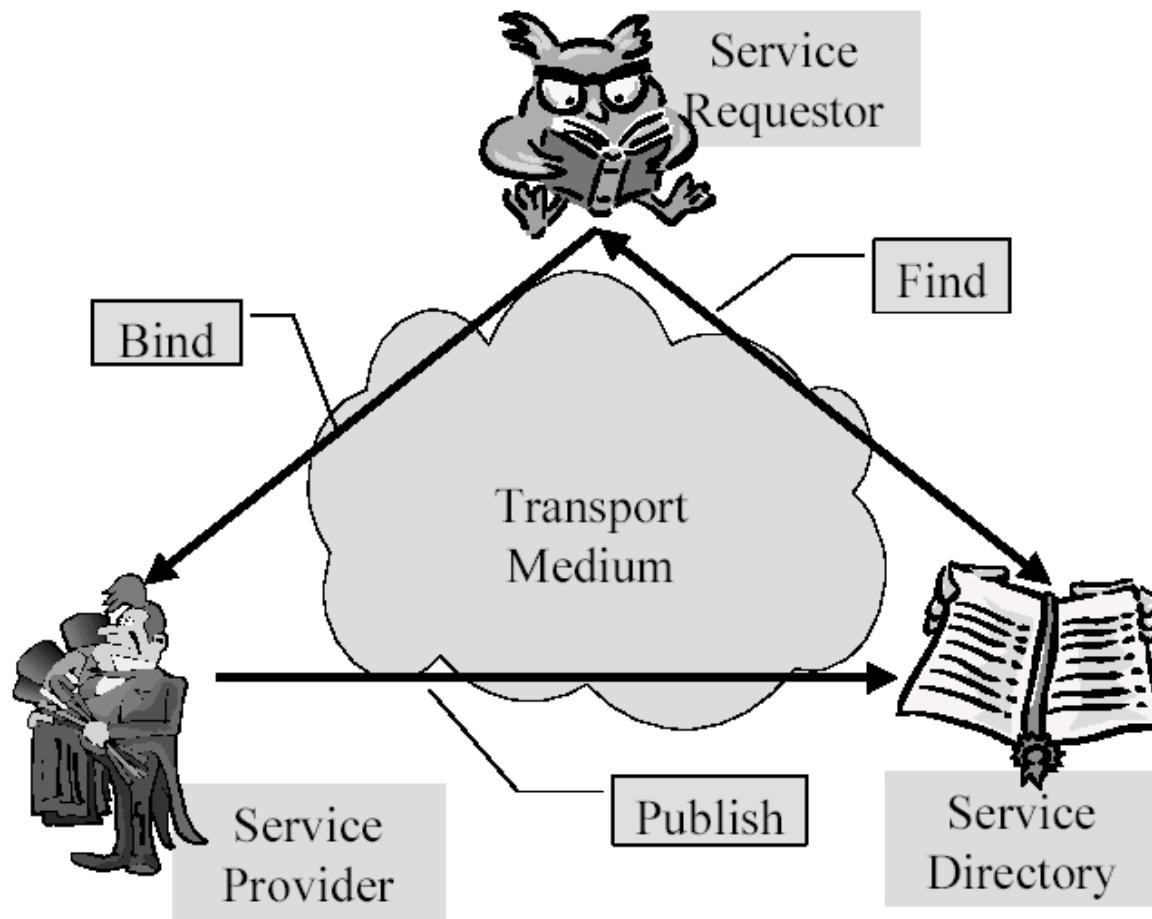
```
<service name = "StockQuoteService">  
  <documentation>My first service</documentation>  
  <port name= "StockQuotePort"  
        binding="tns:StockQuoteSoapBinding">  
    <soap:address  
        location= "http://muelle.com/stockquote">  
    </port>  
</service>
```

Wie macht man Dienste verfügbar?

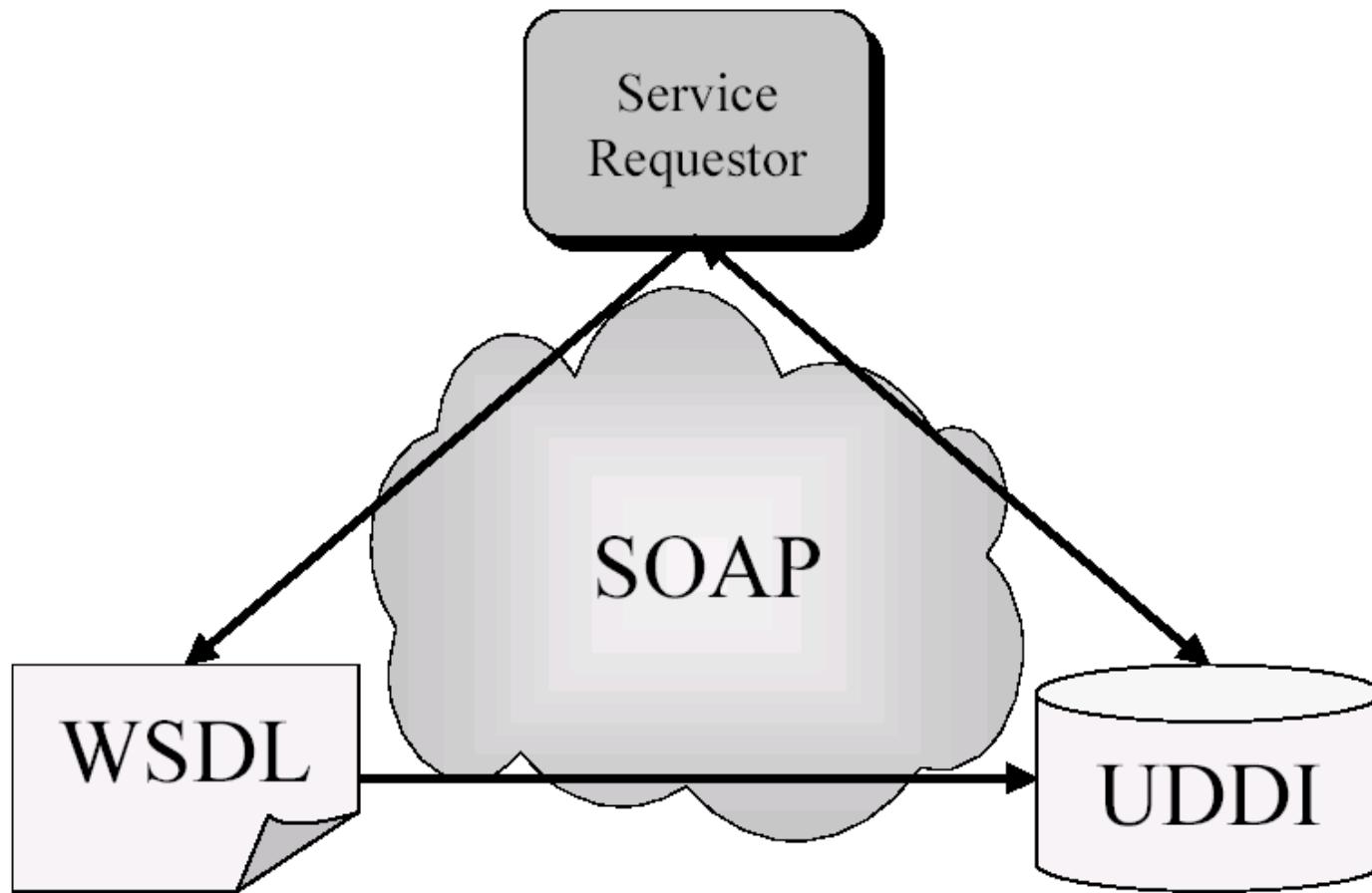


Zusammenhang mit der Service-Orientierten Architektur

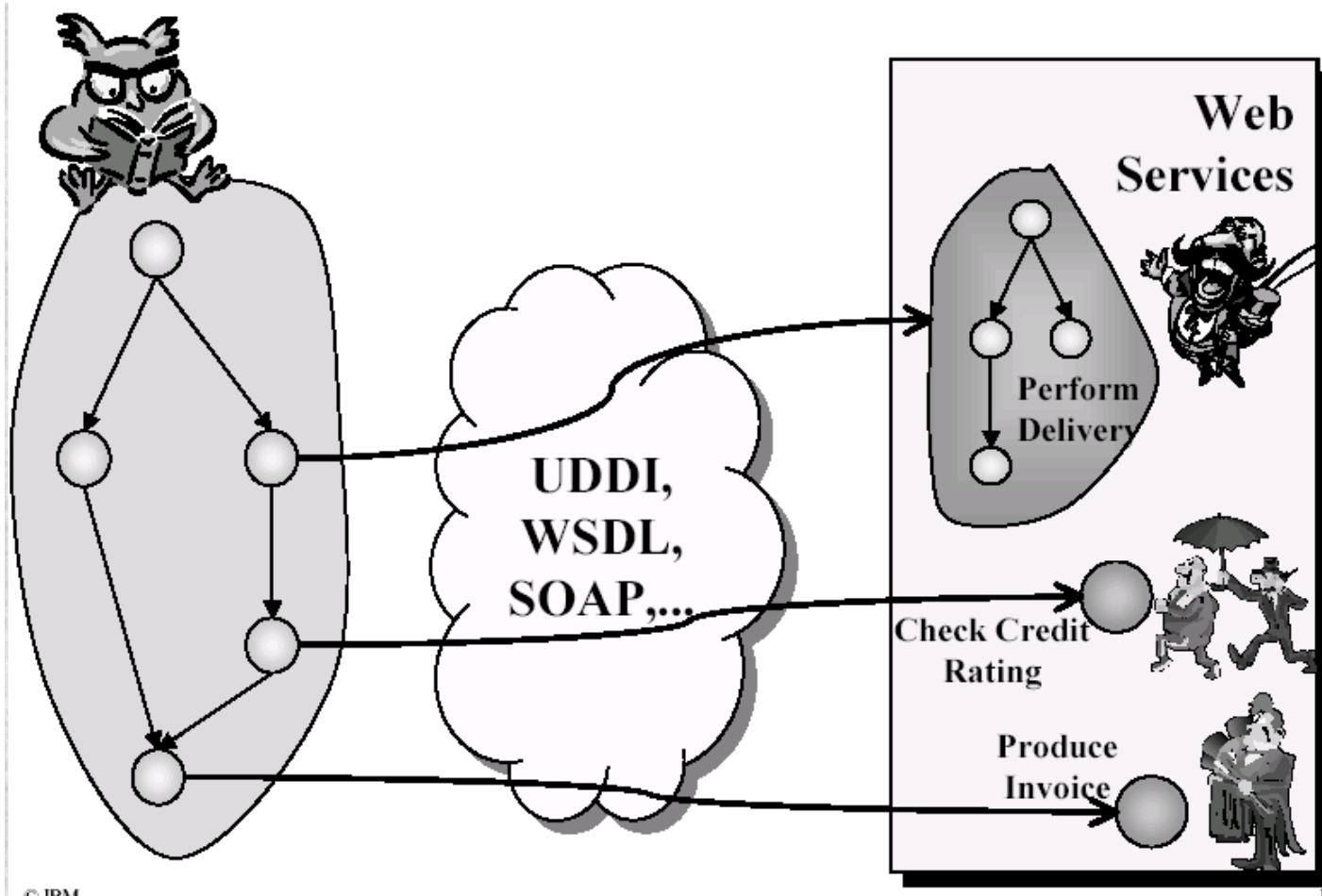
SOA (Wiederholung)



...und Zusammenhang mit Web Services



SOA in Aktion



Was wird benötigt?

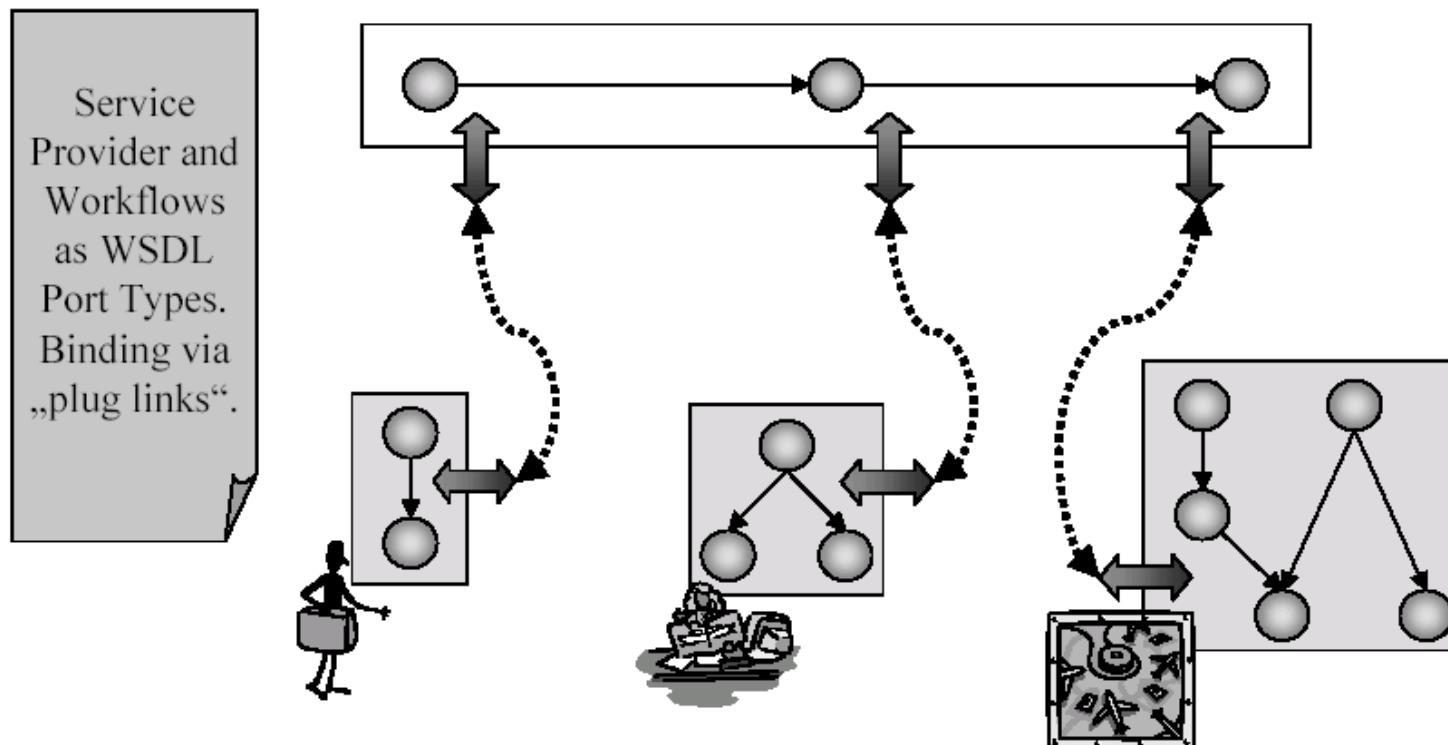
Wir brauchen eine Sprache die es erlaubt...

- ◆ Abläufe (flows) als Web Services zu spezifizieren
- ◆ Abläufe zwischen Web Services zu spezifizieren
- ◆ Neue Web Services als Aggregationen über Web Services zu spezifizieren
- ◆ ...eine beliebige Mischung davon zu spezifizieren

Standardisierungshistorie zur Web-Service-basierten Workflowmodellierung:

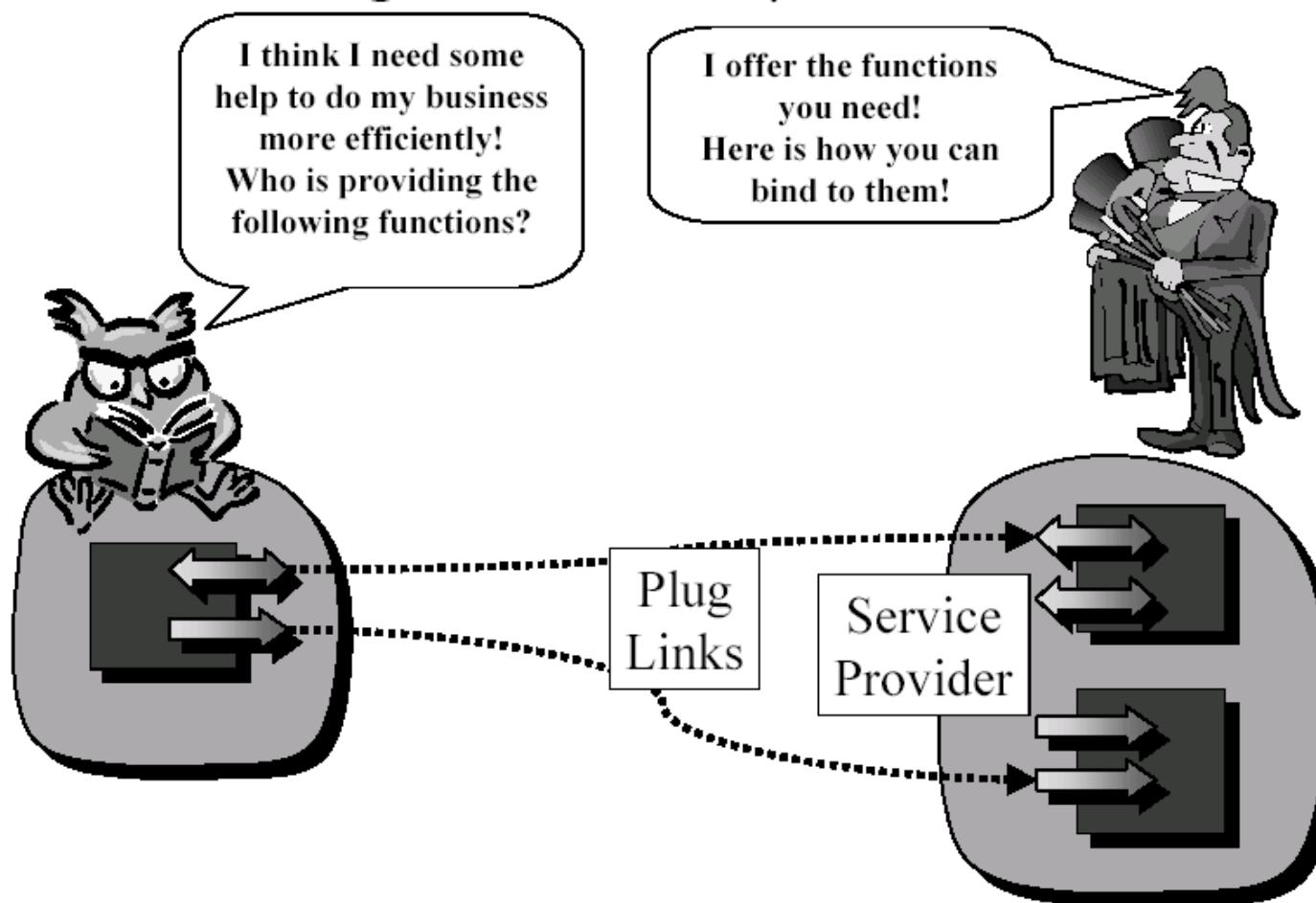
- ◆ WSFL (Web Services Flow Language) - IBM
- ◆ XLANG – Microsoft
- ◆ BPeL4WS (Business Process execution Language for Web Services) baut als Ansatz zur Standardisierung auf WSFL und XLANG auf.
- ◆ Aktueller Standard (aus BPeL4WS weiterentwickelt): **WS-BPEL**

Beziehung zur Service-orientierten Architektur

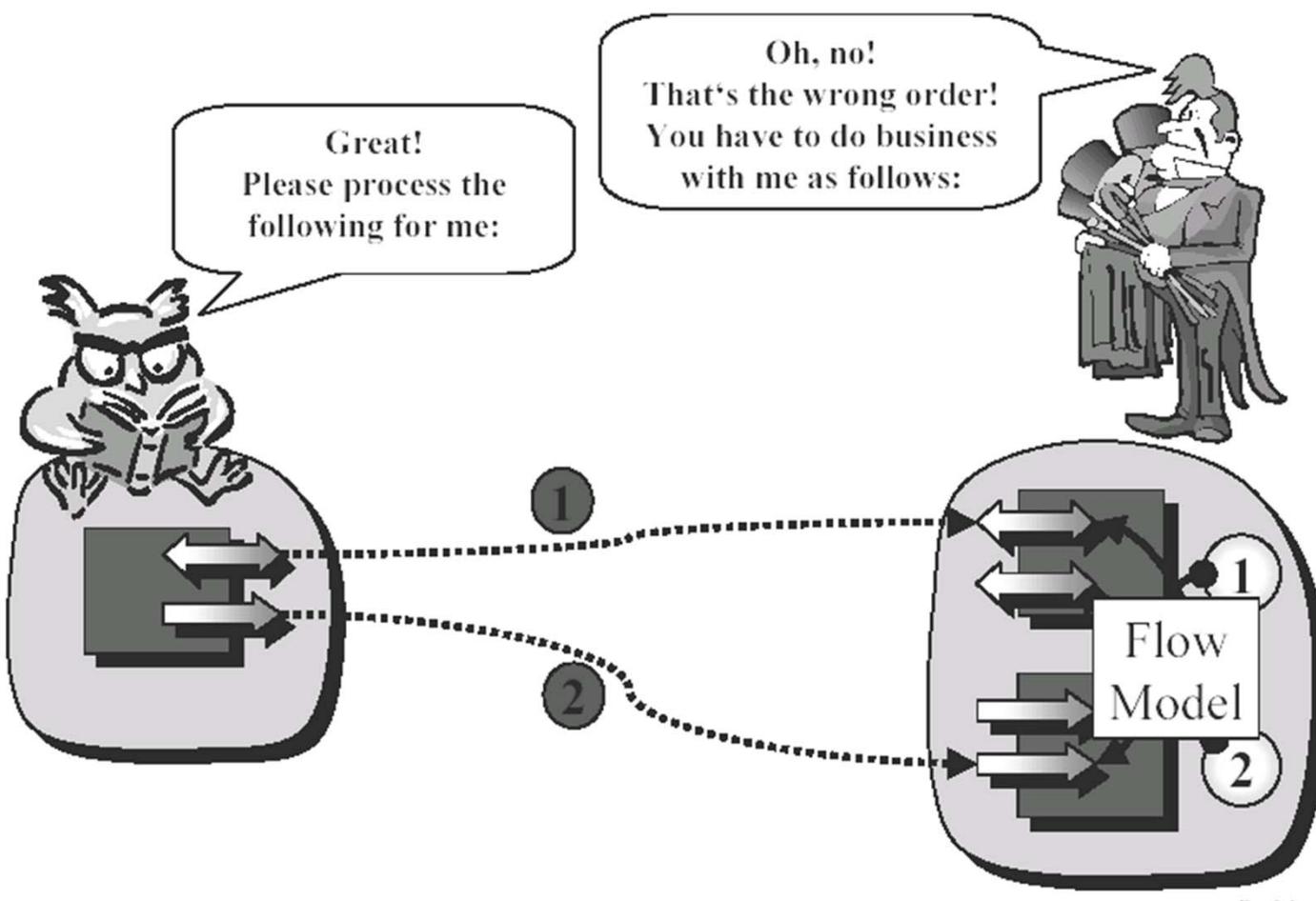


Eine abstraktere Sicht: High-Level-Scenarios

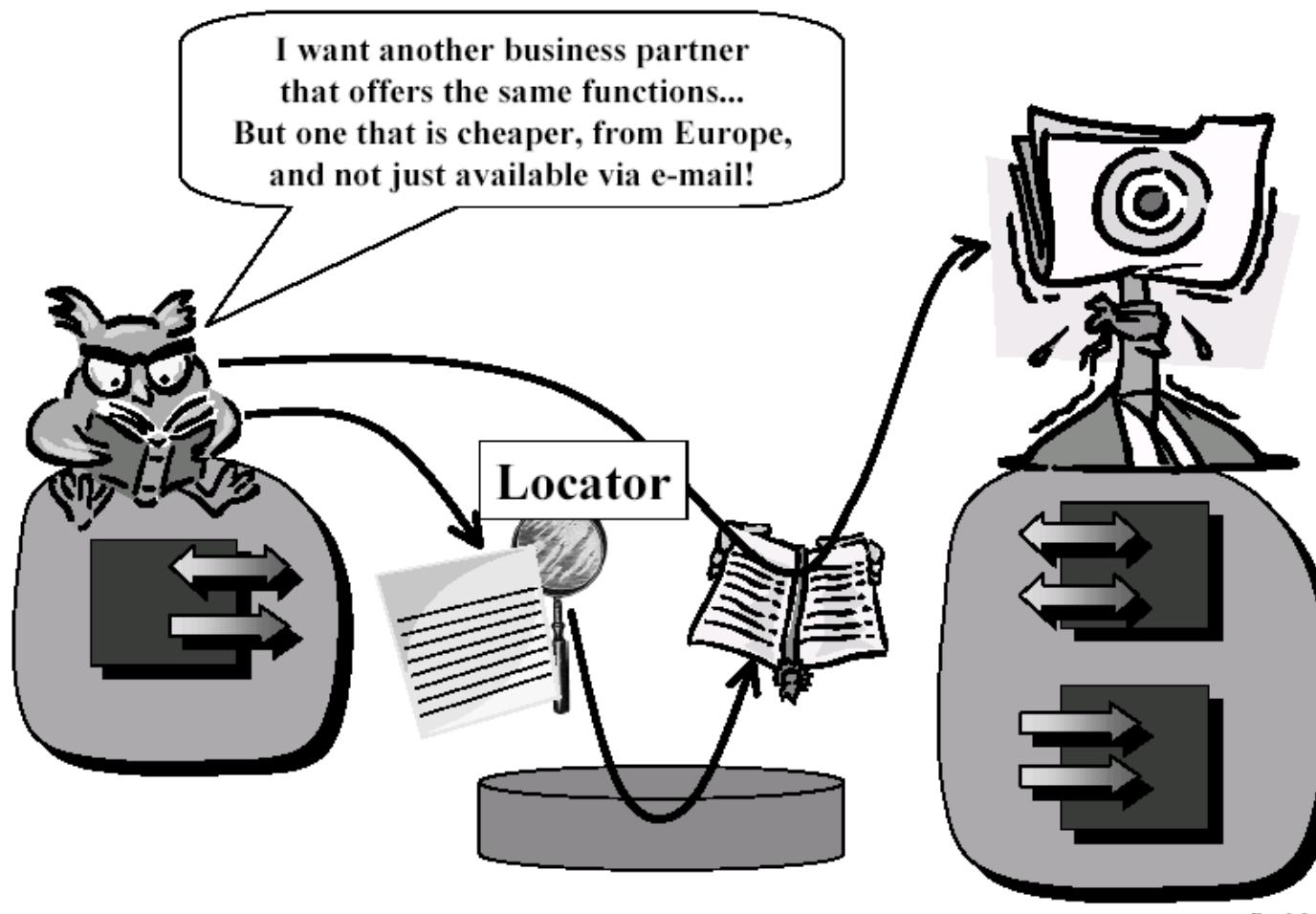
Partnerschaften errichten



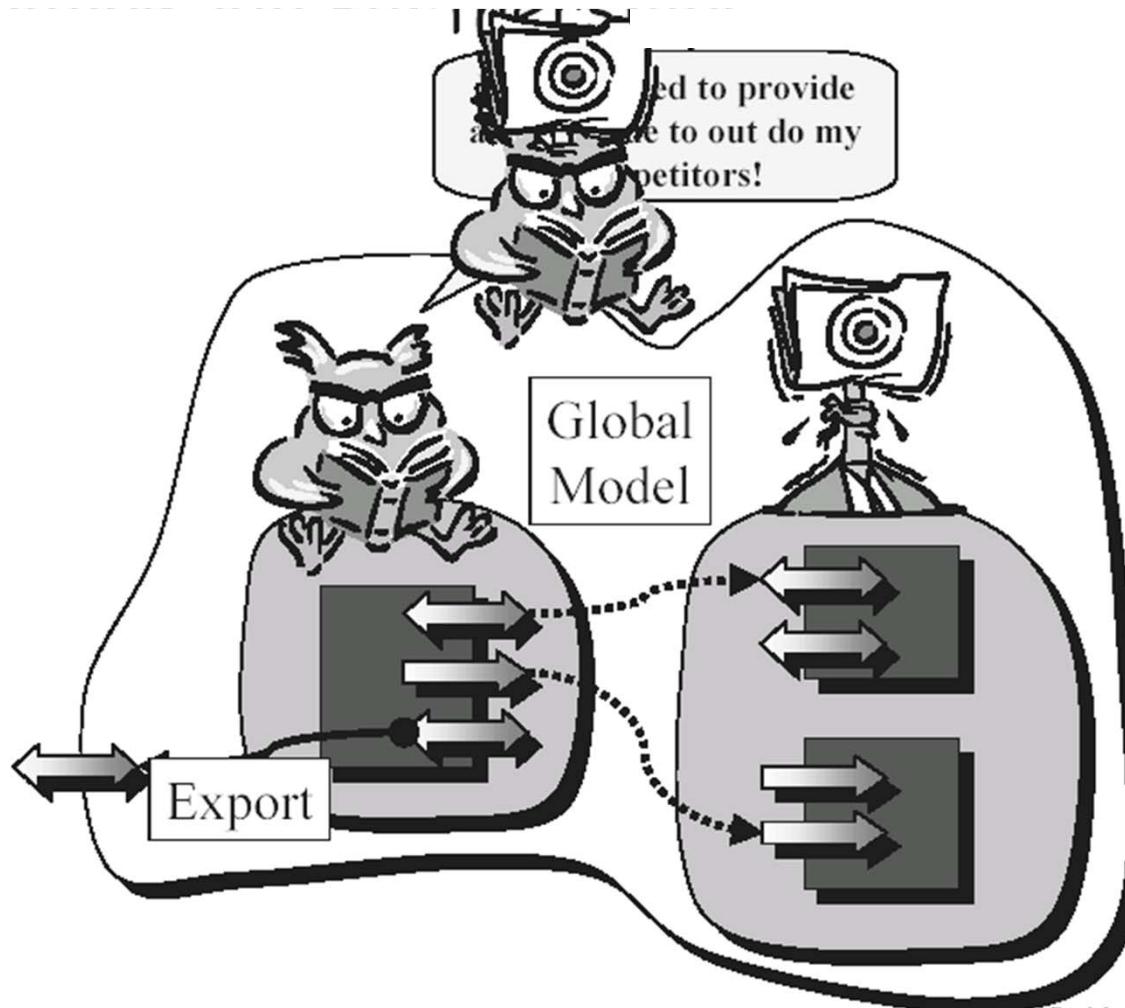
Ordnung zählt – d.h. der Ablauf ist wichtig: Flow Model



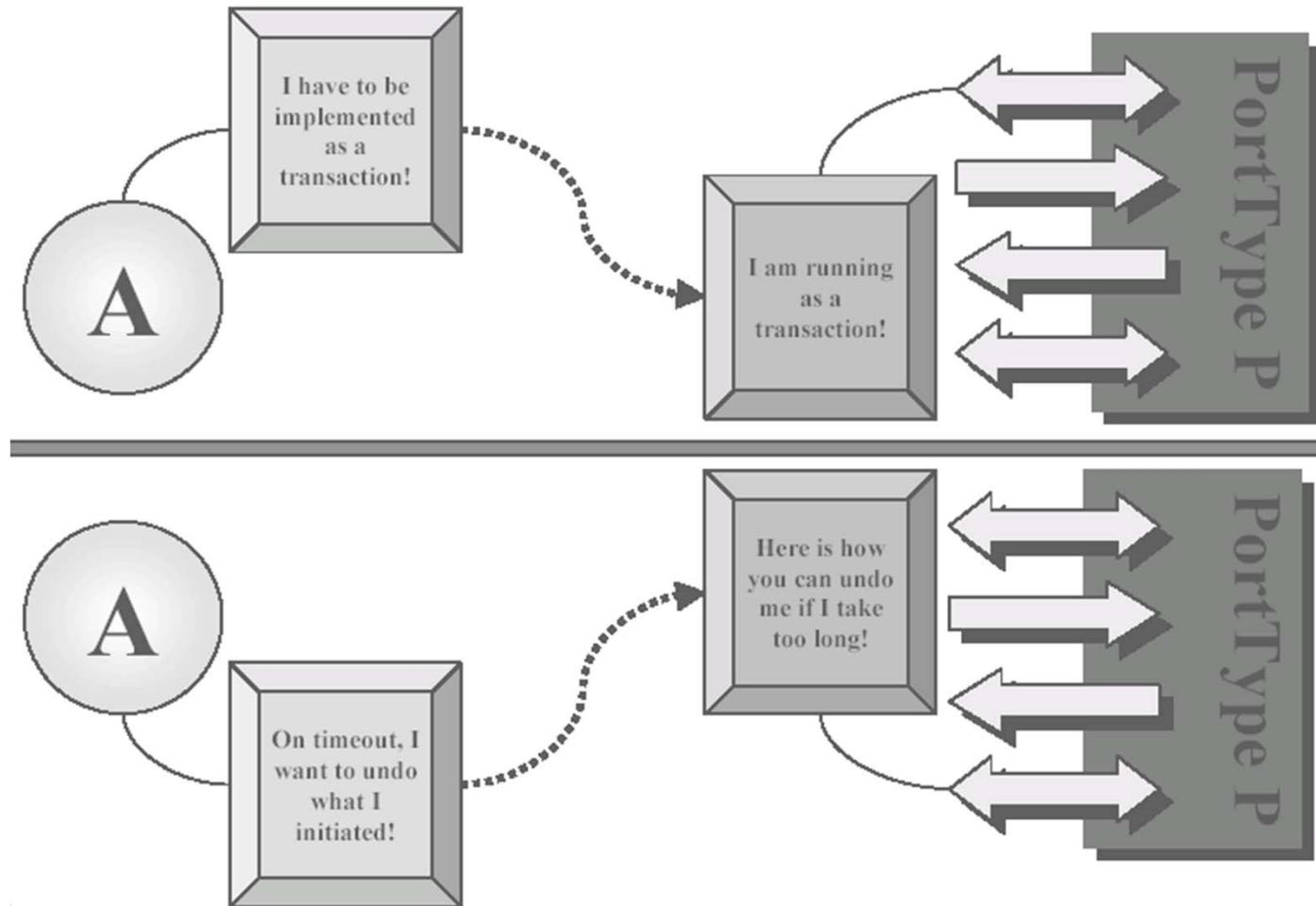
Flexibilität: dynamischer Austausch von Service-Aufrufen



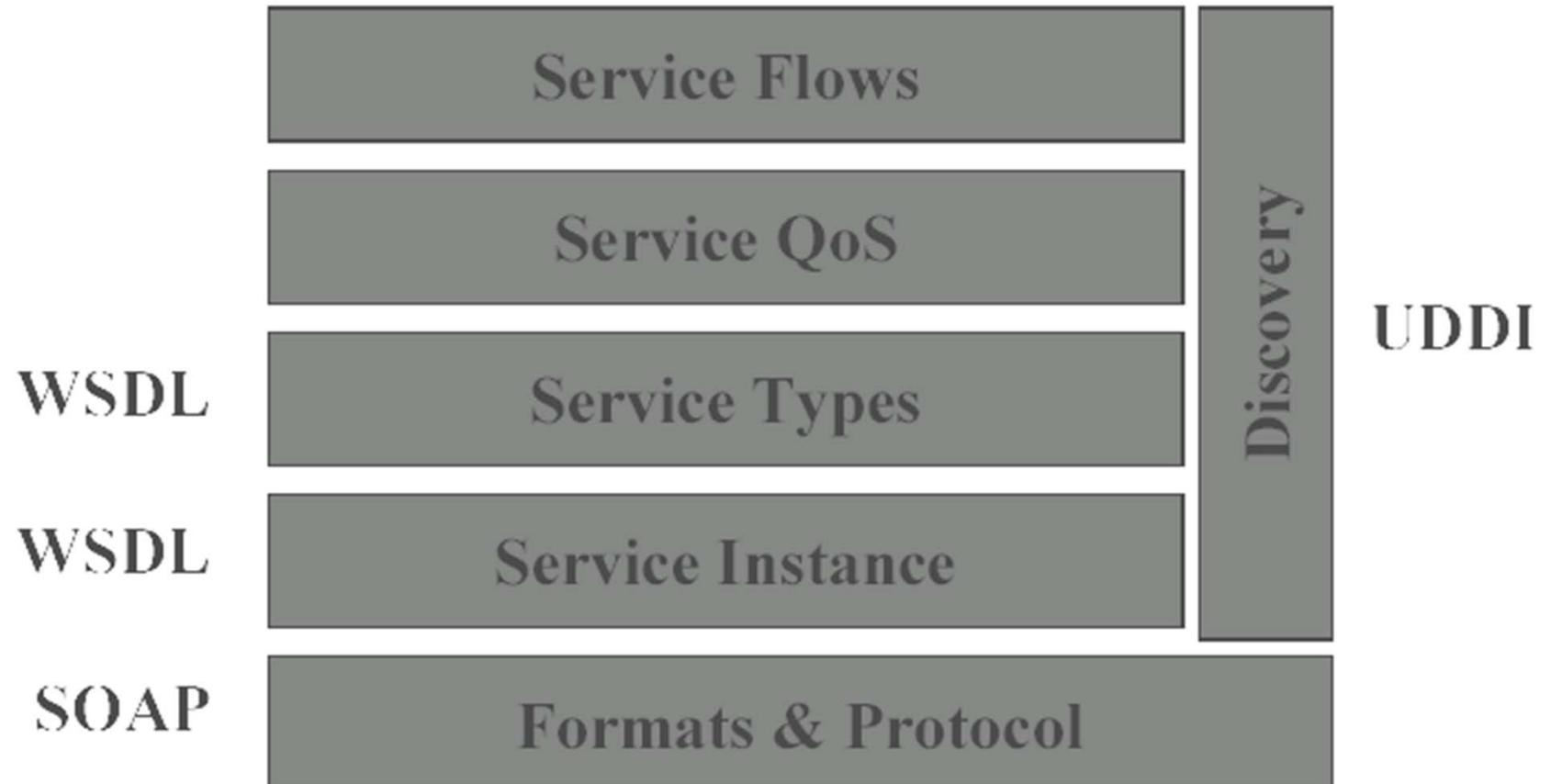
Zusatznutzen durch Partnerschaft!



Wichtiges Konzept: Qualität von Services

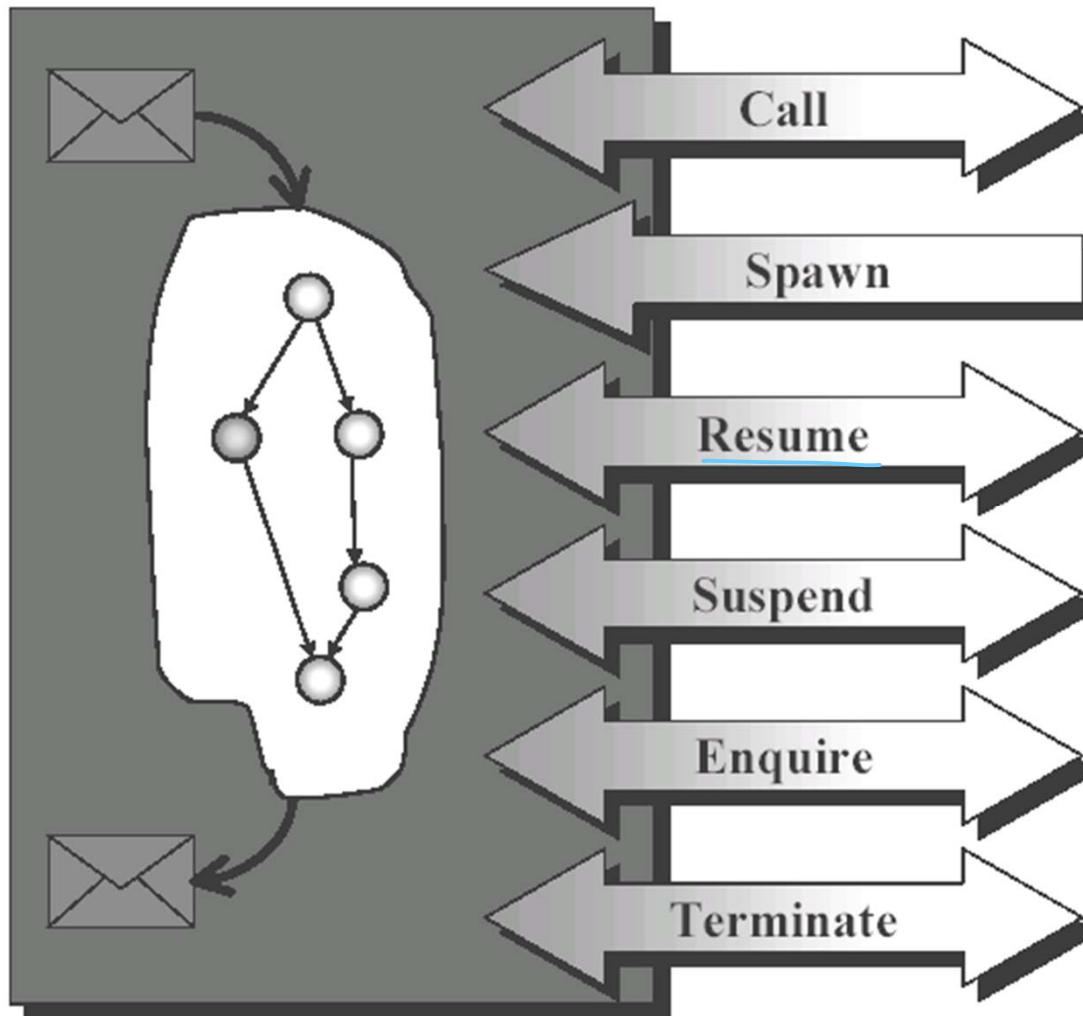


Der Web Services Stack

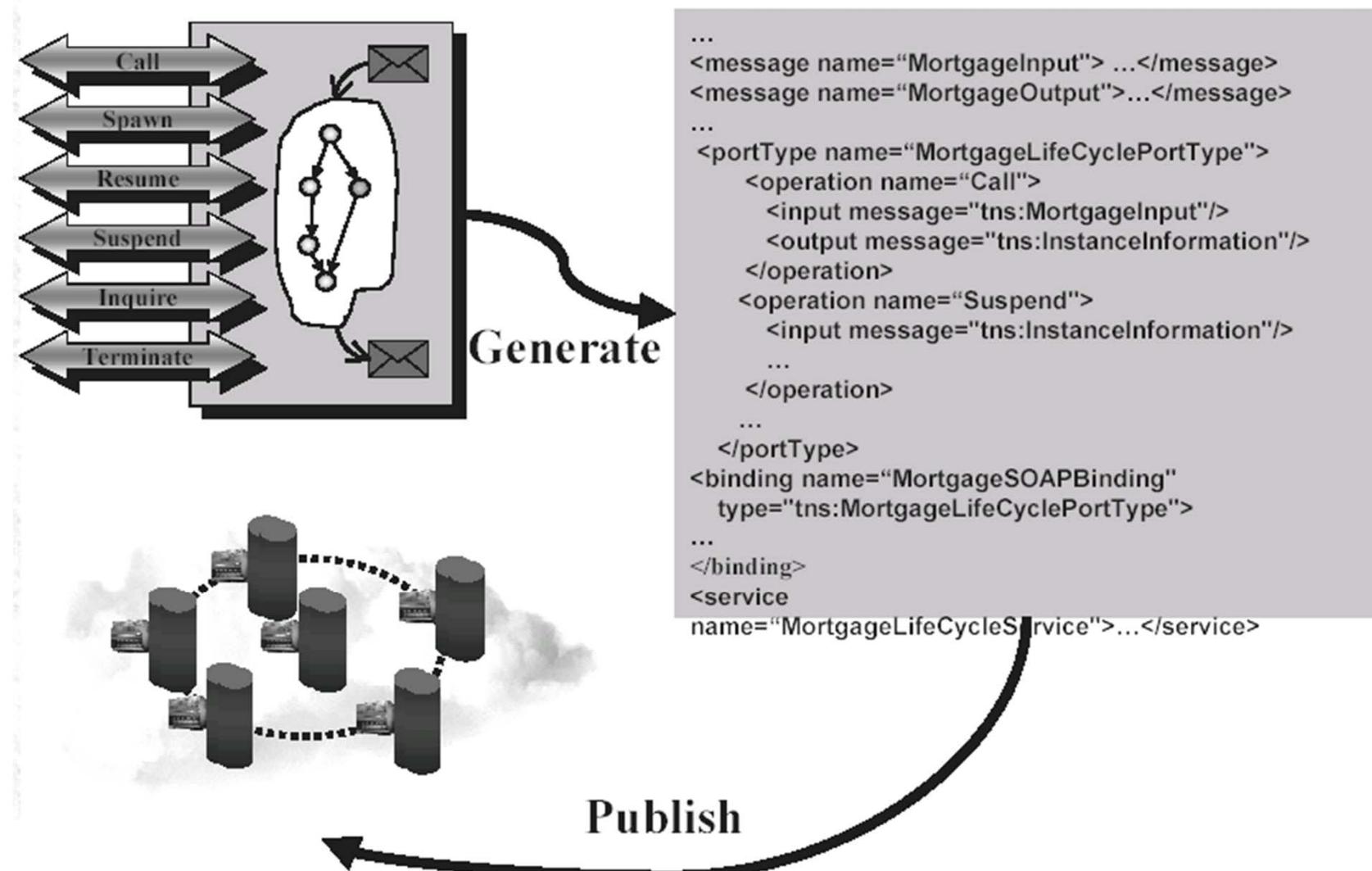


Workflows als Web Services

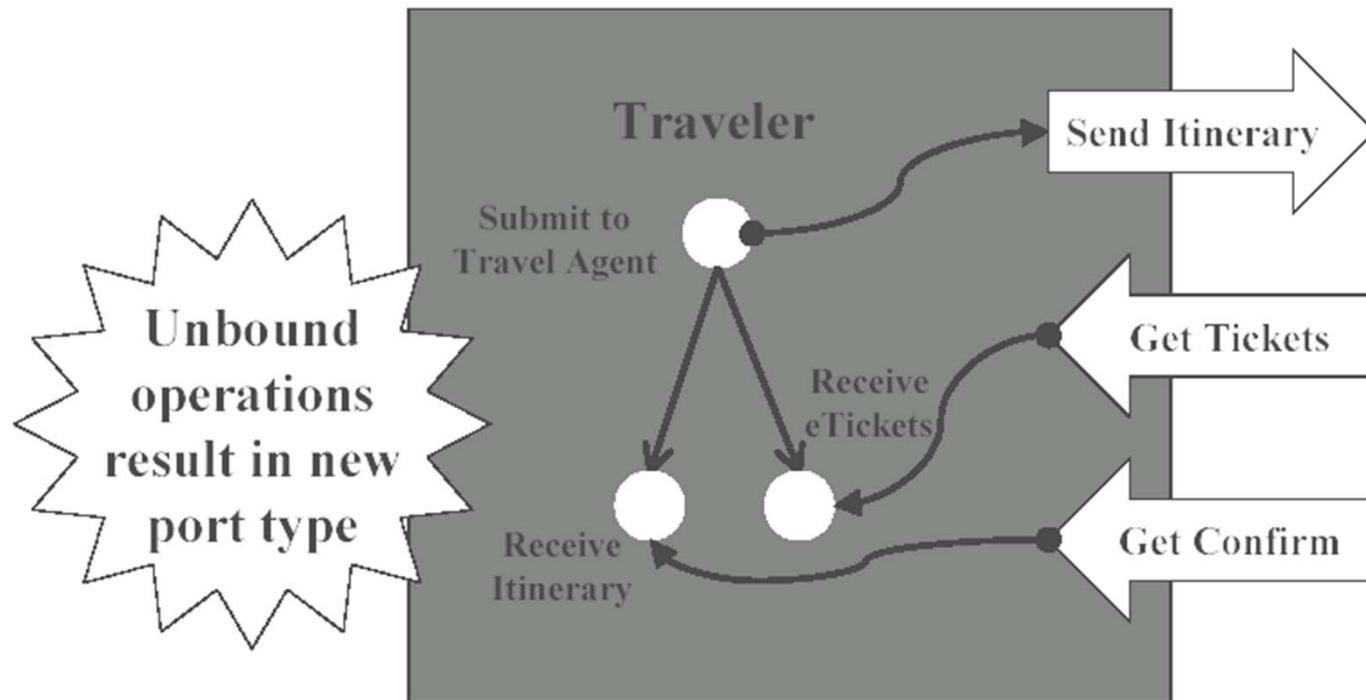
Workflows als Web Services



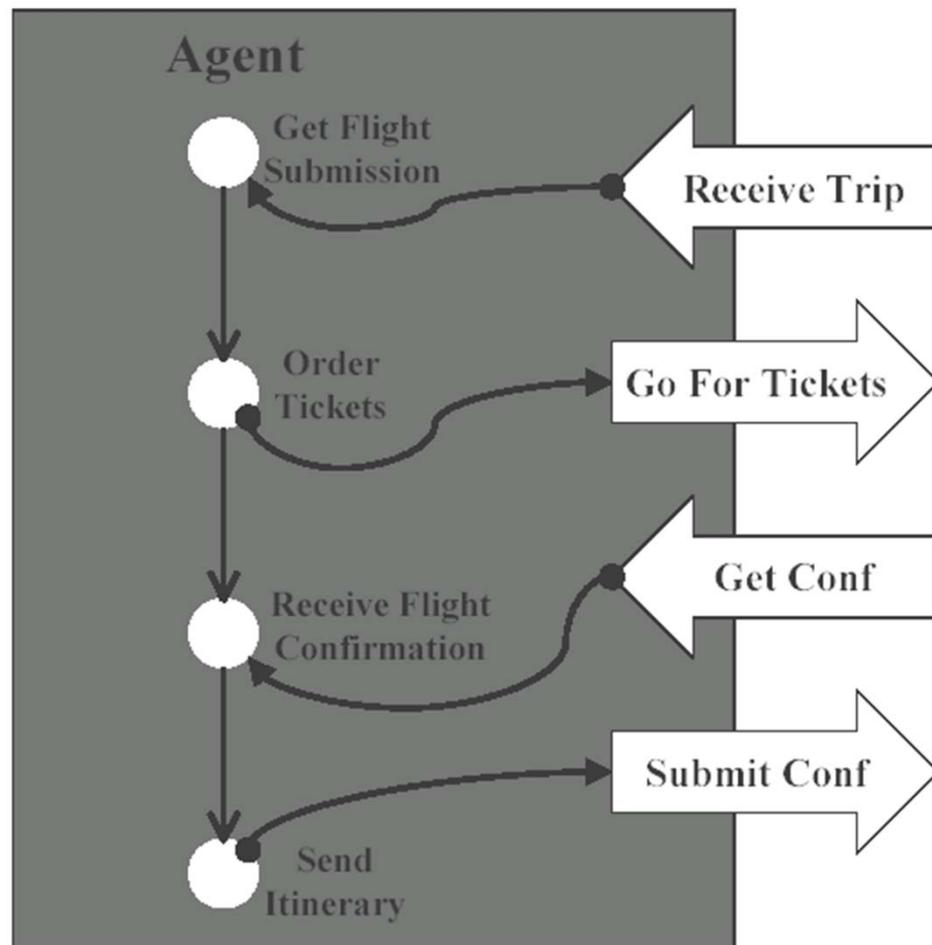
Workflow Lifecycle Services veröffentlichen



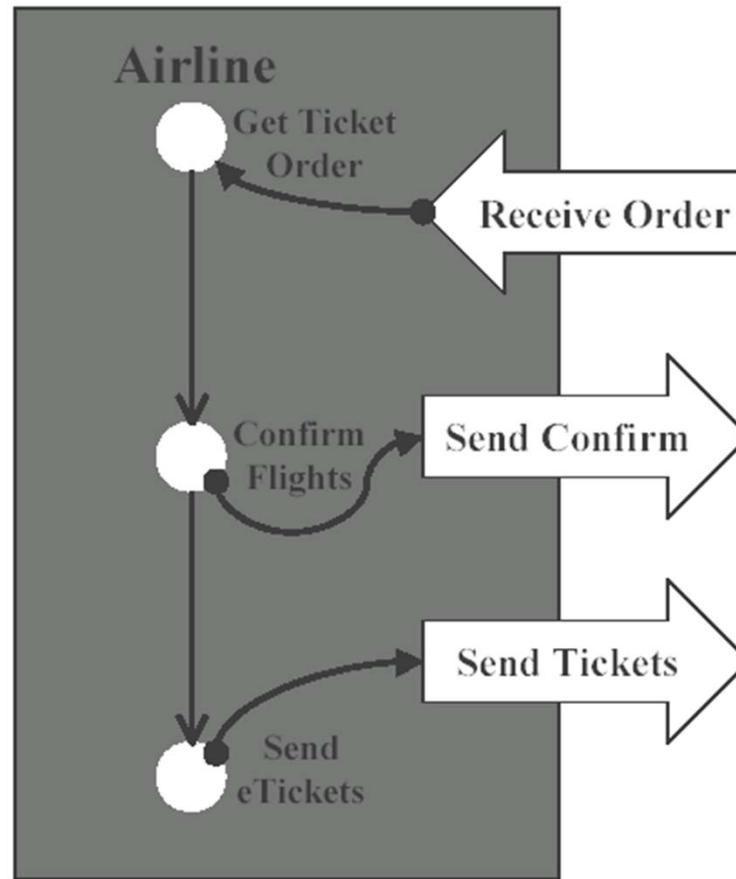
Beispiel: Reiseablauf als zusammengesetzter Service



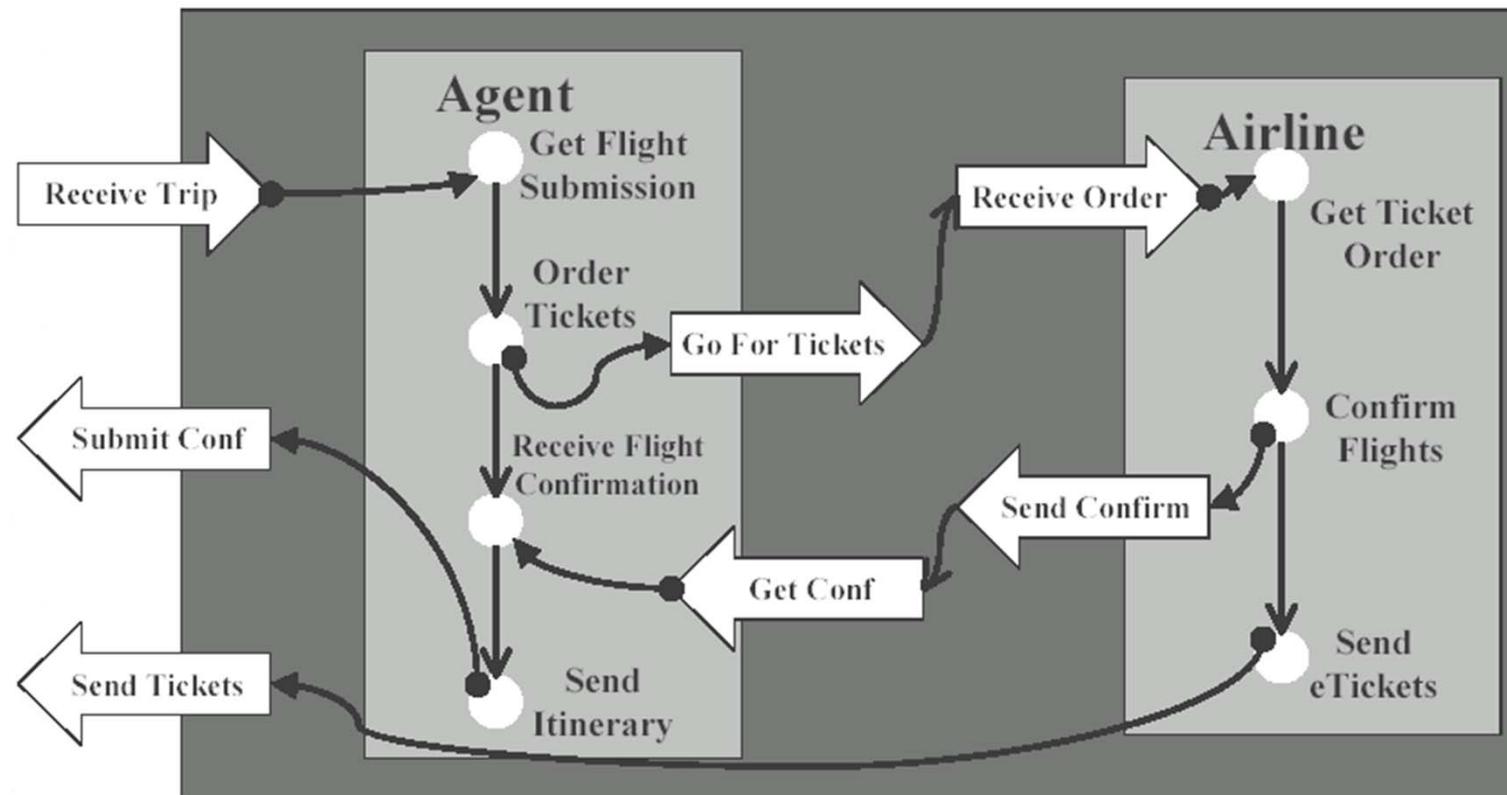
Agent als zusammengesetzter Service



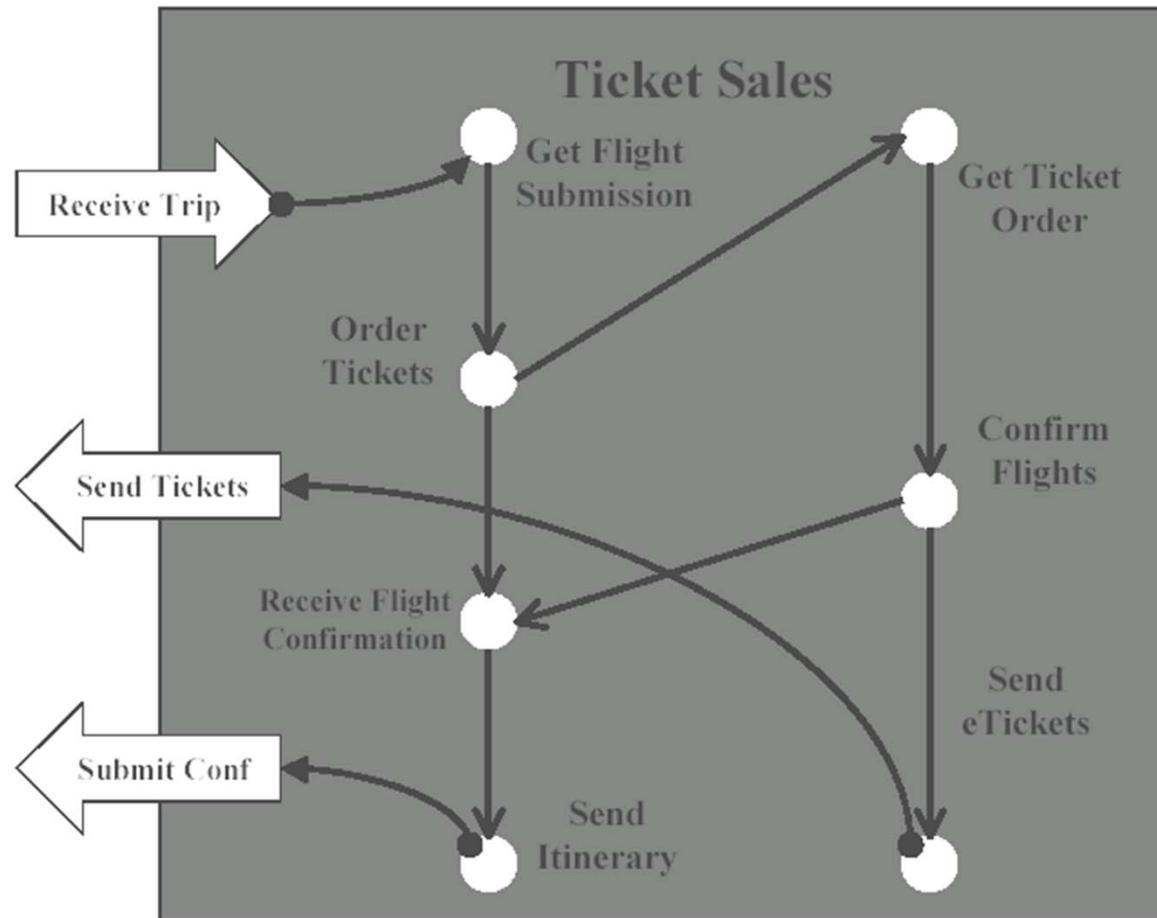
Fluggesellschaft als zusammengesetzter Service

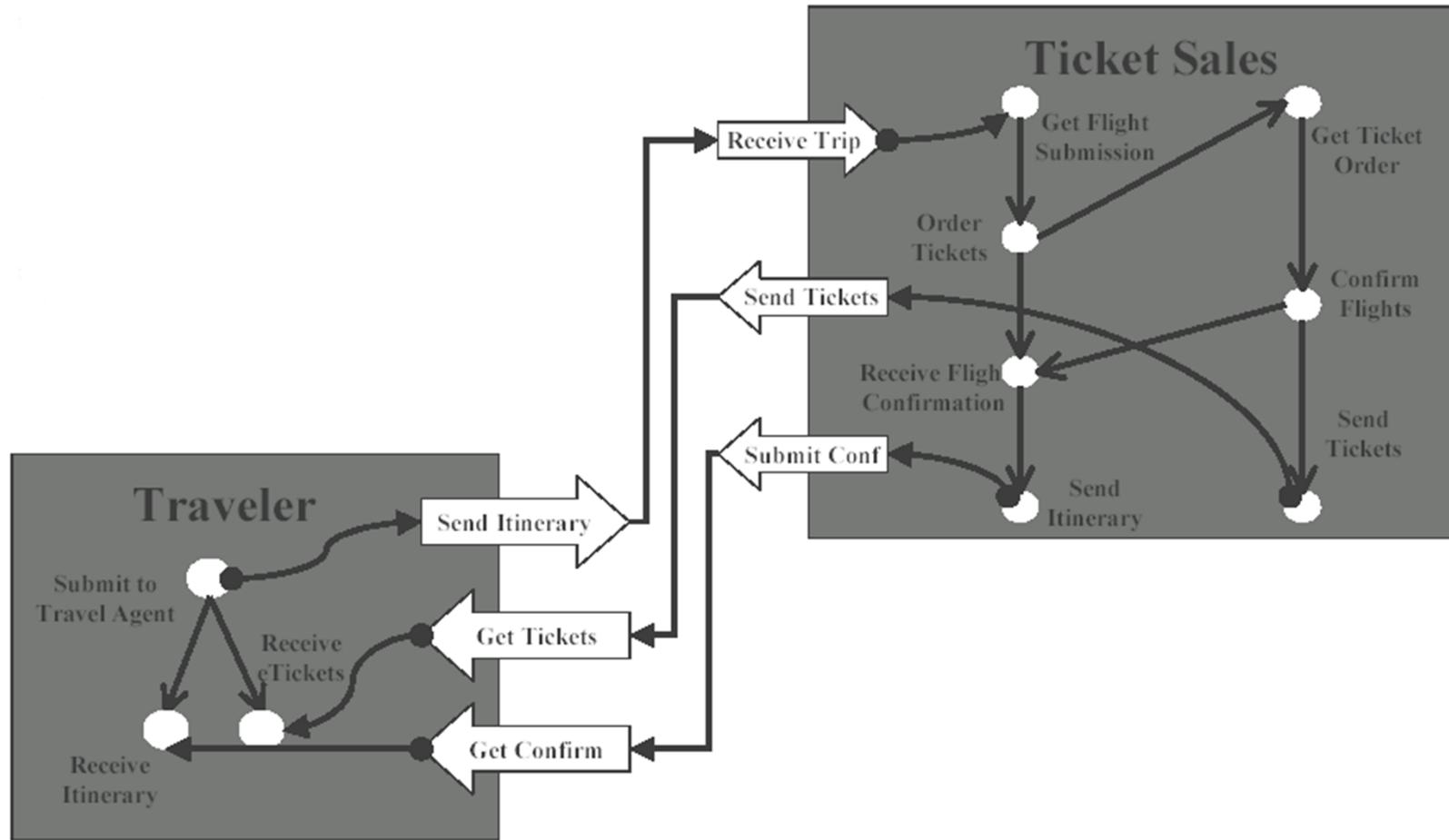


rekursive Komposition...



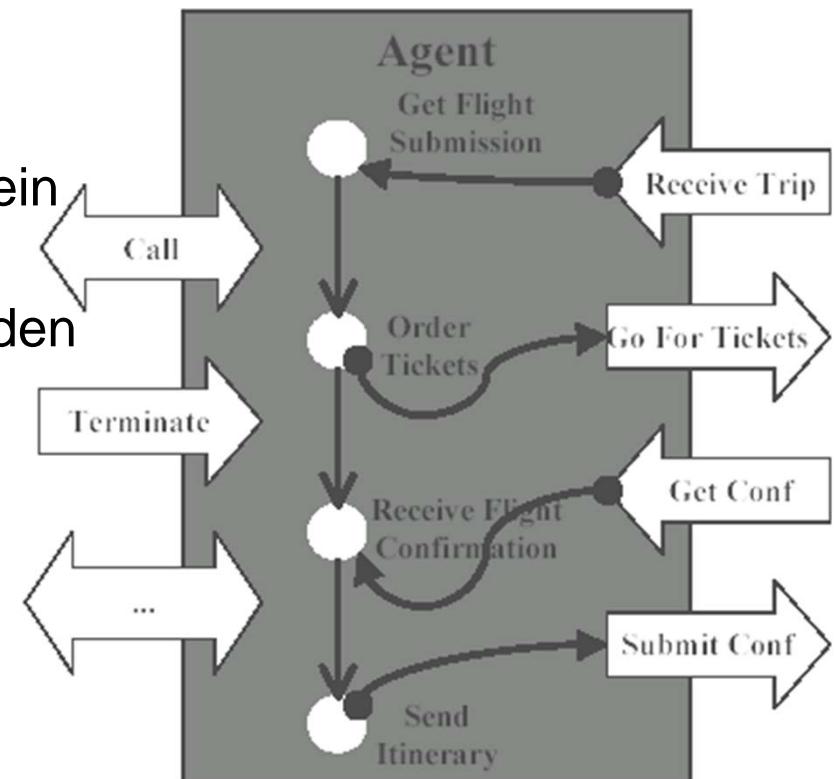
...und dies ist die Sicht von außerhalb





Vervollständigung des Bildes

- ◆ Ein Web Service Ablauf sollte zwei Arten von Operationen anbieten
 - die exportierten offenen („dangling“) Operationen, die ein Binding erfordern
 - die Lifecycle-Operationen für den Ablauf als Ganzes



Zusammenfassung

- ◆ Workflows sind festgelegte Geschäftsprozesse
- ◆ Workflows steuern die Produktion von großen Unternehmen
- ◆ Workflows über Unternehmensgrenzen gewinnen an Bedeutung
- ◆ Workflow Technologie wird zum Schlüsselkonzept von SOA (service-oriented architectures)

Exemplarische Fragen – Kapitel 6 (1)

- ◆ Was versteht man unter Web Services?
- ◆ Was sind die Grundkomponenten zur Unterstützung von Web Services?
- ◆ Wie können Abläufe mit Web Services realisiert werden?
- ◆ Was ist eine Web Service Choreographie?
- ◆ Was versteht man unter Web Service Orchestrierung?

Exemplarische Fragen – Kapitel 6 (2)

- ◆ Nennen Sie Standardisierungsvorschläge für Choreographie und für Orchestrierung?
- ◆ Was sind Eigenschaften von Ablaufunterstützung für virtuelle Unternehmen?
- ◆ Nennen Sie Erweiterungen von Workflows im Internet in Richtung ‚Semantic Web‘.