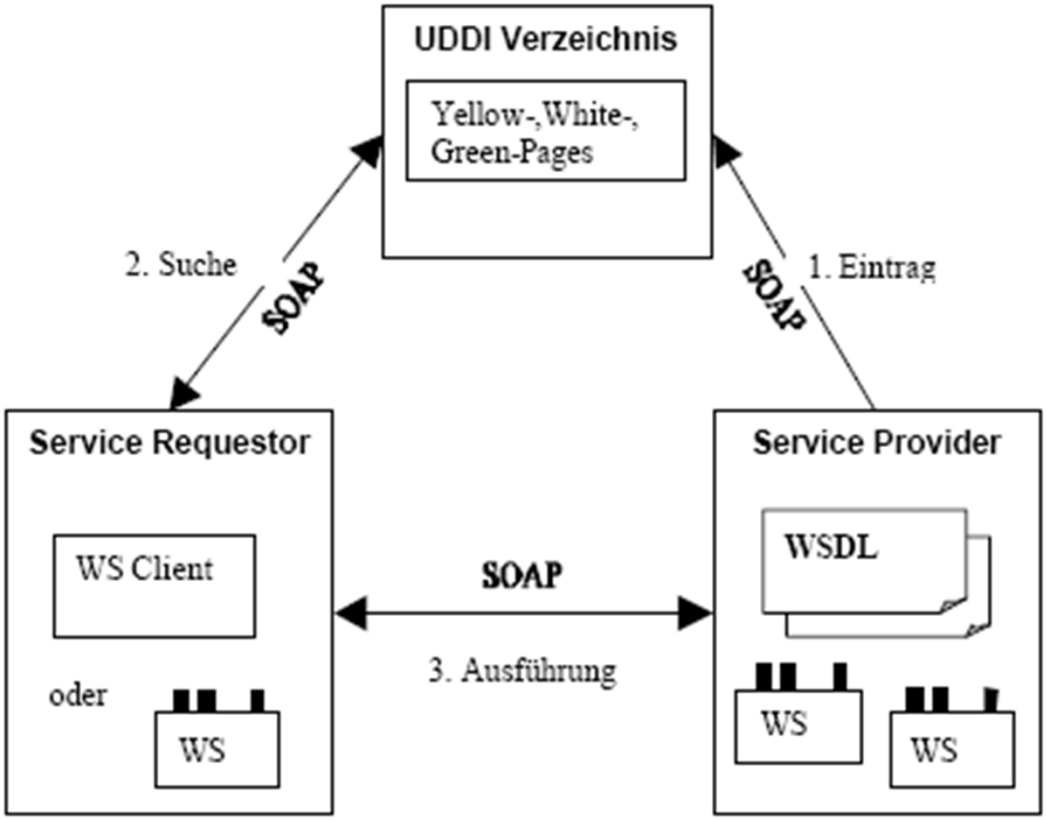
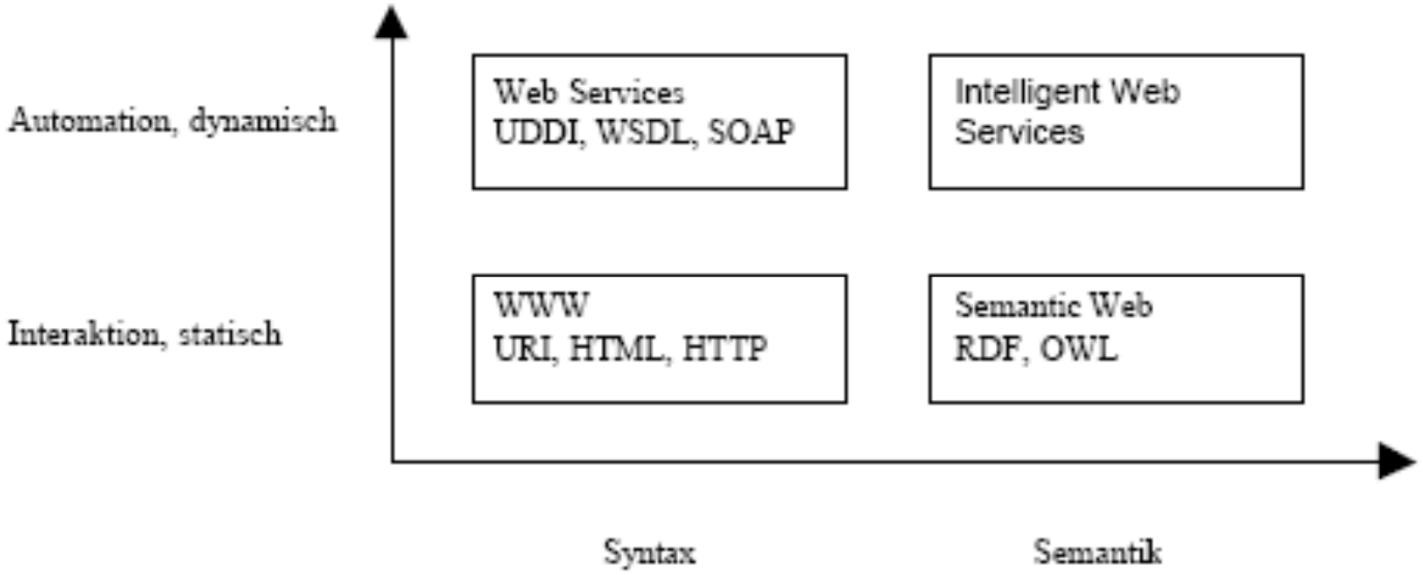
Kapitel 6:
Workflows im Internet

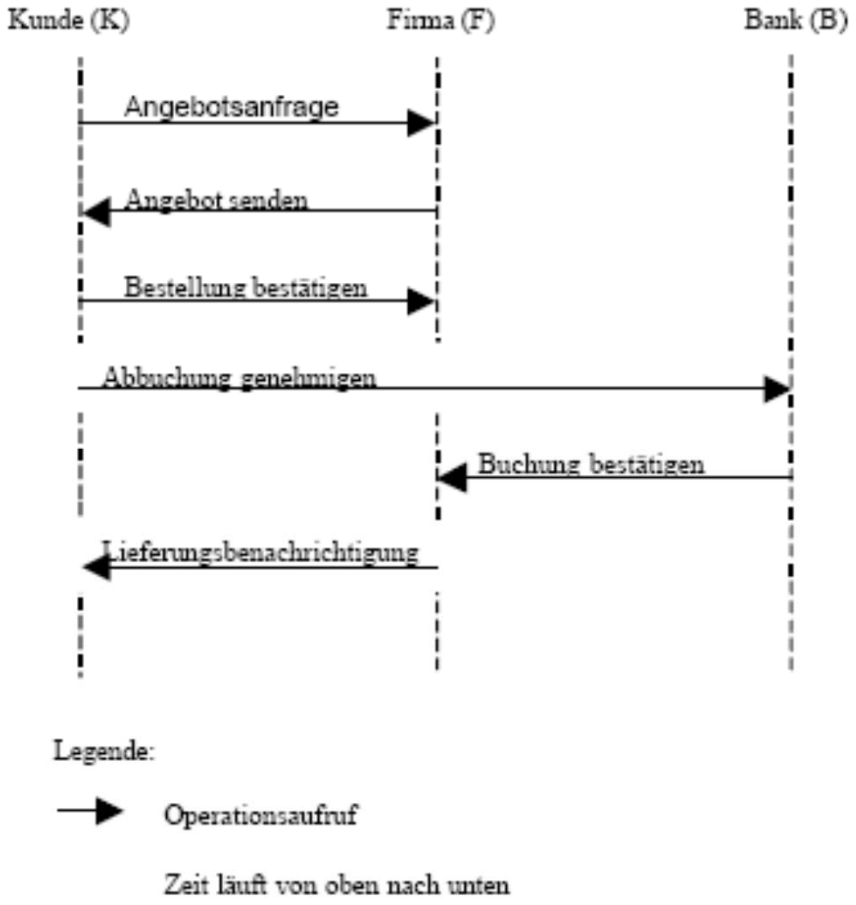
Service-orientierte Architektur (SOA) mit Web-Services



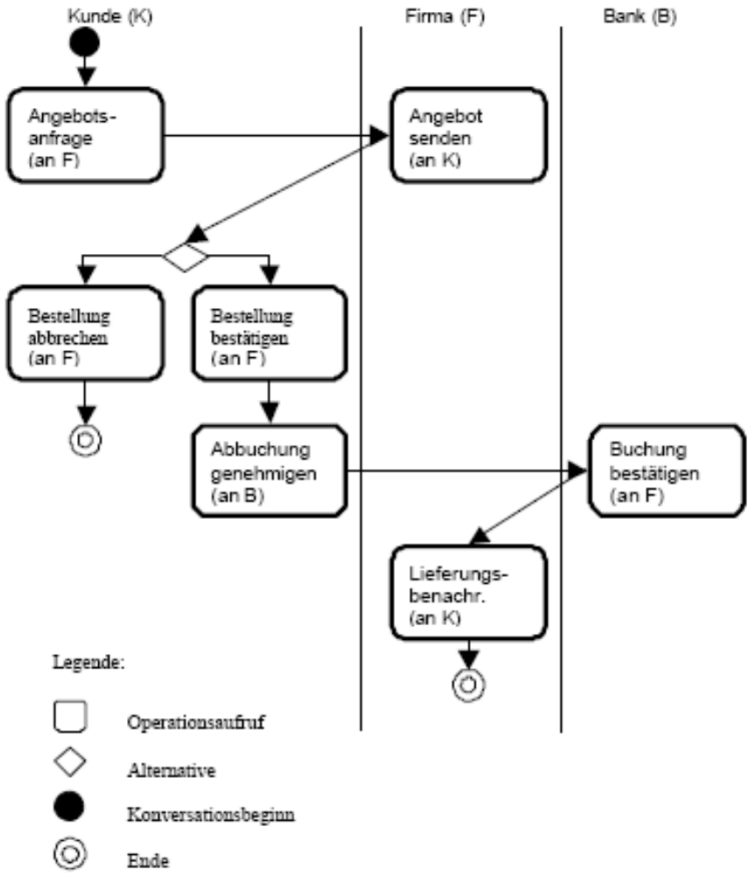
Einordnung von Semantic Web



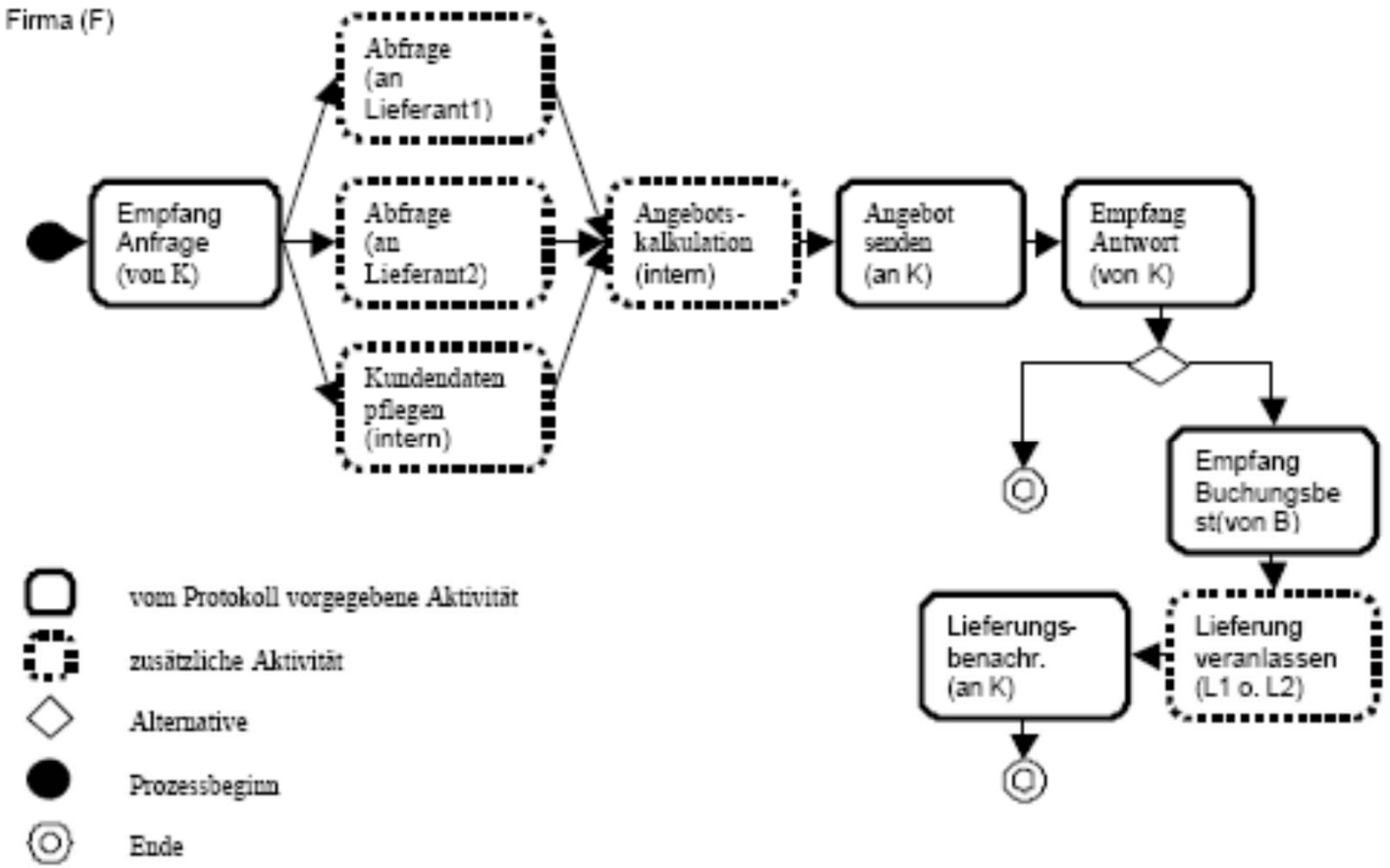
Konversation – Nachrichtenabfolge mit drei Partnern



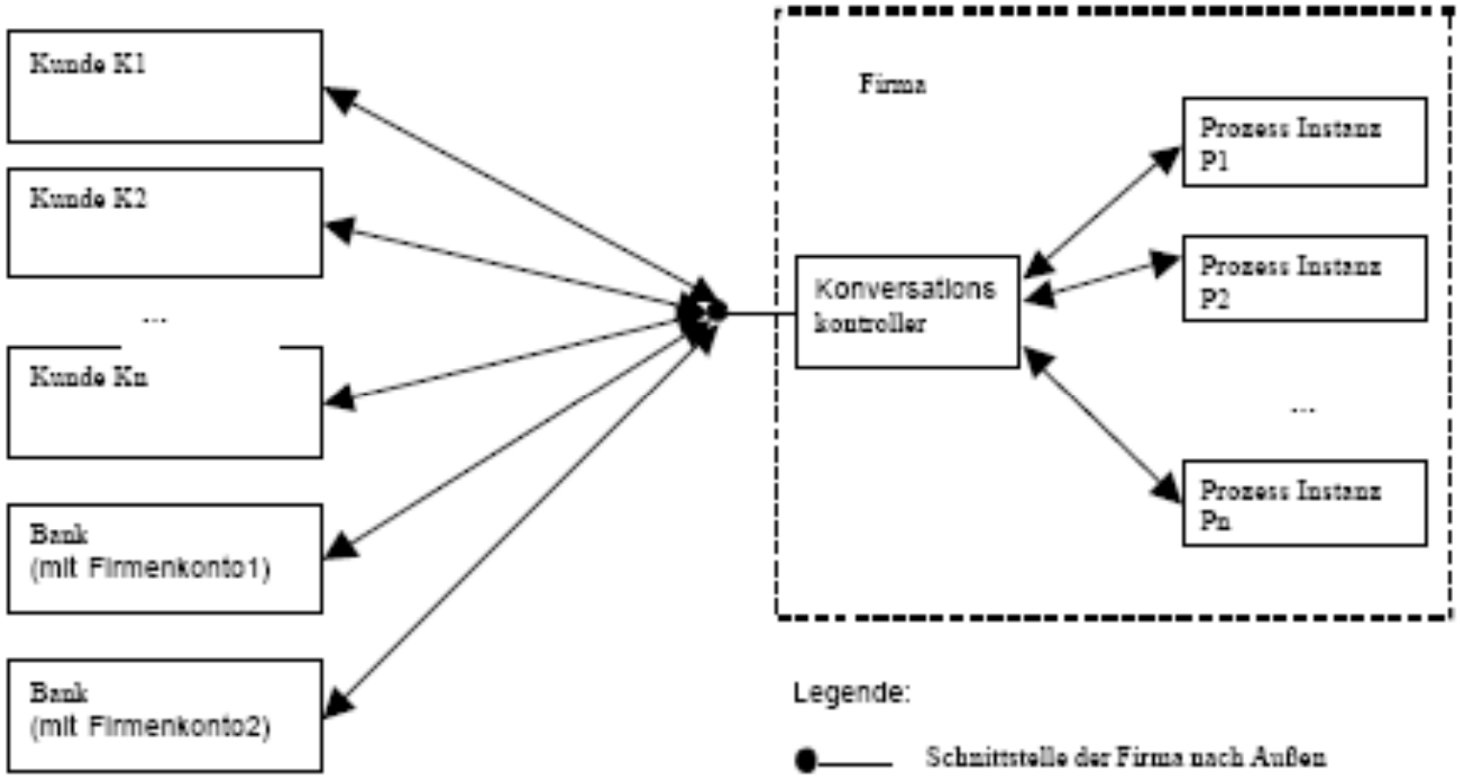
Choreographie - Koordinationsprotokoll



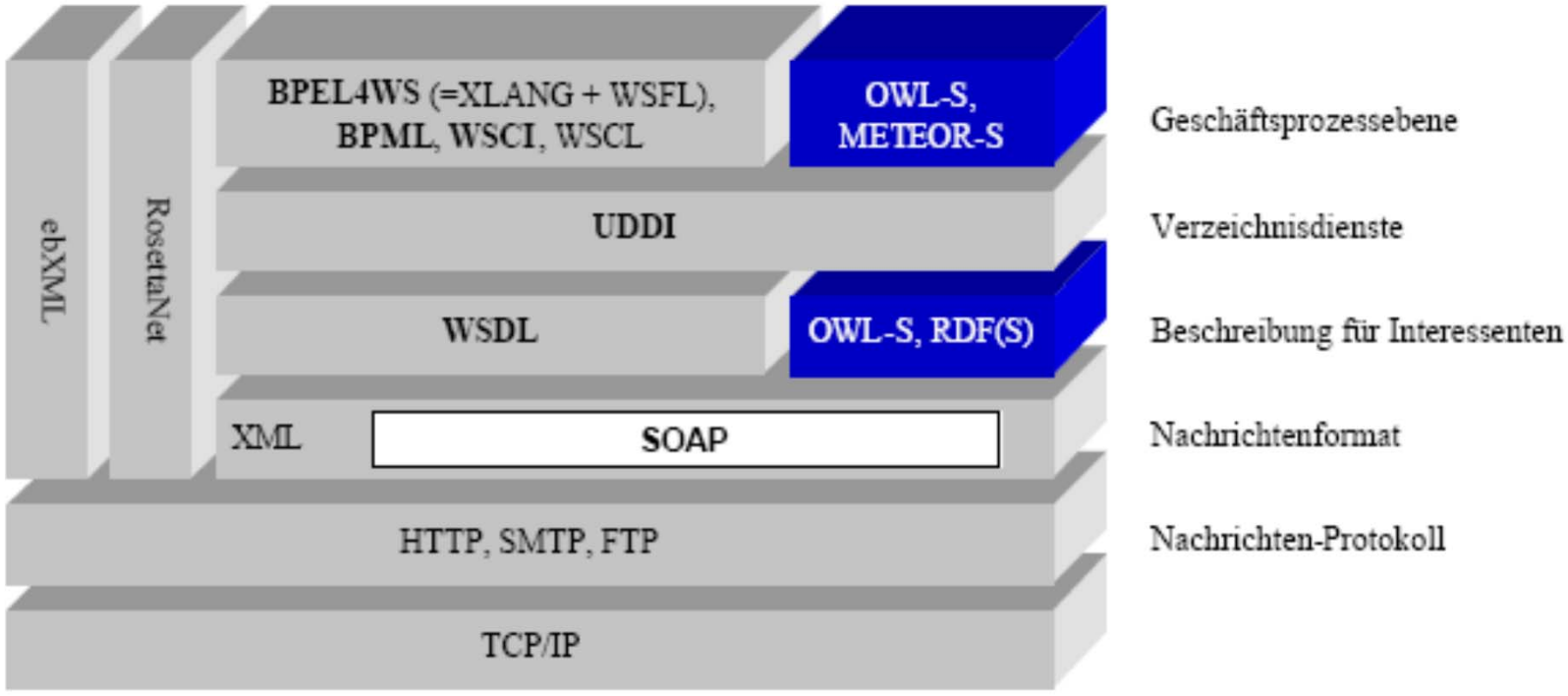
Orchestrierung von Web-Services



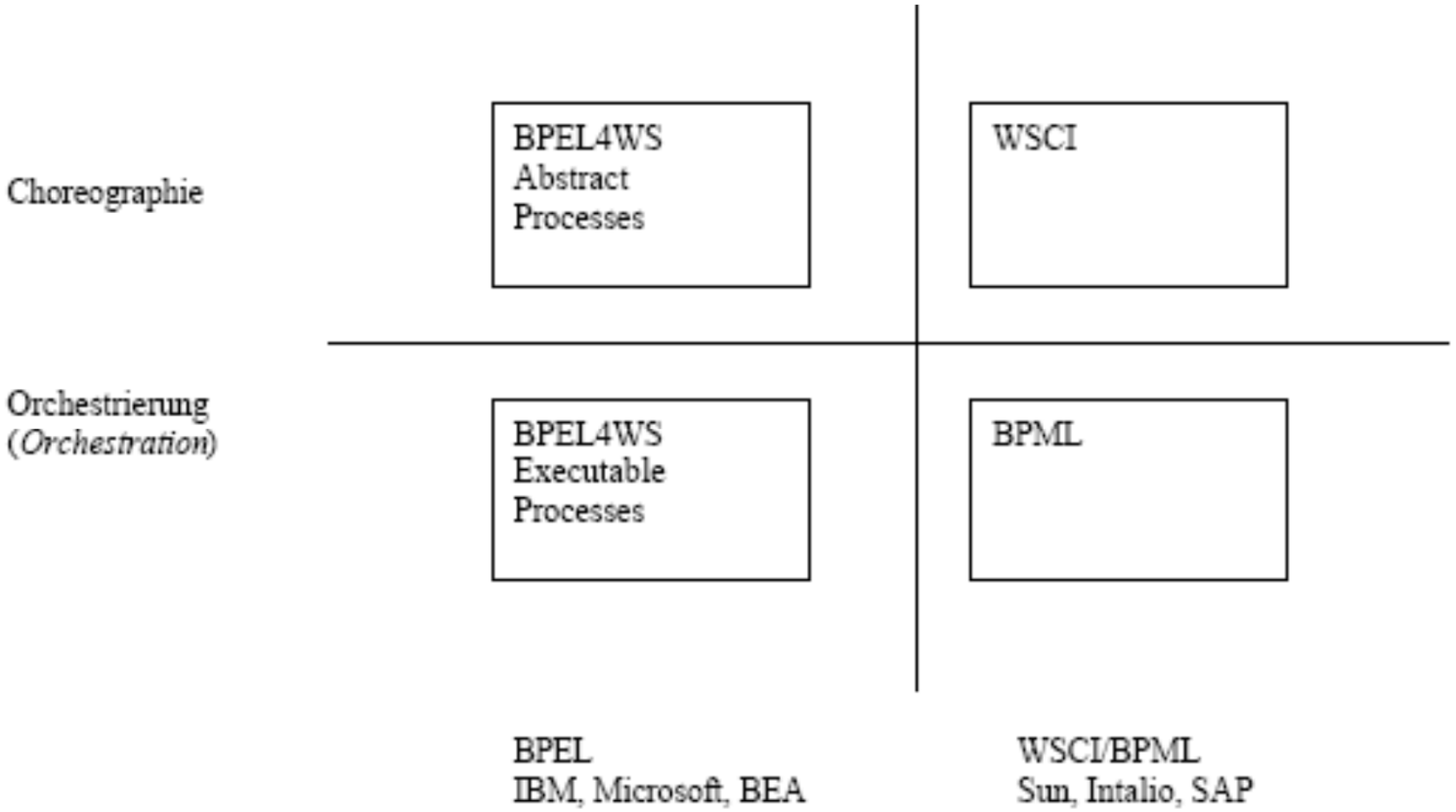
Konversationsunterstützung



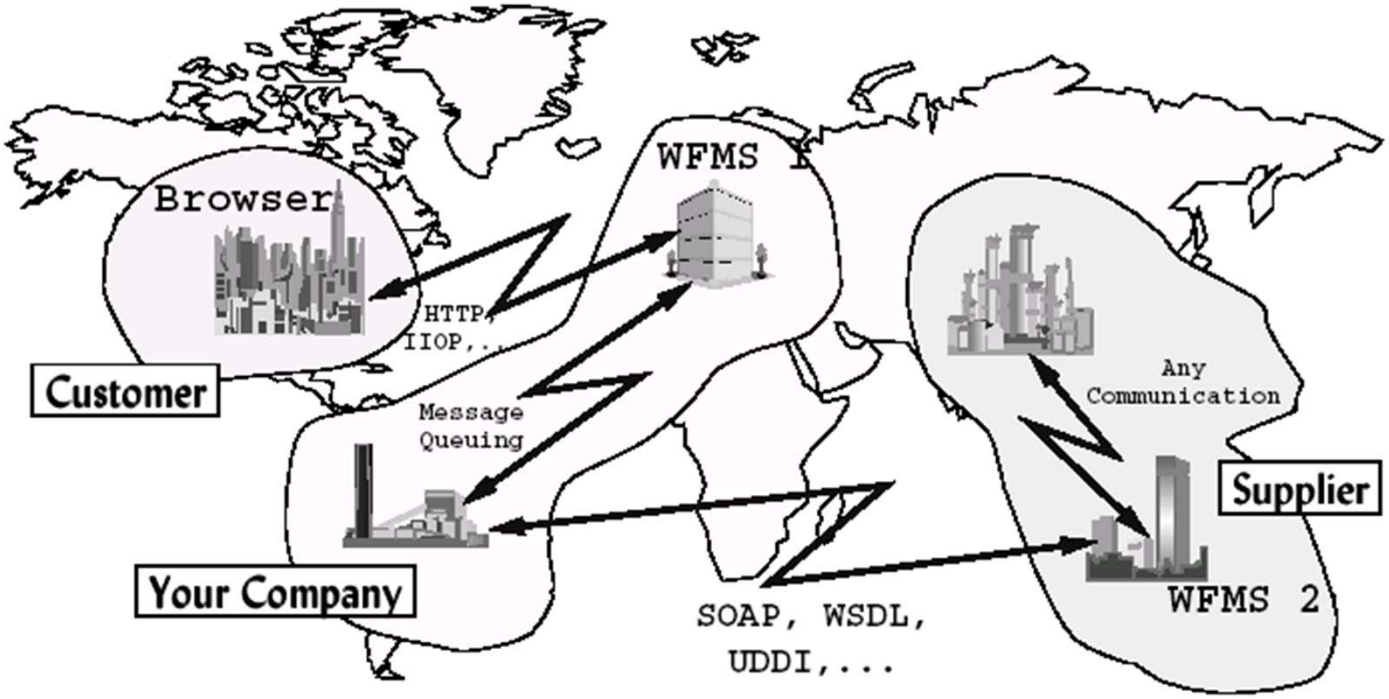
Web Service Stack - Semantikunterstützung



Standardisierungssprachen



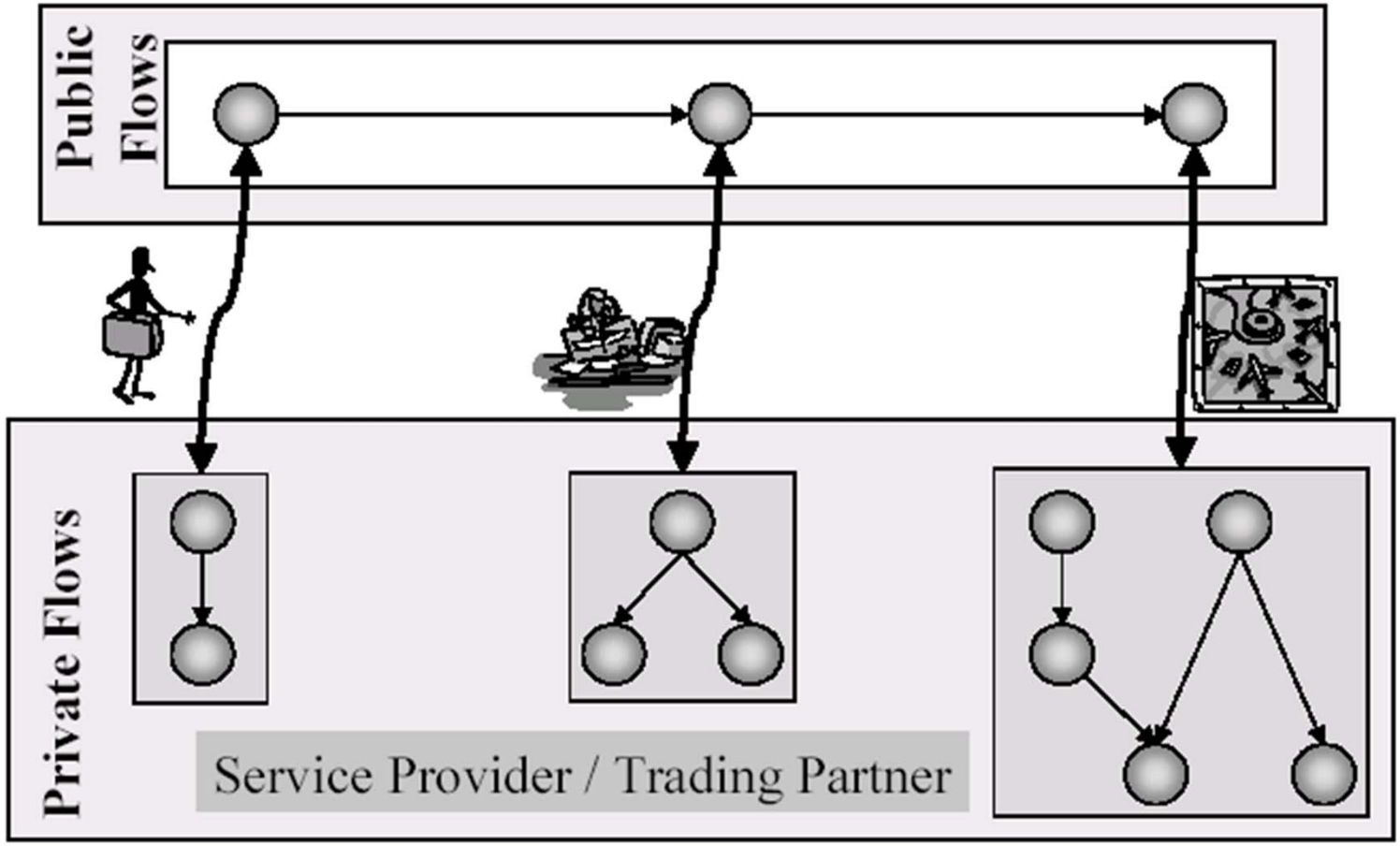
Virtual Enterprise: Scenario



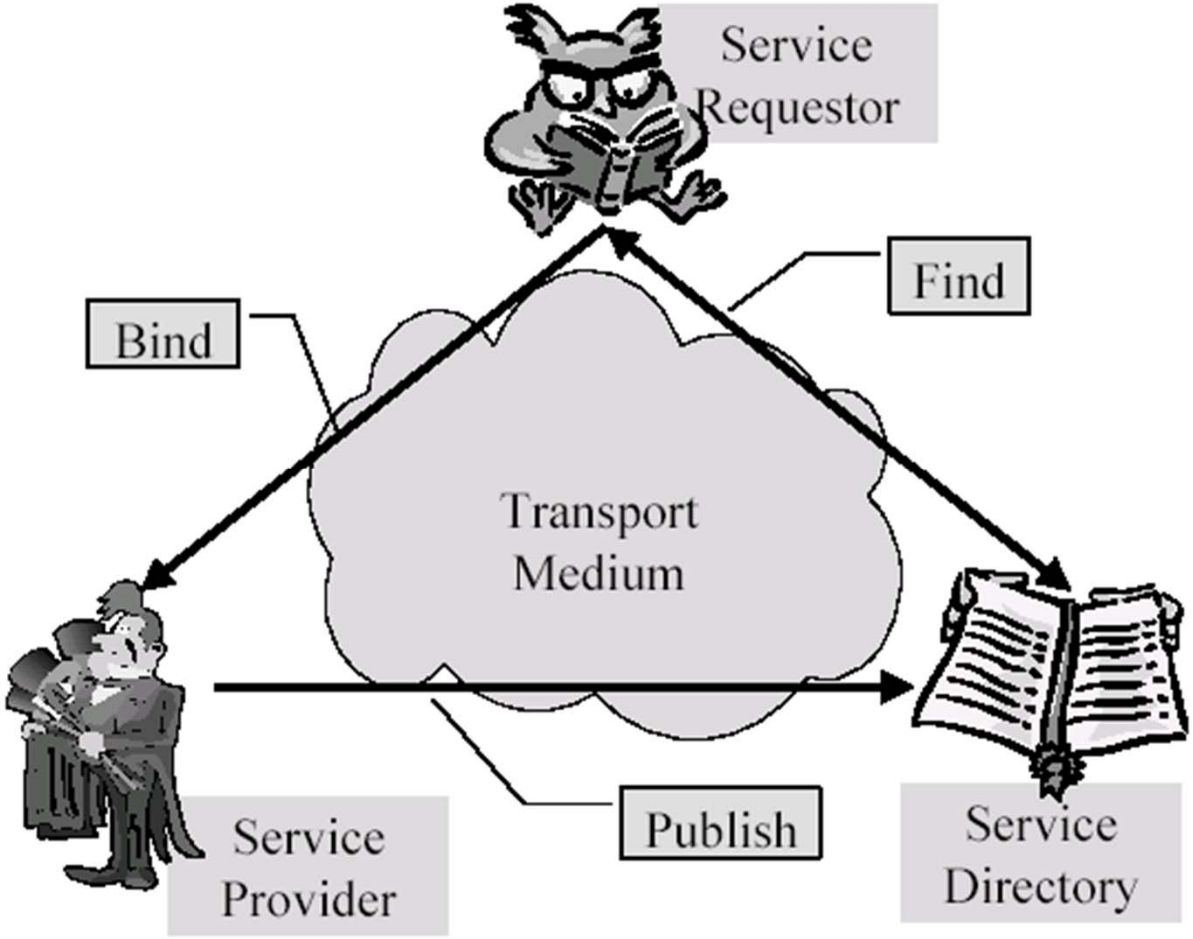
Workflows zwischen Partnern

- 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
  - Spezifiziere 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 Unternehmenskompentenzen 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!

Wie nutzt man Dienste (Web Services)?

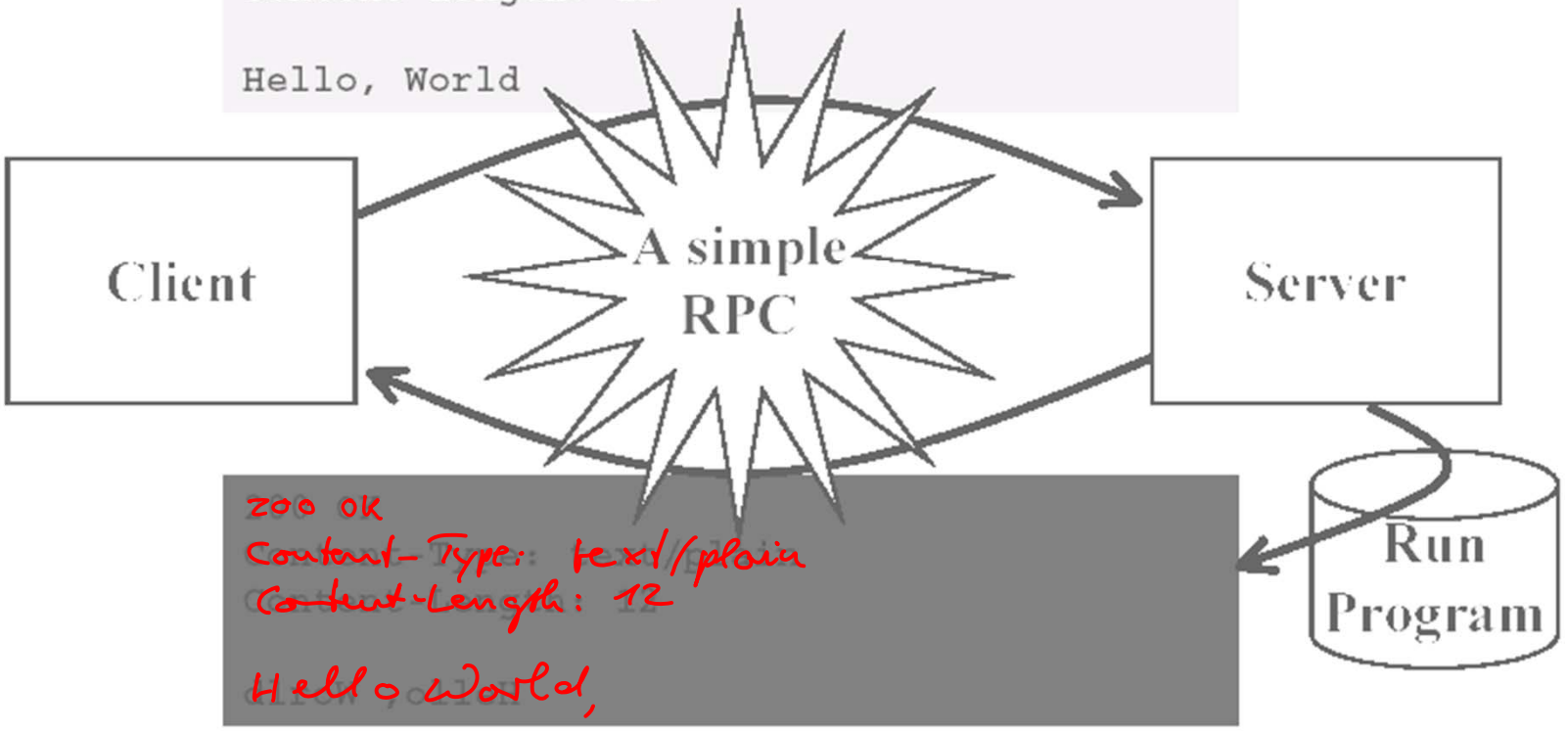
- Firewalls stellen Hindernisse dar für den unbeschränkten (ubiquitous!) Zugang zu Diensten
- Die Vielfalt und Heterogenität von Programmiersprachen, Programmierparadigmen, Server-Umgebungen, etc. … machen die Orchestrierung zu einem Albtraum
- …

Wir brauchen einen global verfügbaren Aufrufmechanismus!: SOAP

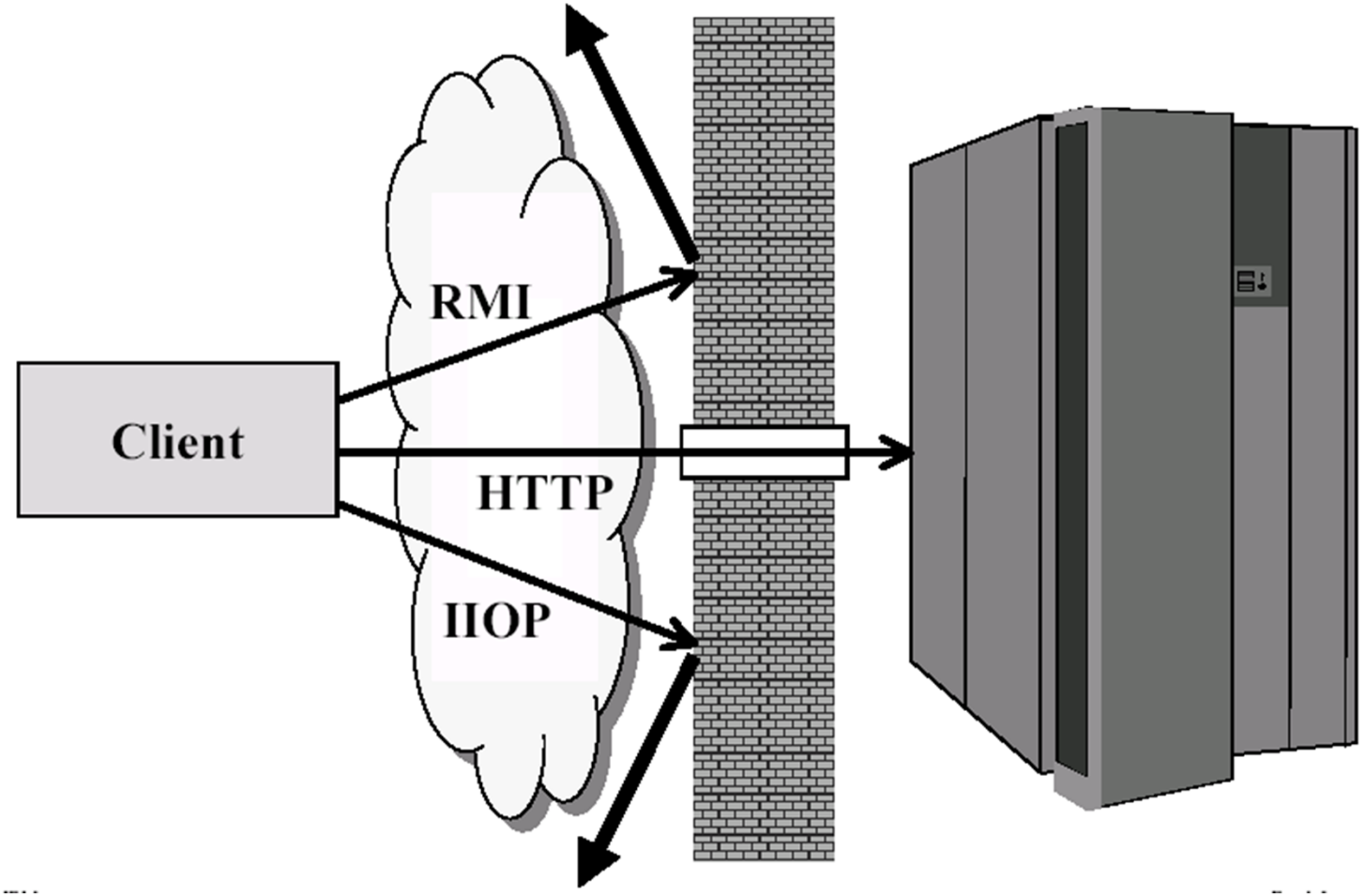
SOAP

HTTP POST: Aufruf-Code

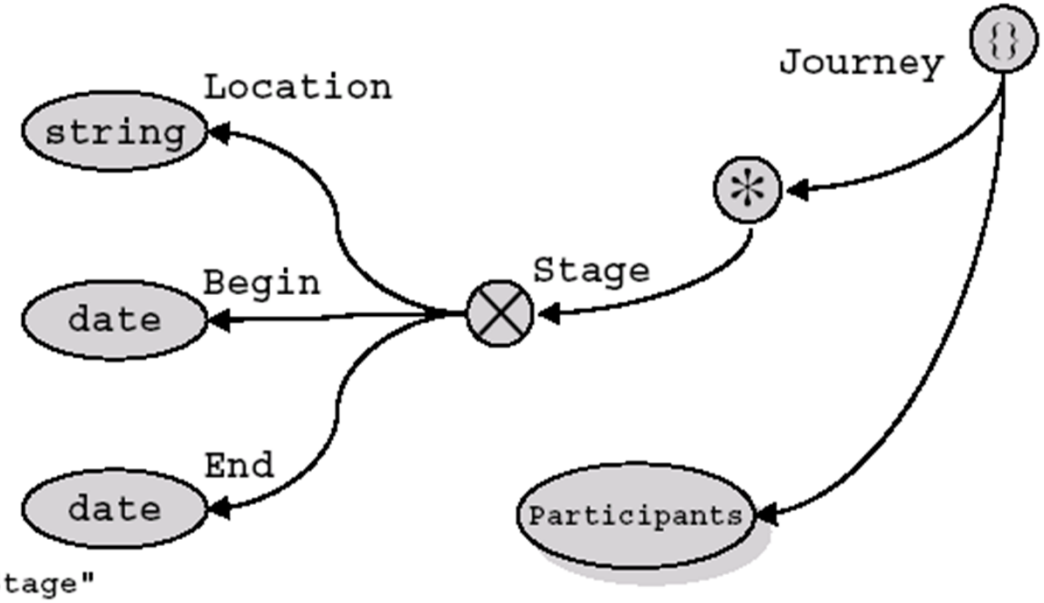
```
POST /myFunctions/reverse HTTP/1.1
Host: www.Frank-Leymann.com
Content-Type: text/plain
Content-Length: 12

Hello, World
```



Die Firewall „durchtunneln“

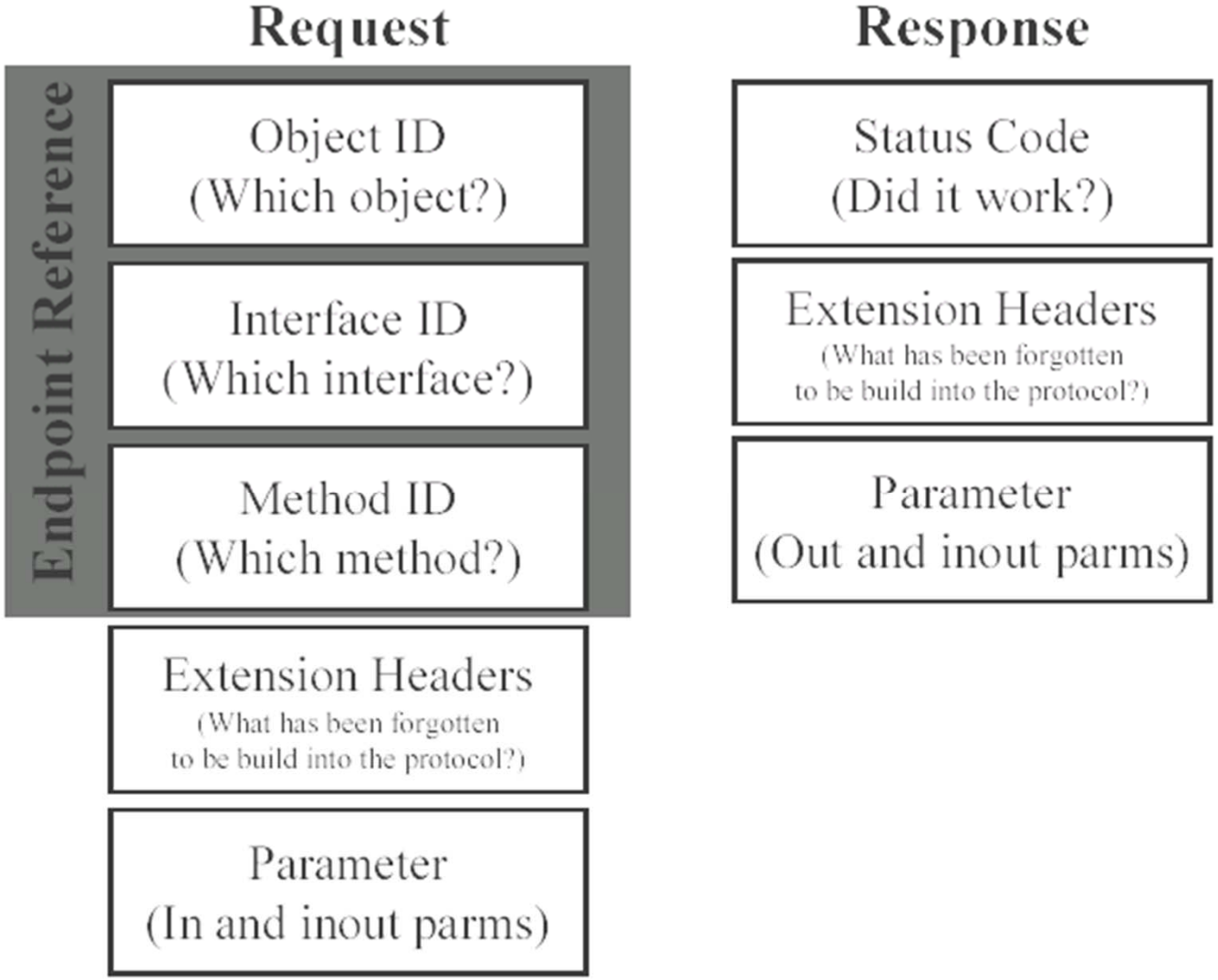


Ein XSD-Beispiel

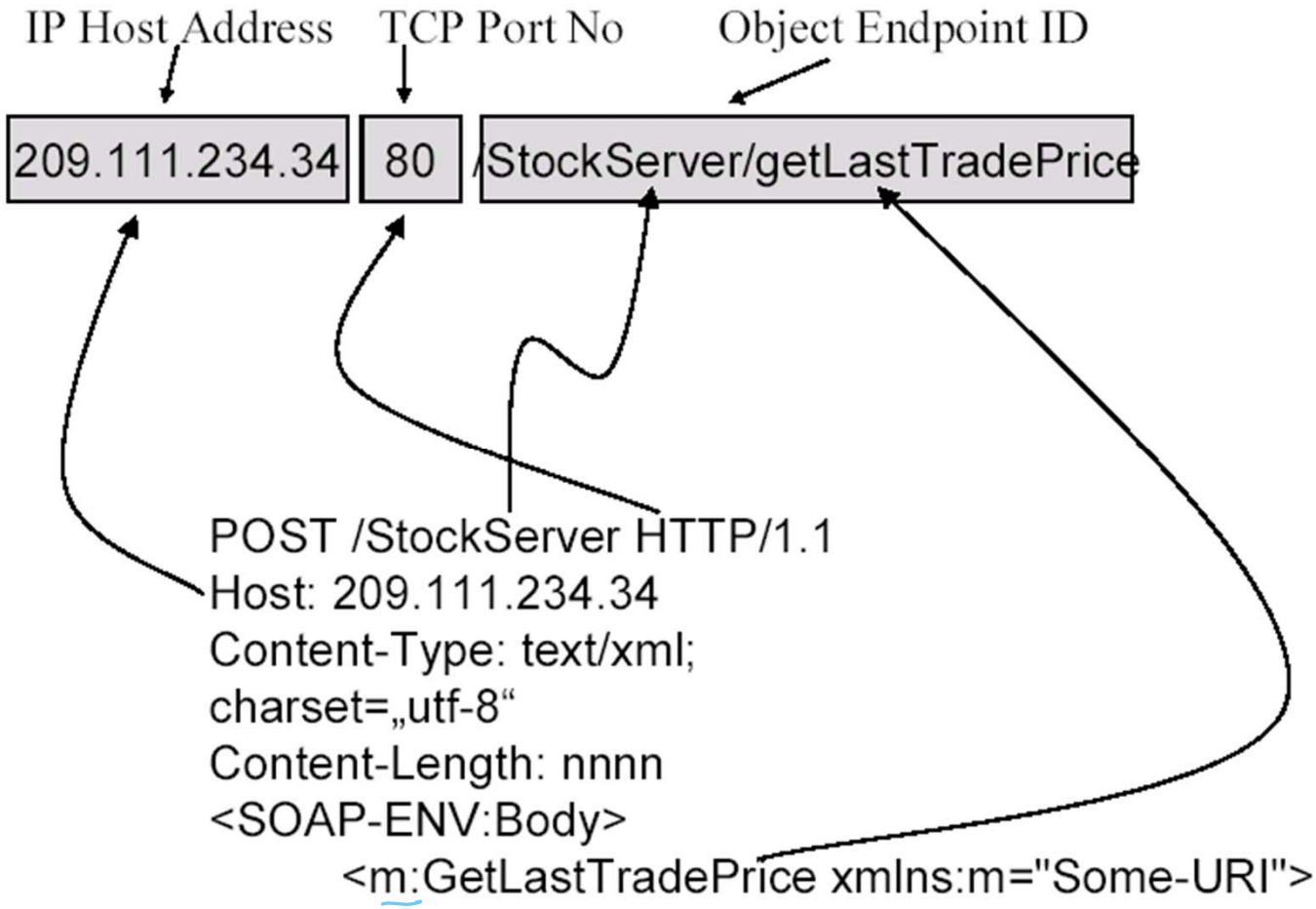



```
<element name="Journey">
        <complexType>
            <all>
                <element name="Stage"
                         minOccurs="1" maxOccurs="unbounded">
                    <complexType>
                        <sequence>
                            <element name="Location" type="string"/>
                            <element name="Begin" type="date"/>
                            <element name="End" type="date"/>
                        </sequence>
                    </complexType>
                </element>
                <element ref="participants"/>
            </all>
        </complexType>
    </element>
```

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-Inspectione Dokumente im Basisverzeichnis des Webservers der jeweiligen Unternehmung.
- Unterverzeichnisse / Hierarchien sind möglich
- Dynamische Definitiond er 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
  - Links (z.B. zu Inspection-Unterverzeichnissen oder UDDI-Verzeichnissen)

Wie findet man Dienste?

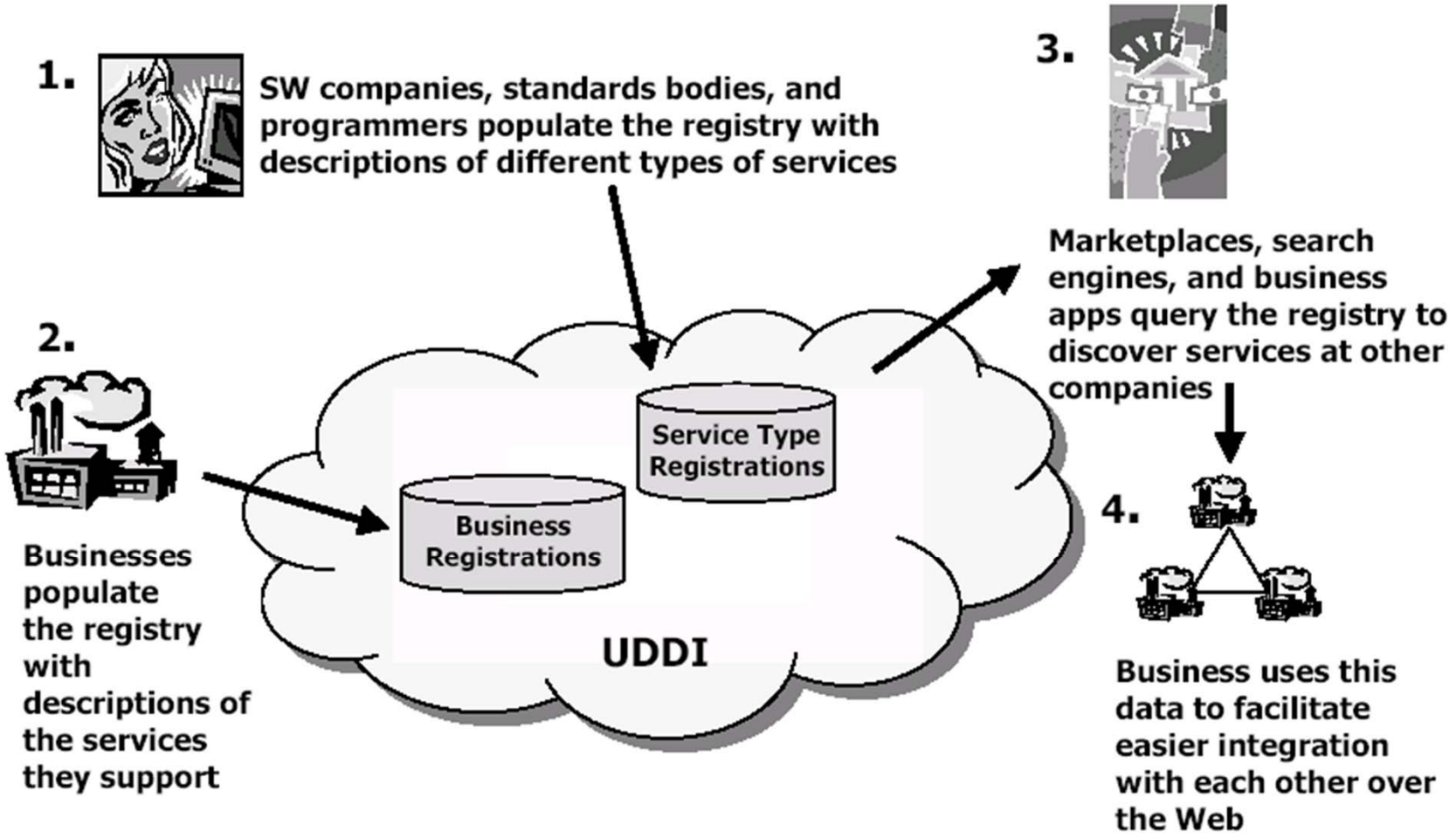
Wir brauchen ein global verfügbares Verzeichnis!

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

UDDI

(Universal Description, Discovery & Integration)

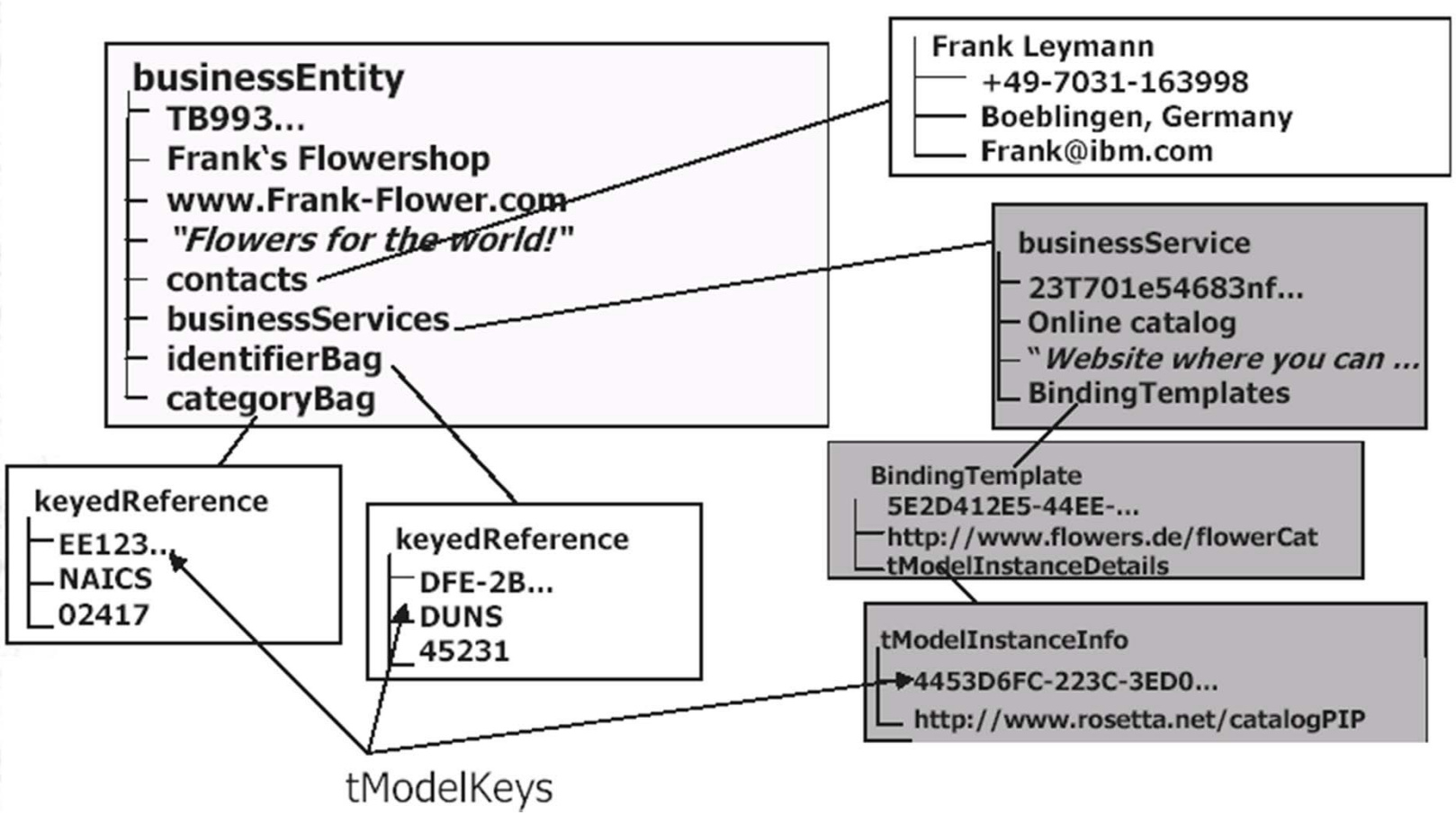
Wie arbeitet UDDI



Registry-Daten für UDDI

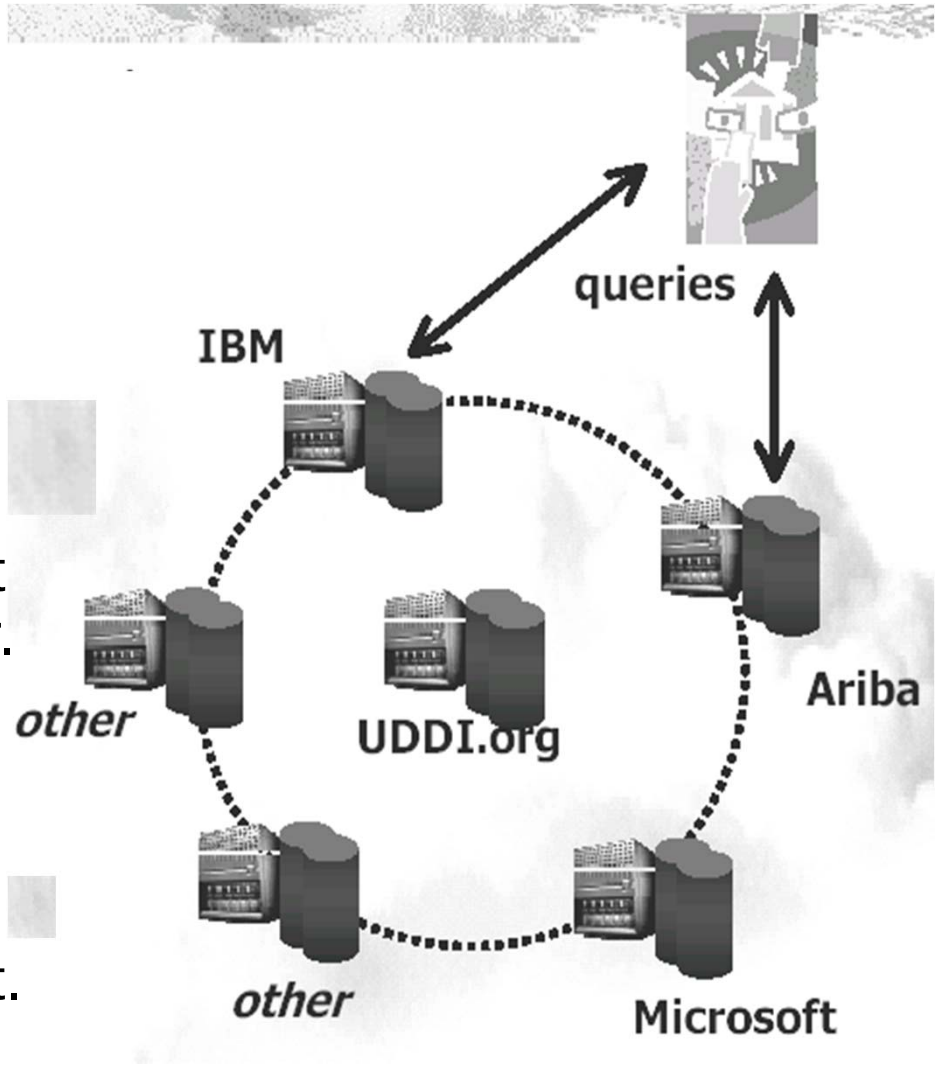
- Firmen registrieren öffentliche Informationen über sich: UBR „UDDI Business Registry
  - Öffentlicher Verzeichnisdienst
  - Betrieben von IBM, Microsoft, NTT-Communications und SAP
  - Eingestellt Januar 2006, da nur noch SAP als Knoten (Standardisierungsstelle) verblieb.
  - Nachweis der Machbarkeit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

Beispiel einer Registrierung



Registrierung

- 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

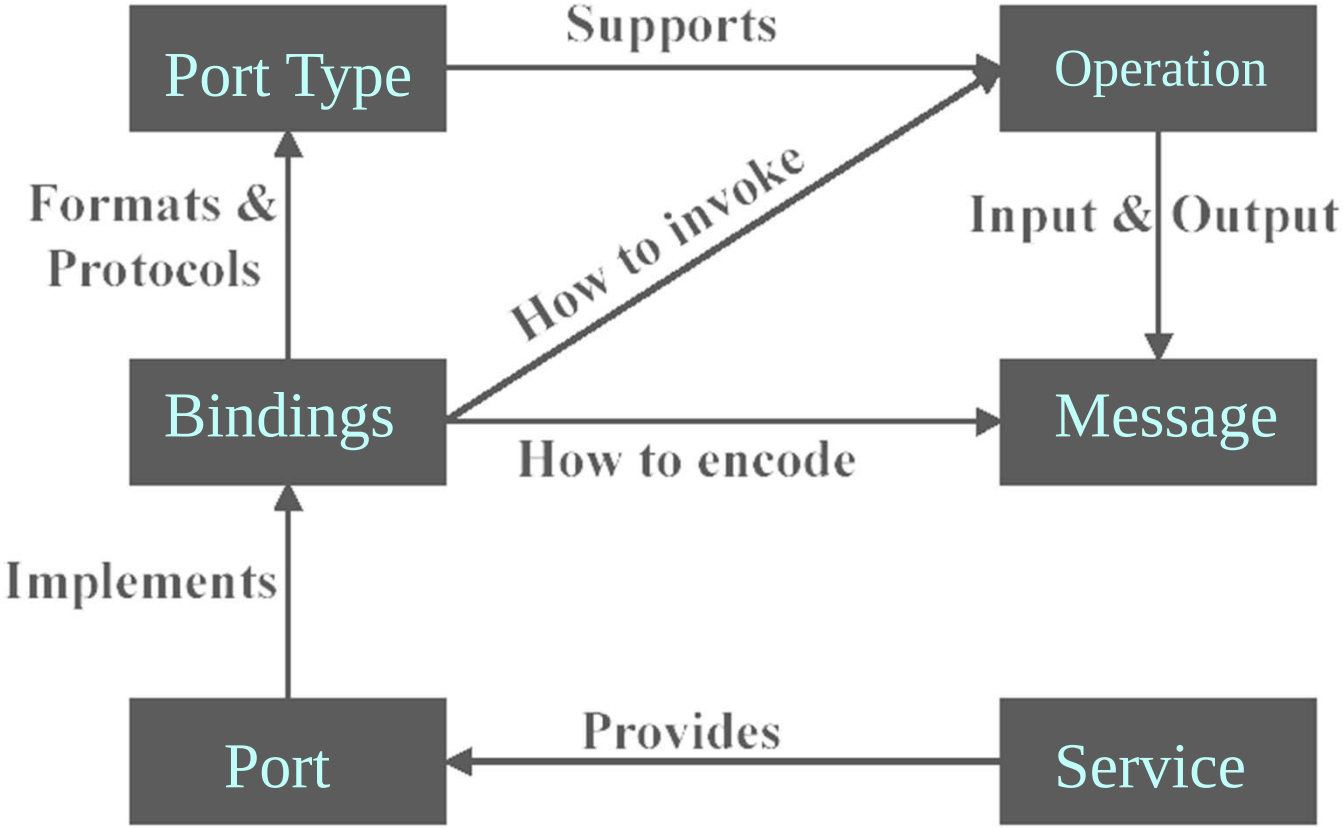
Wie definiert man neue Web Services?

- 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!

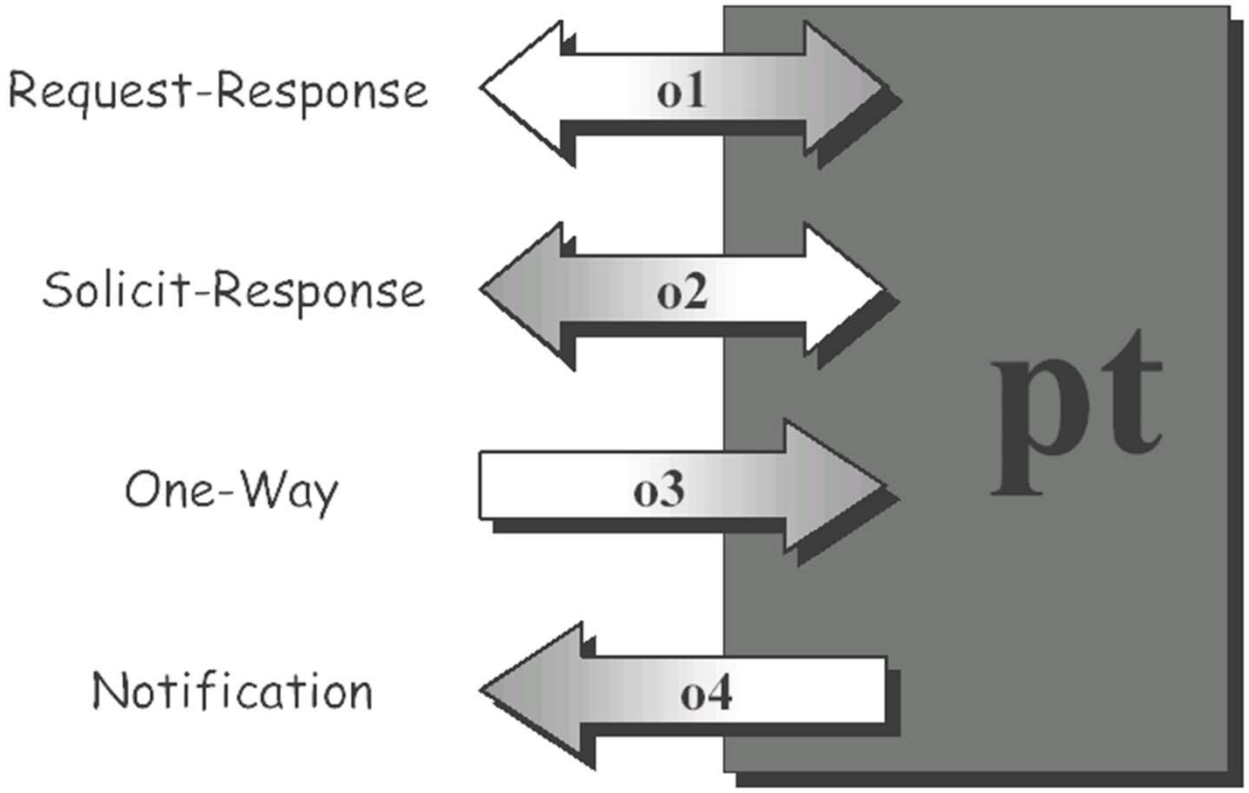
Benötigt wird ein Sprache zur Spezifikation von Diensten und zur Beschreibung, wie andere die Dienste koppeln (Binding):

WSDL (Web Services Description Language)

WSDL Bestandteile



Port Typen: graphische Darstellung



Beispiel

```
<xml version="1.0"?>
<definitions name = "StockQuoteService"
xmlns:tns=http://muelle.com/stockquote.wsdl
xmlns:xsd=http://muelle.com/sockquote.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:timeInstant"/>
        </message>
        <message name="GetLastTradePriceResponse">
                <part name="result" type="xsd:float"/>
        </message>
        <portType name="StockQuotePortType">
                <operation name="GeLastTradePrice">
                        <input message="tns:GetLastTradePriceRequest"/>
                        <output message="tns:GetLastTradePriceResponse"/>
                </operation>
        </portType> …
```

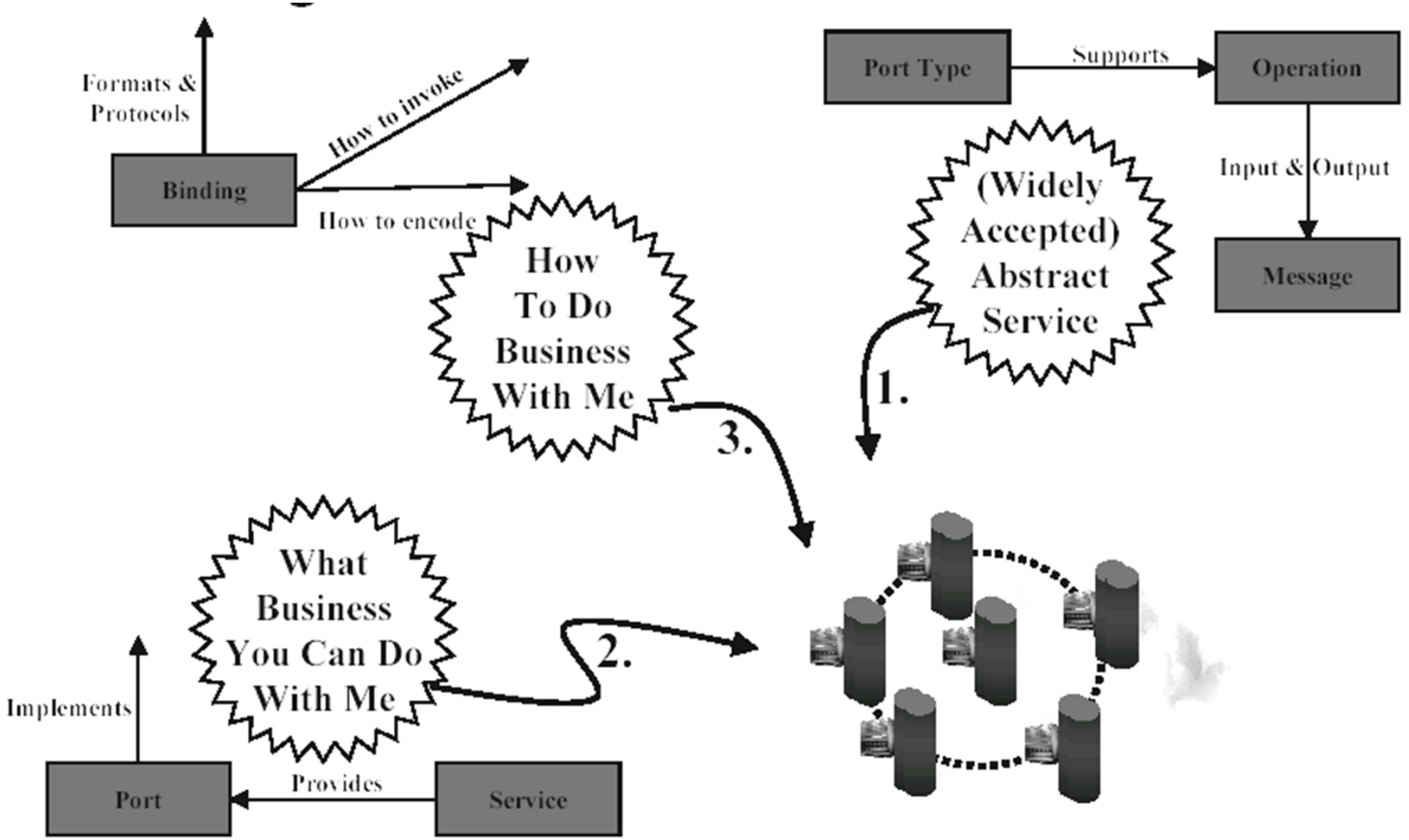
Beispiel

```
<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

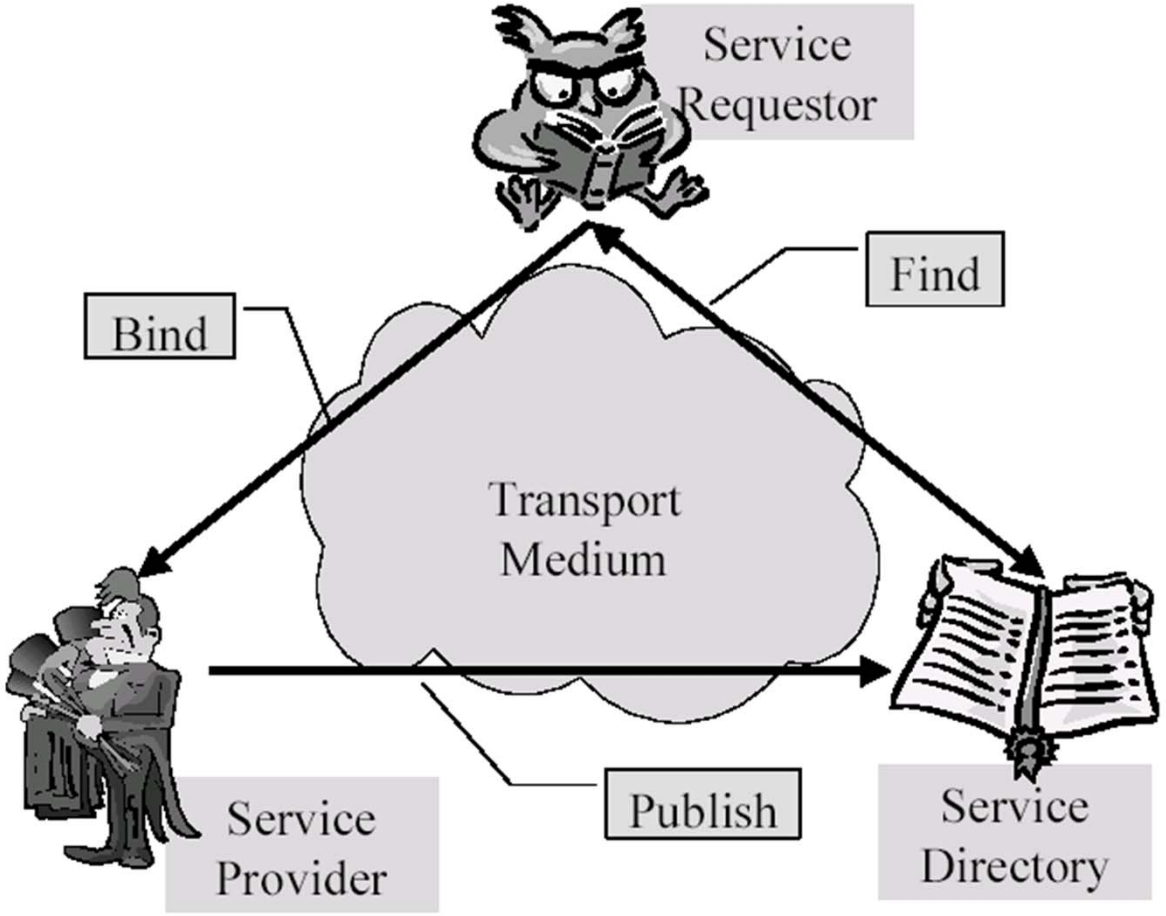
```
<service name = “StockQuoteService”>
<documentation>My first service</documentation>
<port name= “StockQuotePort”
    binding=“tns:StockQuoteSoapBinding”>
<soap:address
    location= “http://muelle.com/stockquote”/>
</port>
</service>
```

Wie macht man Dienste verfügbar

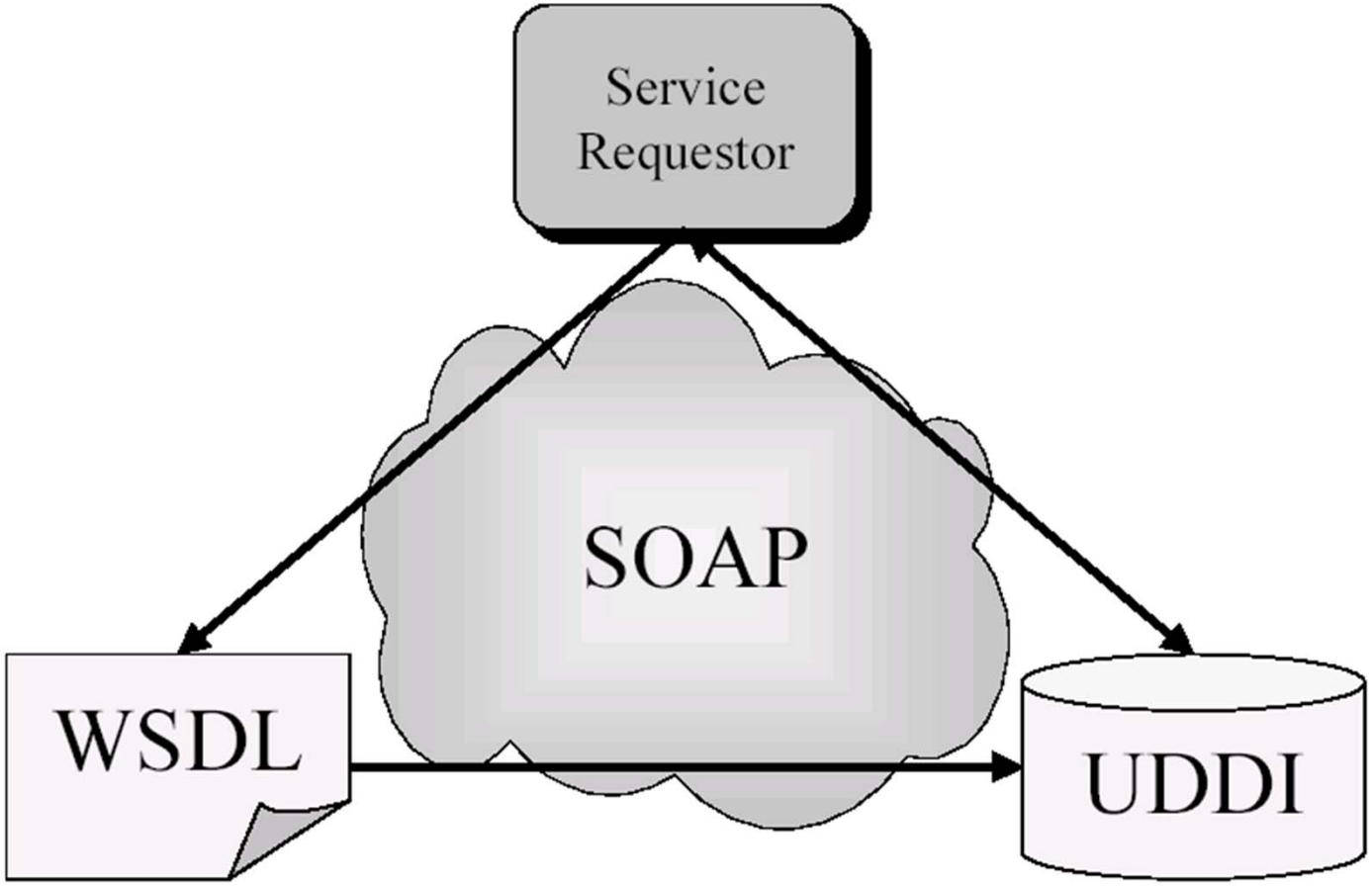


Zusammenhang mit der Service-Orientierten Architektur

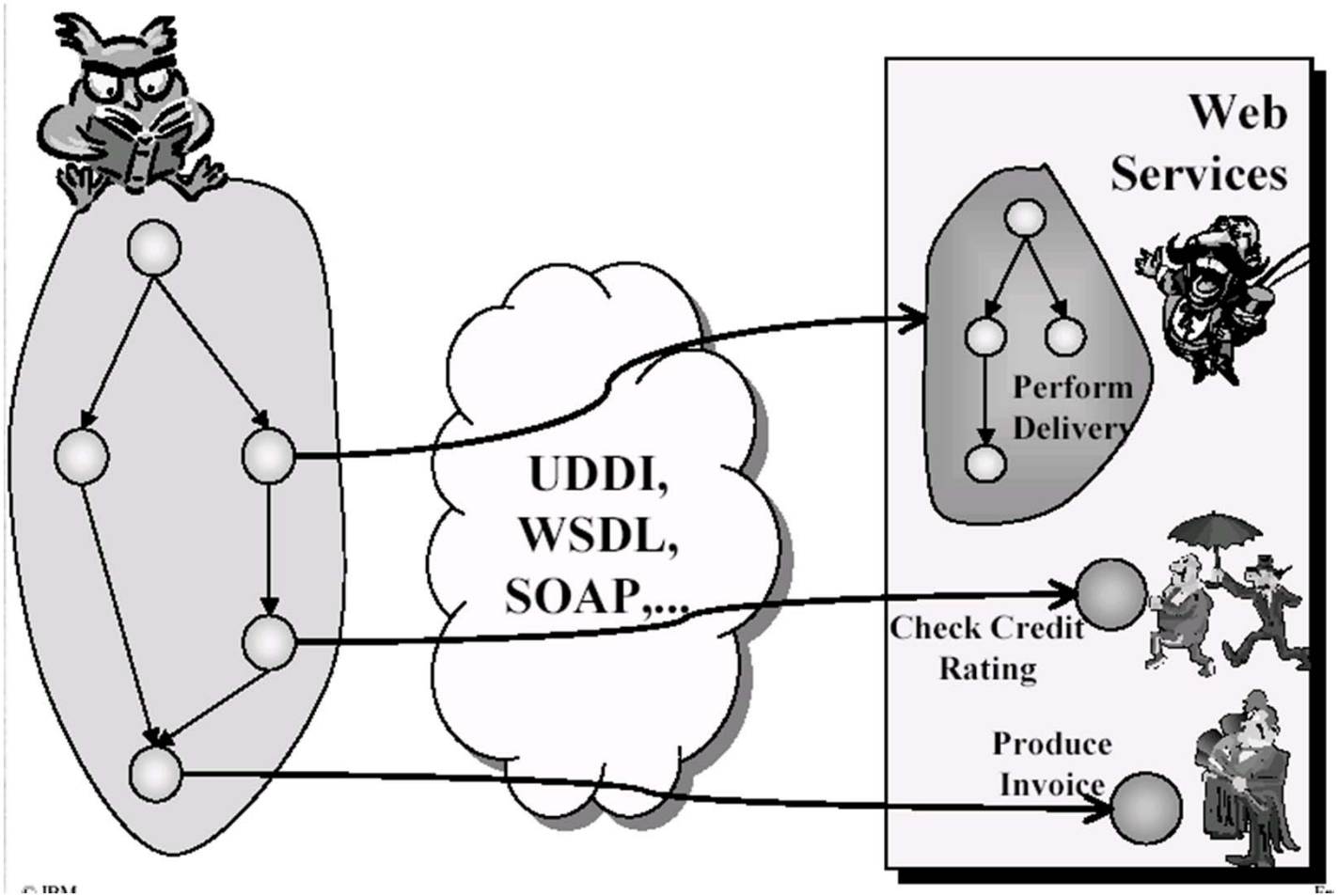
SOA - Wiederholung



...und Zusammenhang mit Web Services



SOA in Aktion



Was wird benötigt?

Wir brauchen eine Sprache die es erlaubt...

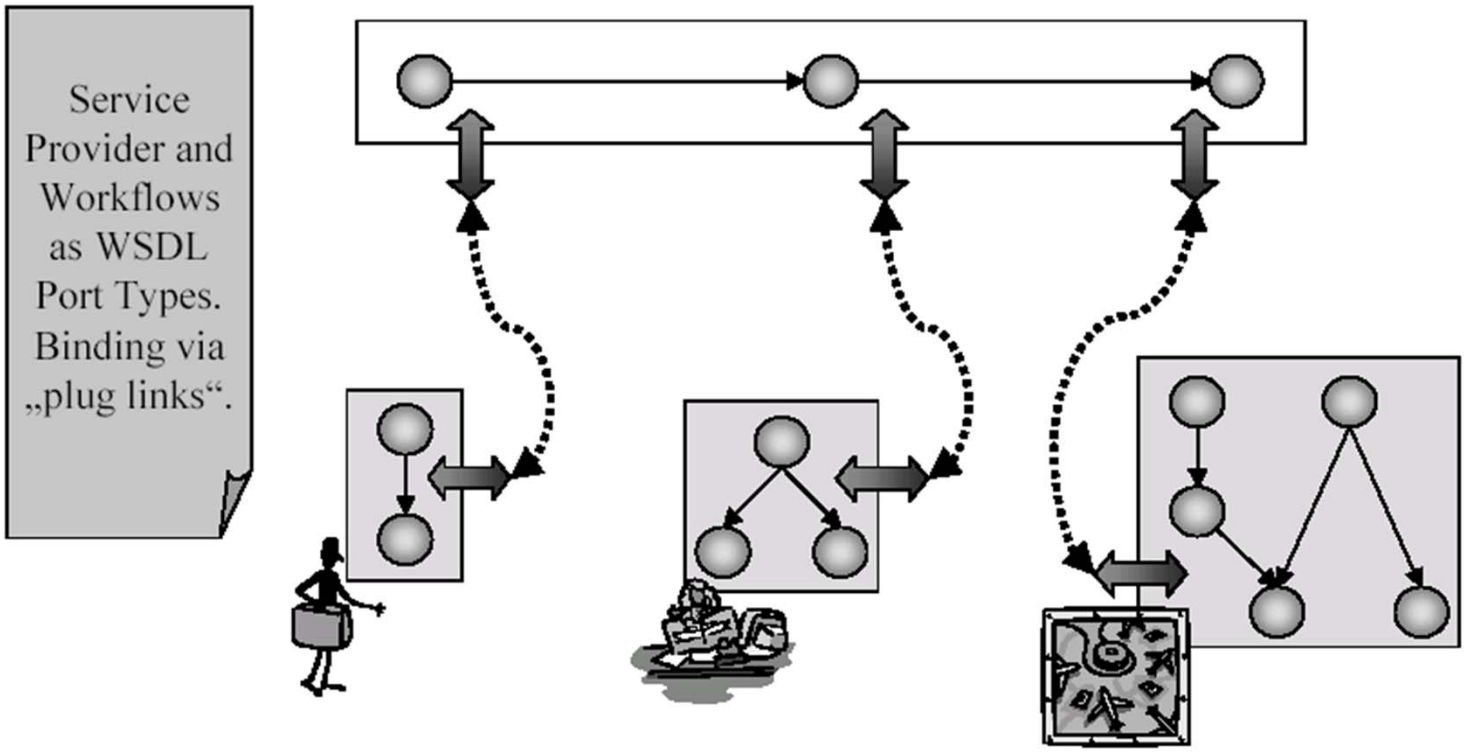
- Abläufe (flows) als Web Services zu spezifizieren
- Abläufe zwischen Web Services zu spezifizieren
- Neue Web Services als Aggregationen über Web Services zu spezifizieren
- ...eine beliebige Mischung davon zu spezifizieren

Web Services Choreographie

Standardisierungsbemühungen:

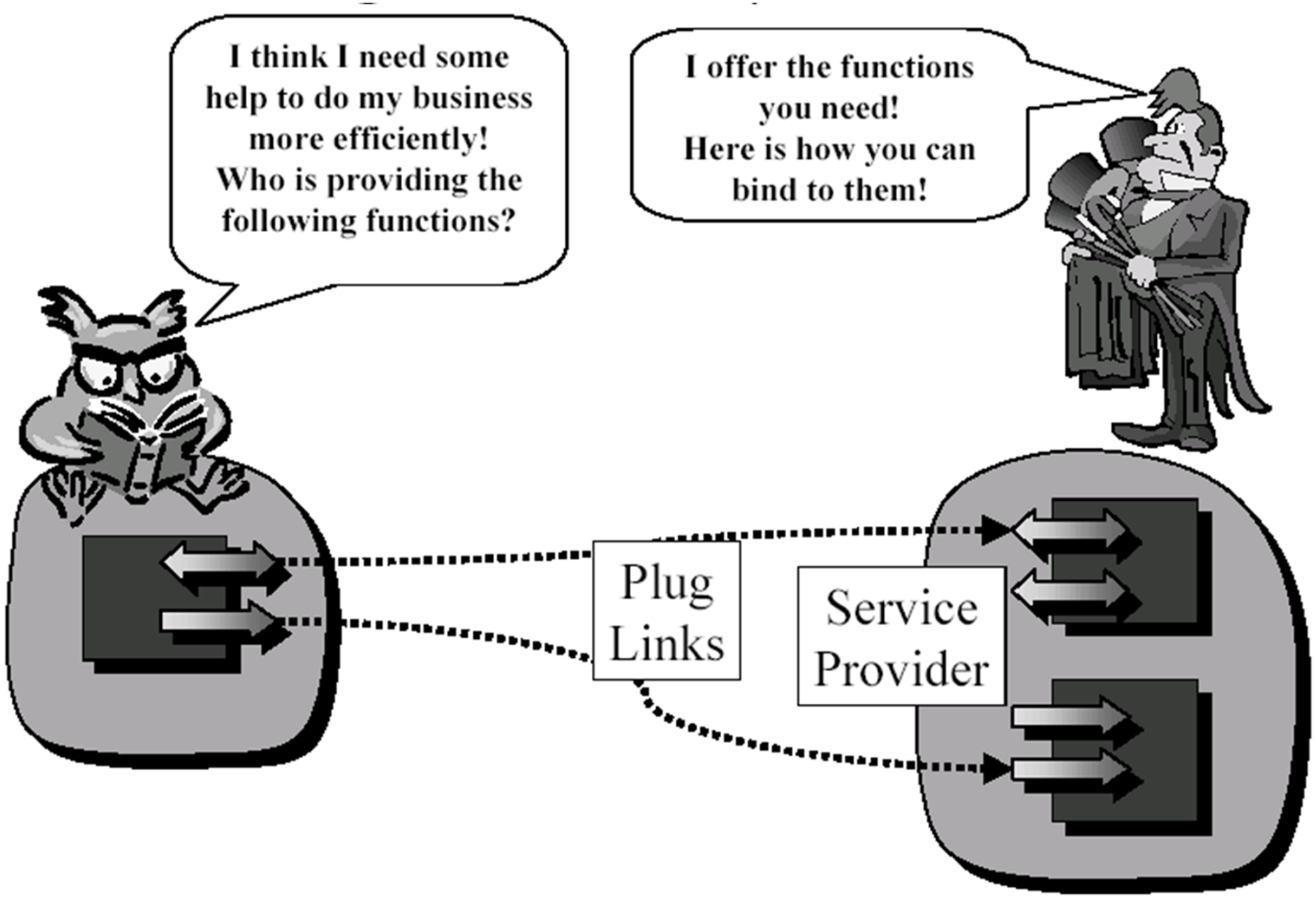
- WSFL (Web Services Flow Language) - IBM
- XLANG – Microsoft
- BPeL4WS (Business Process execution Language for Web Services) baut als Ansatz zur Standardisierung auf WSFL und XLANG auf.

Beziehung zur Service-orientierten Architektur

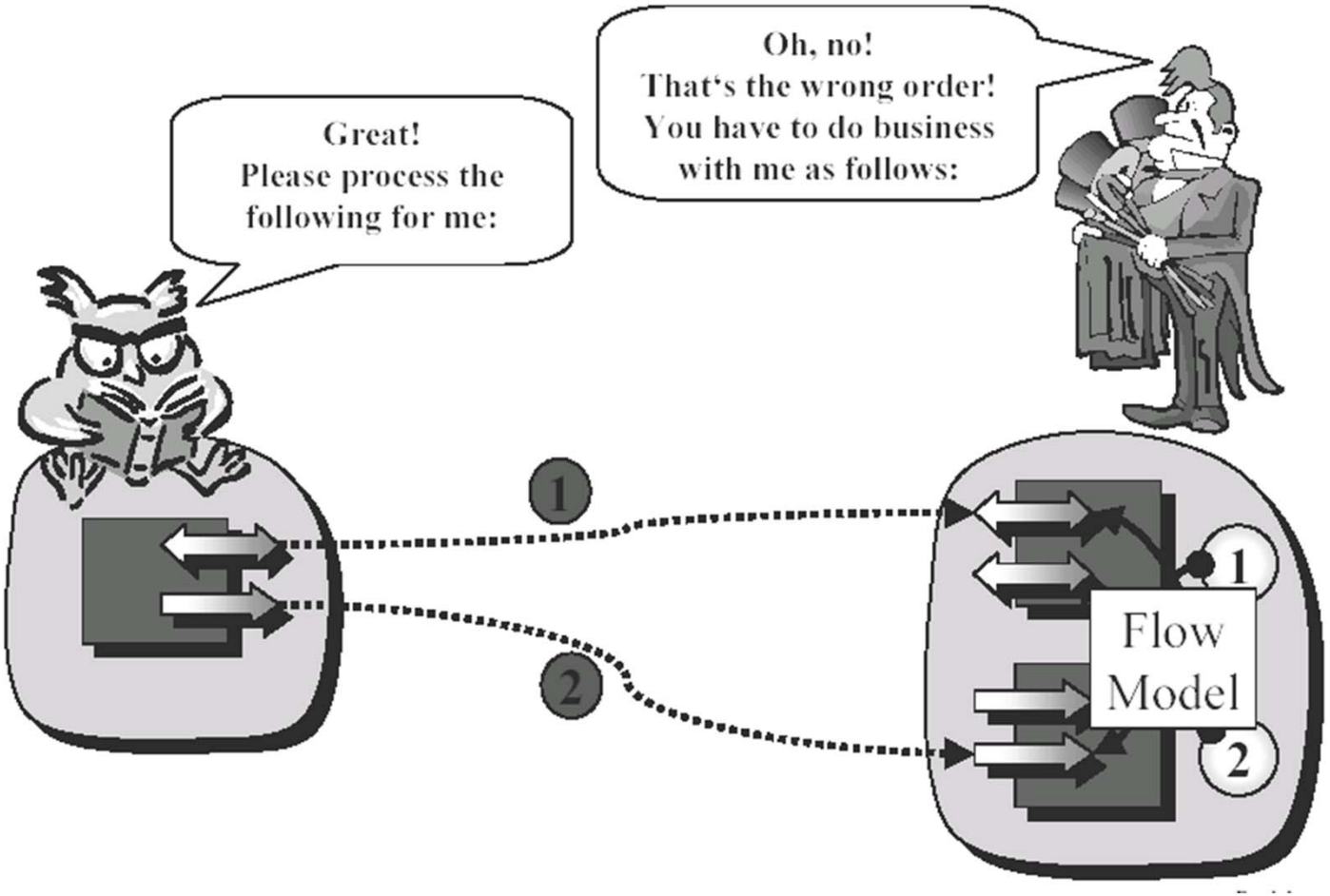


Eine abstraktere Sicht: High-Level Szenarios

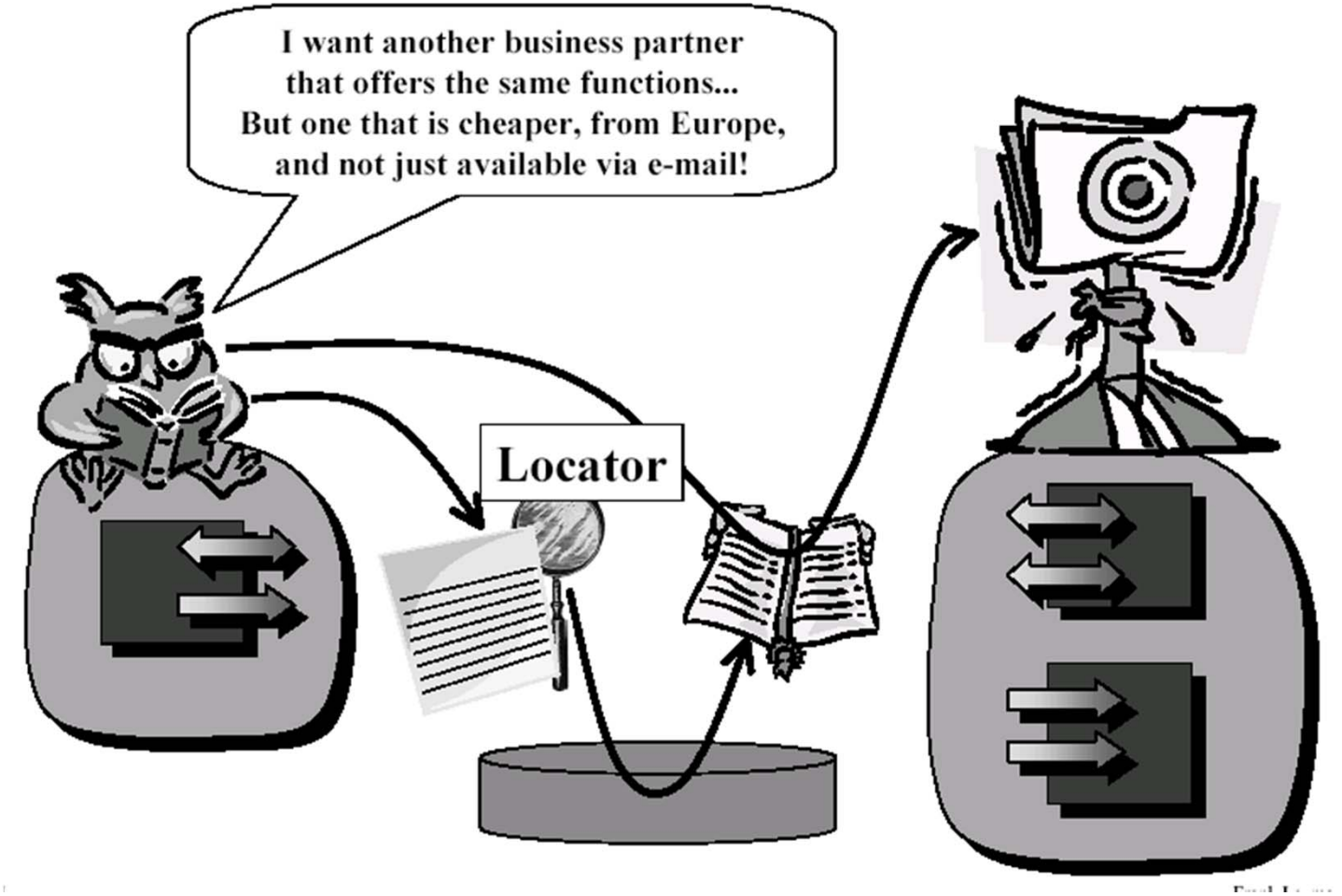
Partnerschaften errichten



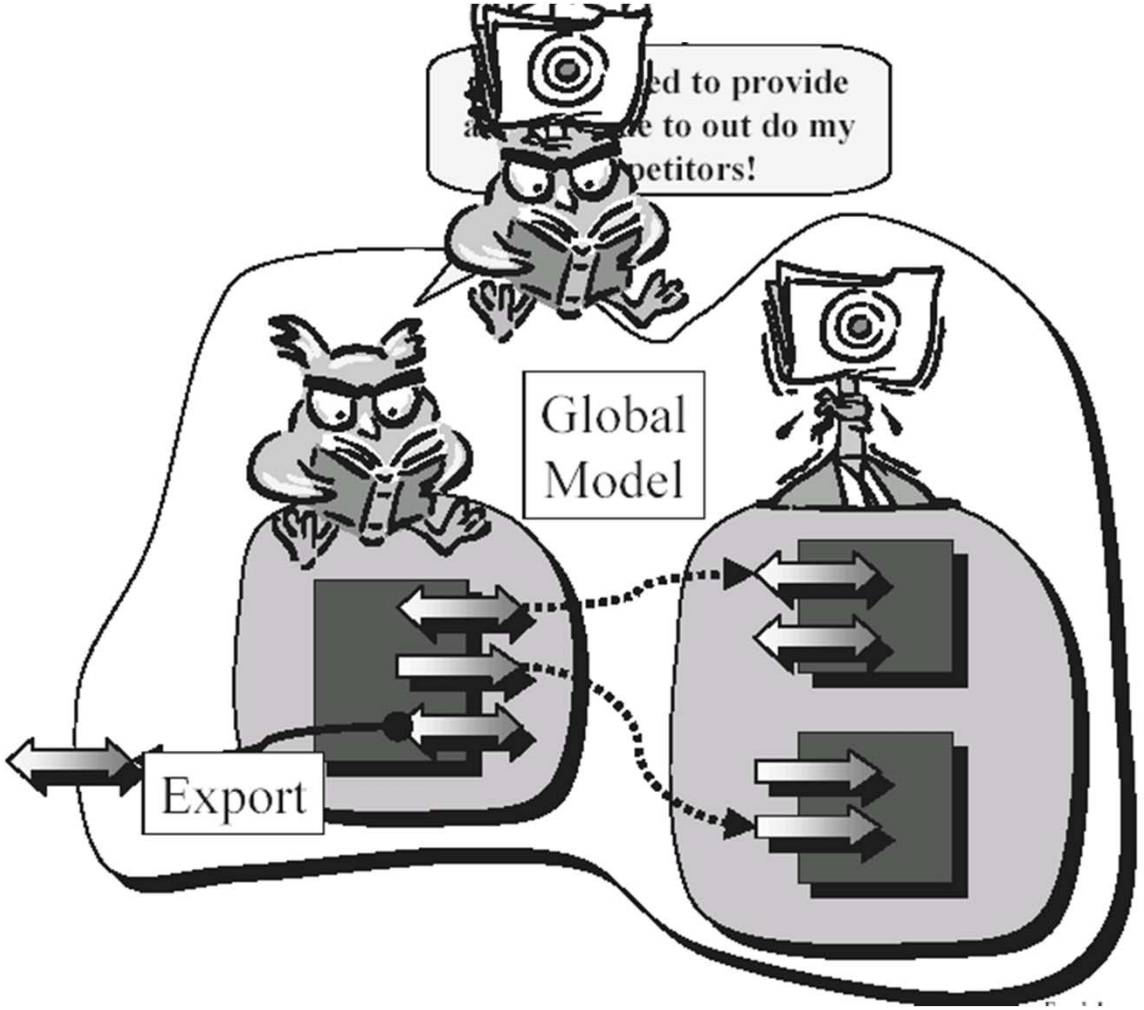
Ordnung zählt!



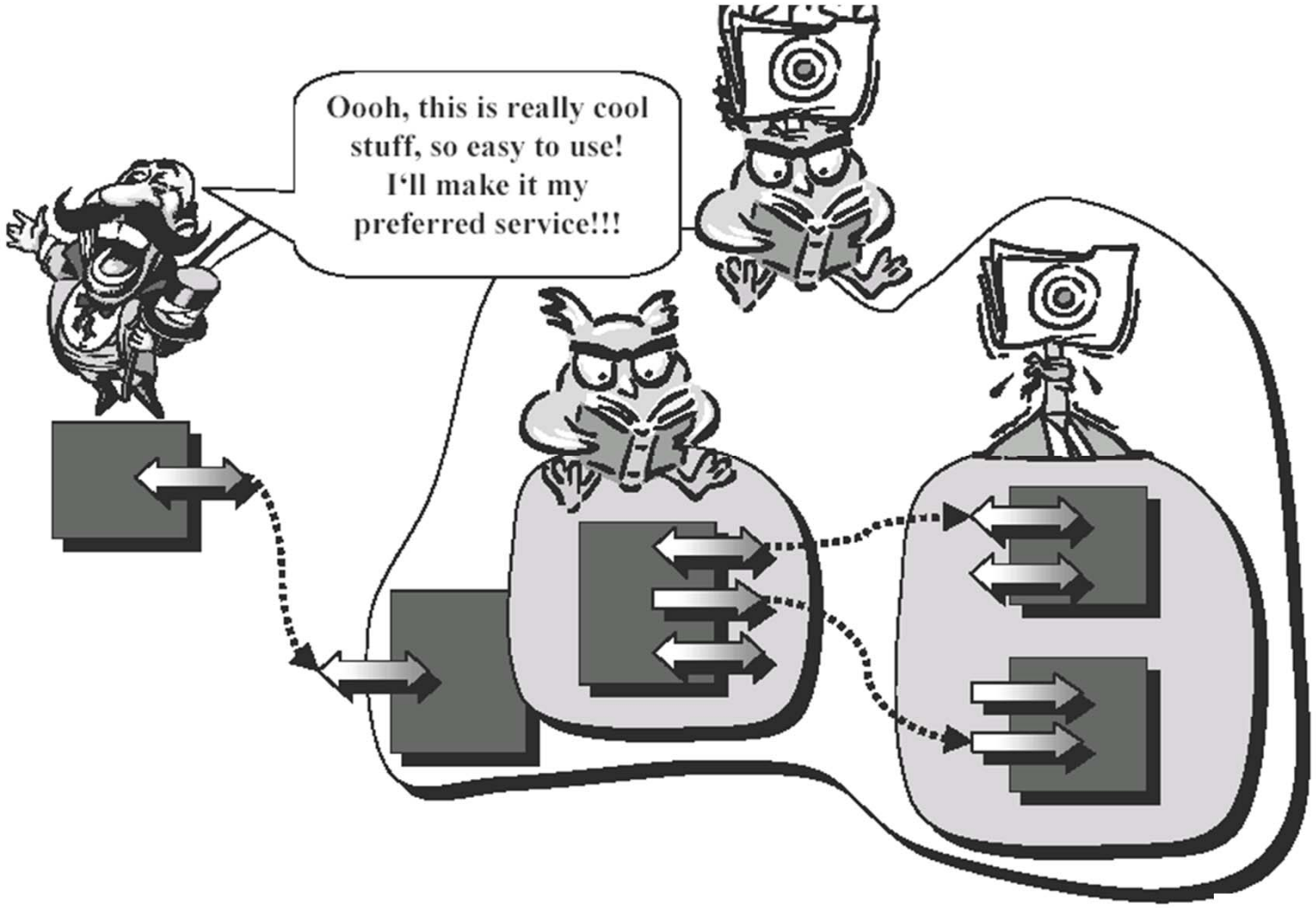
Flexibilität!



Zusatznutzen durch Partnerschaft!

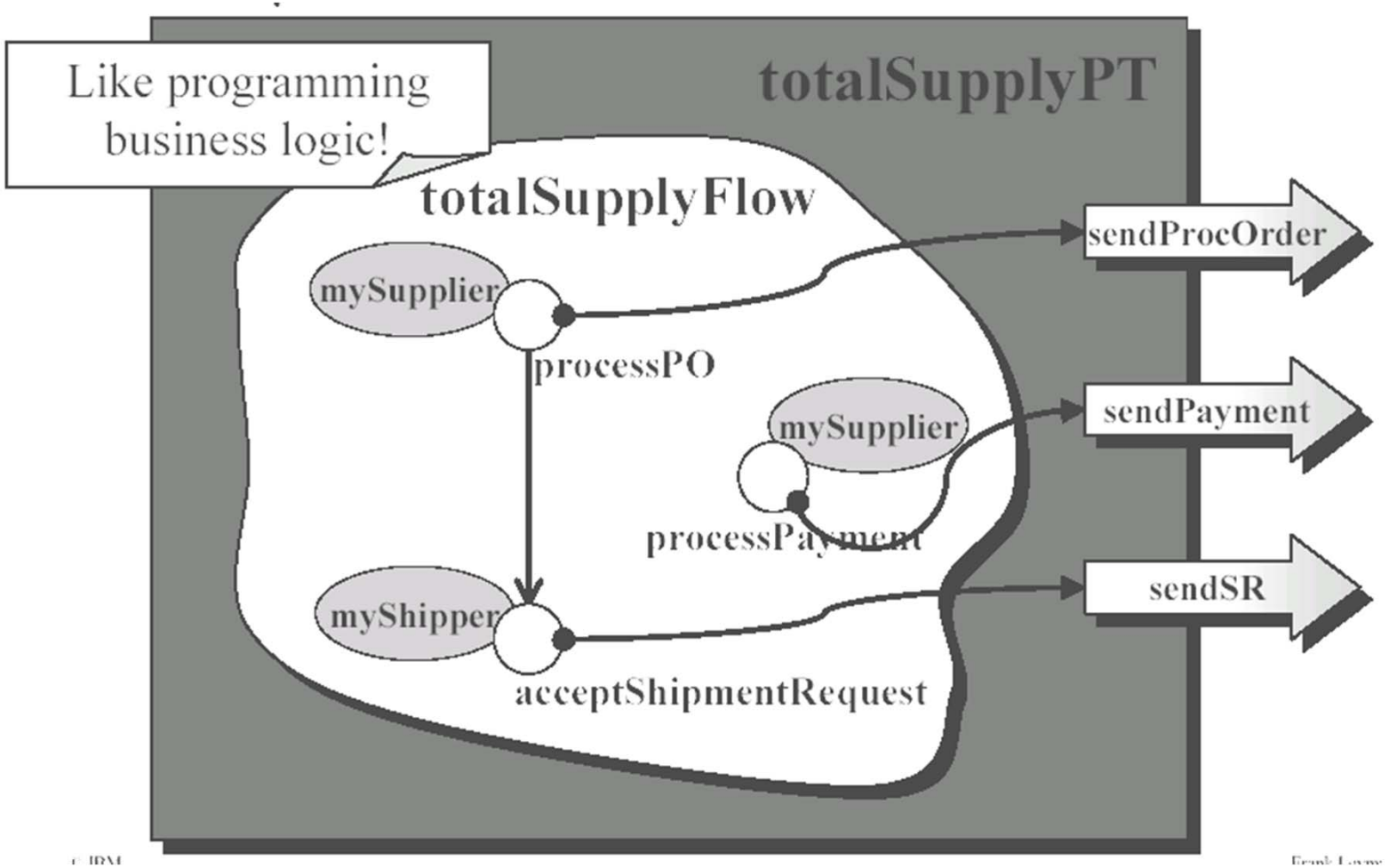


Zusatznutzen durch Partnerschaft (2)!

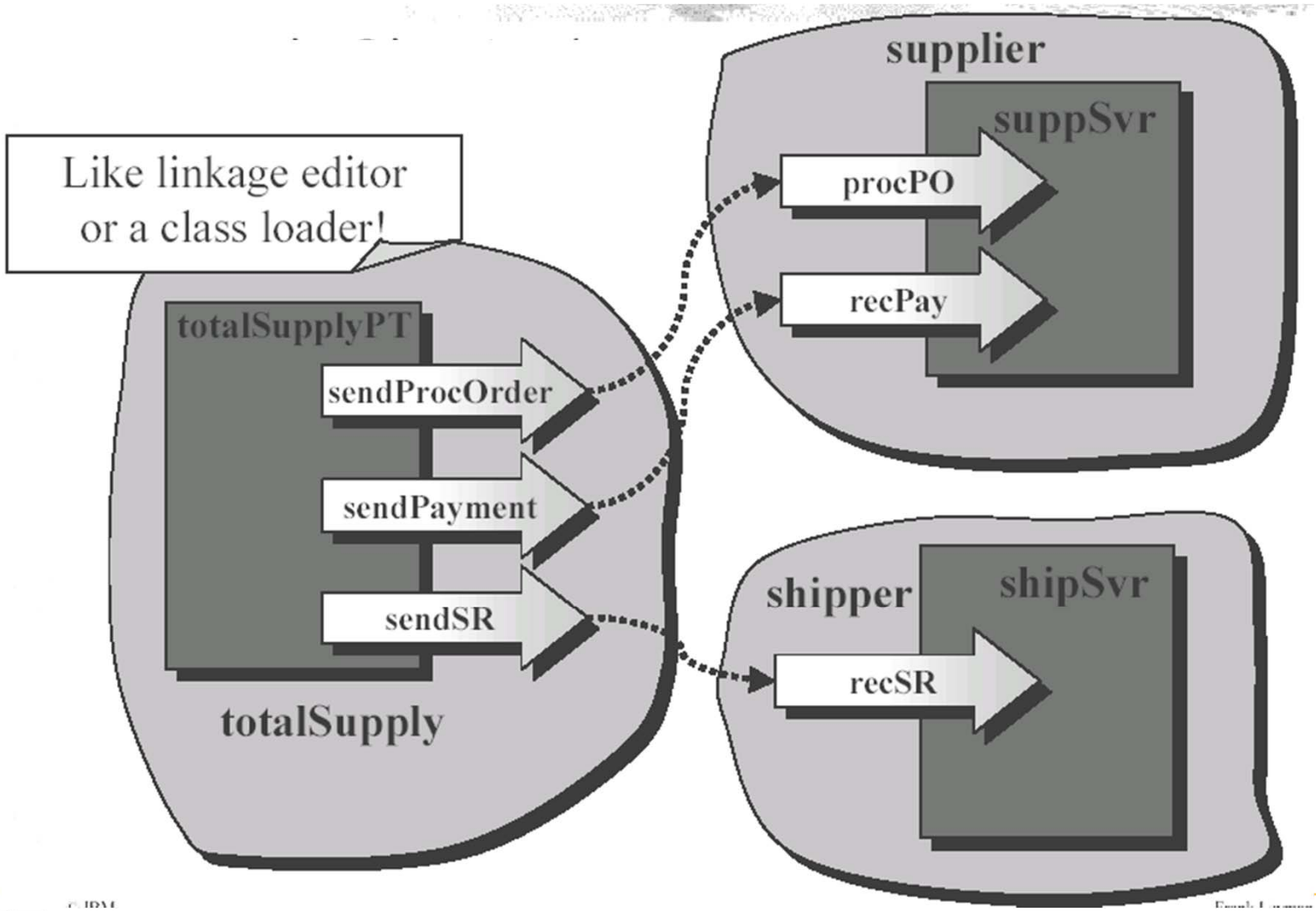


Ein globales Beispiel!

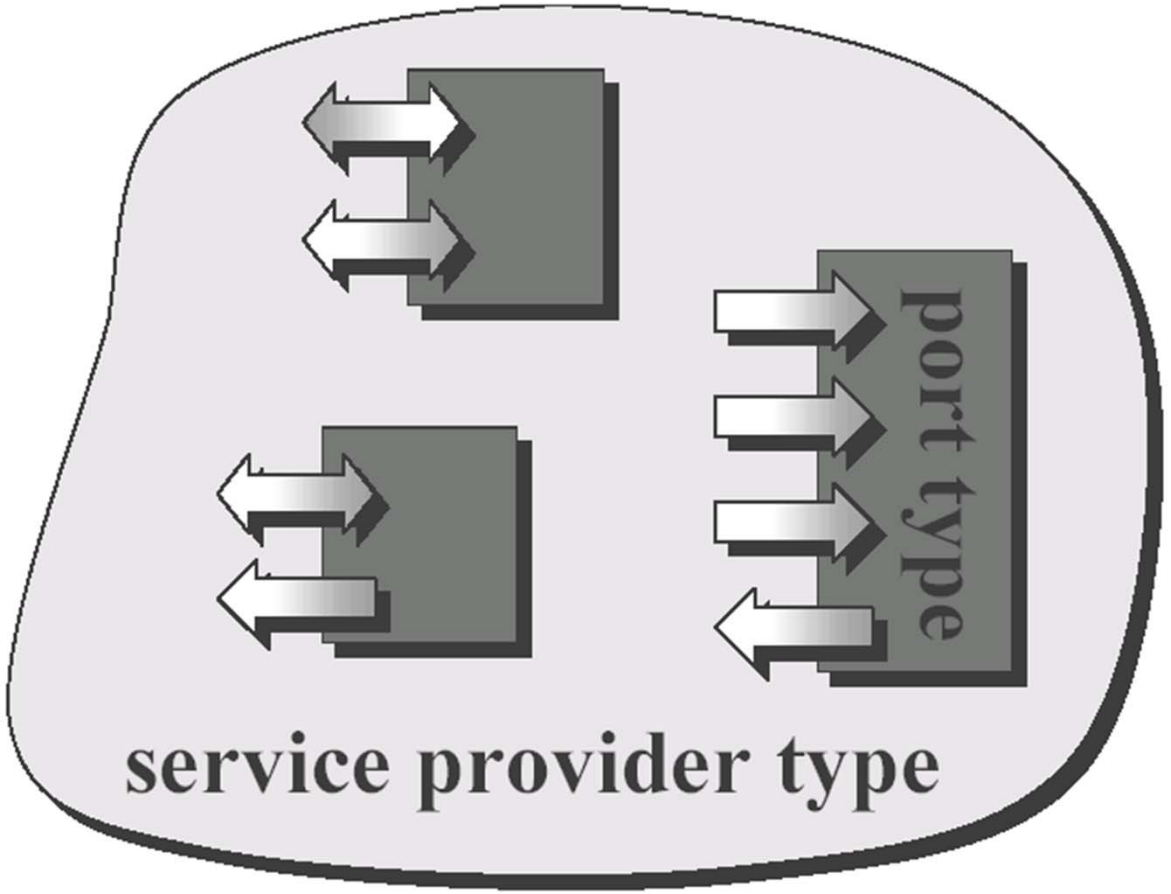
Beispiel-Ablaufmodell



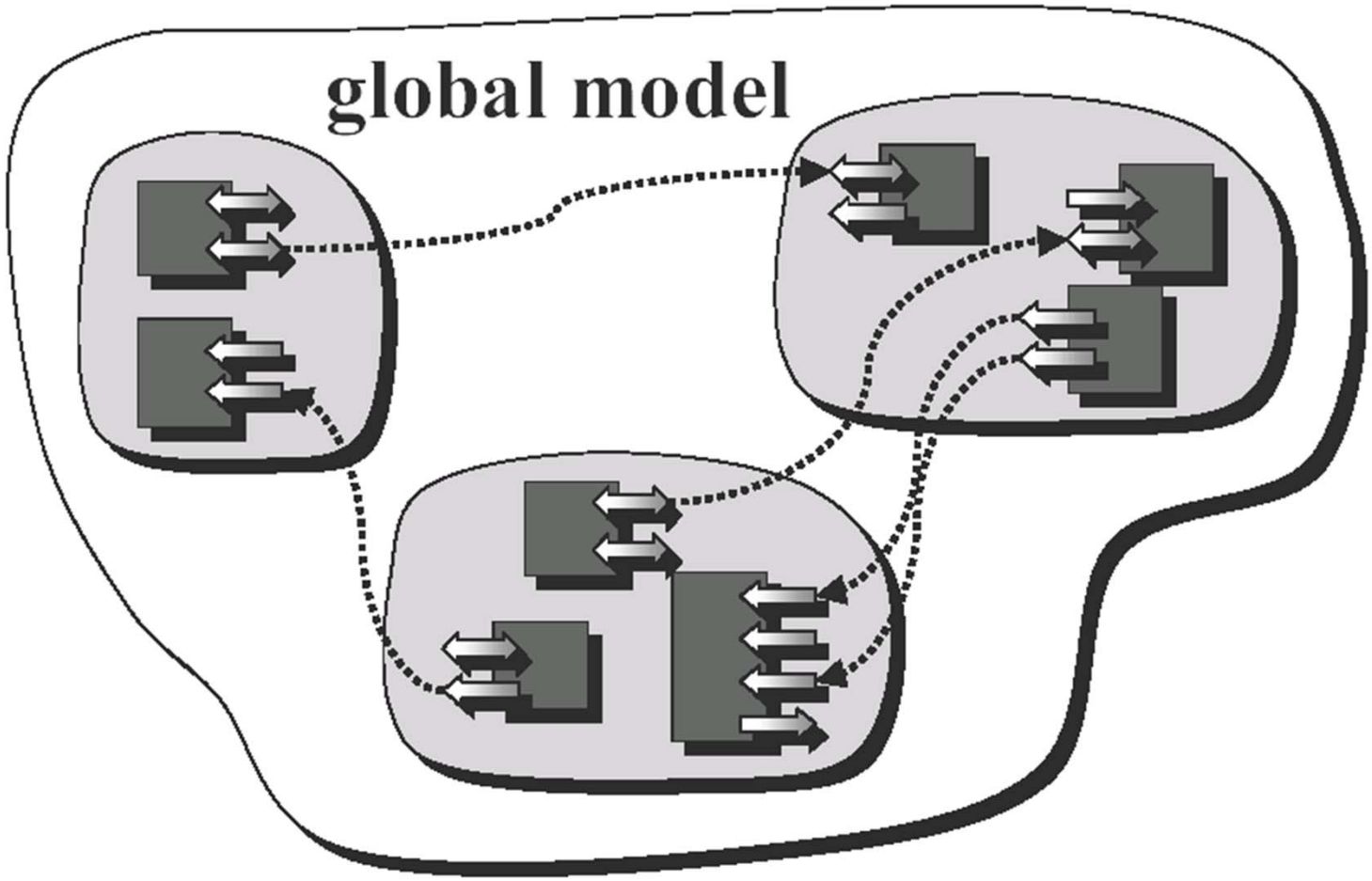
Beispiele für Plug Links



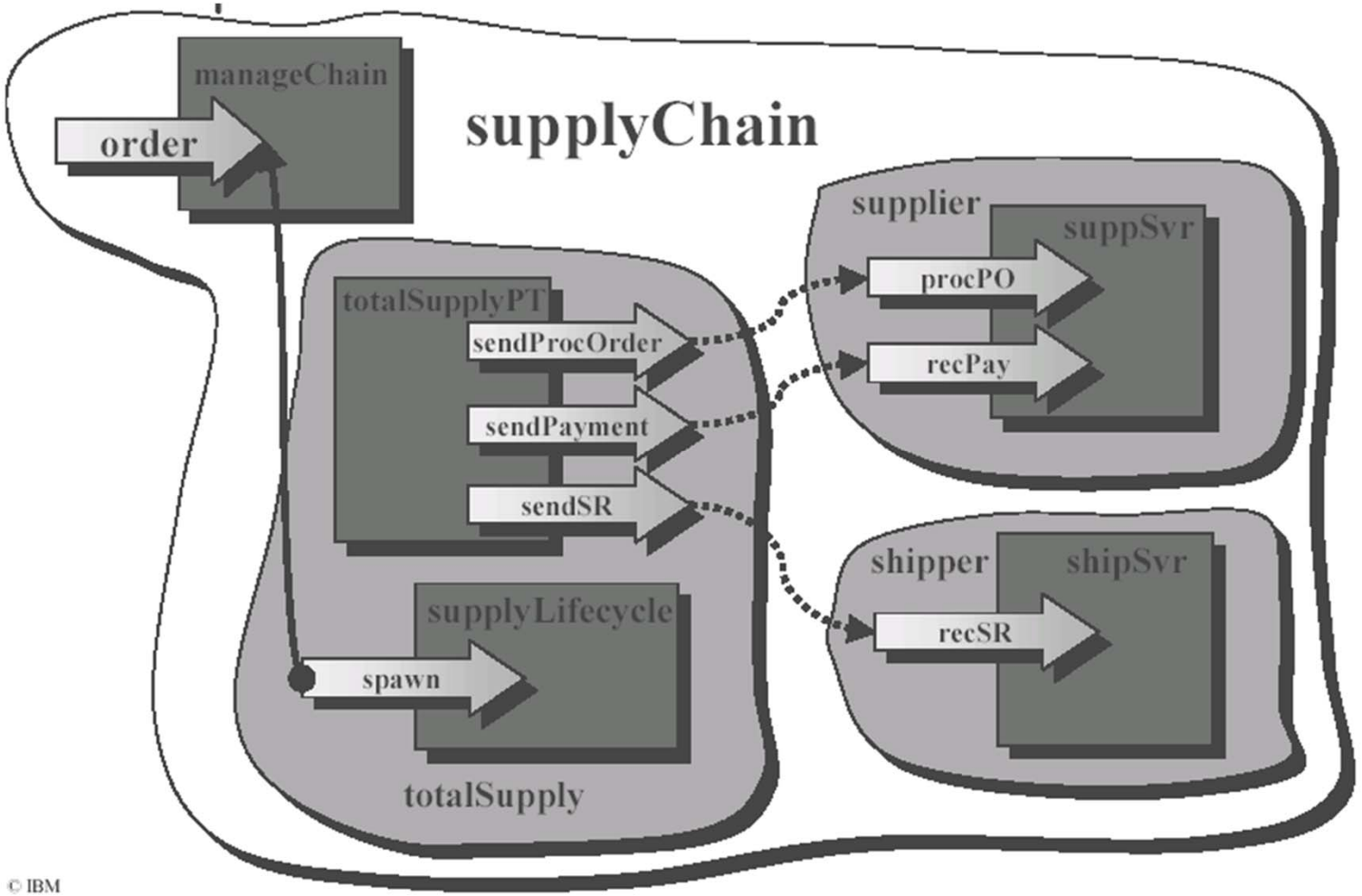
Service Provider Typen



Globales Modell

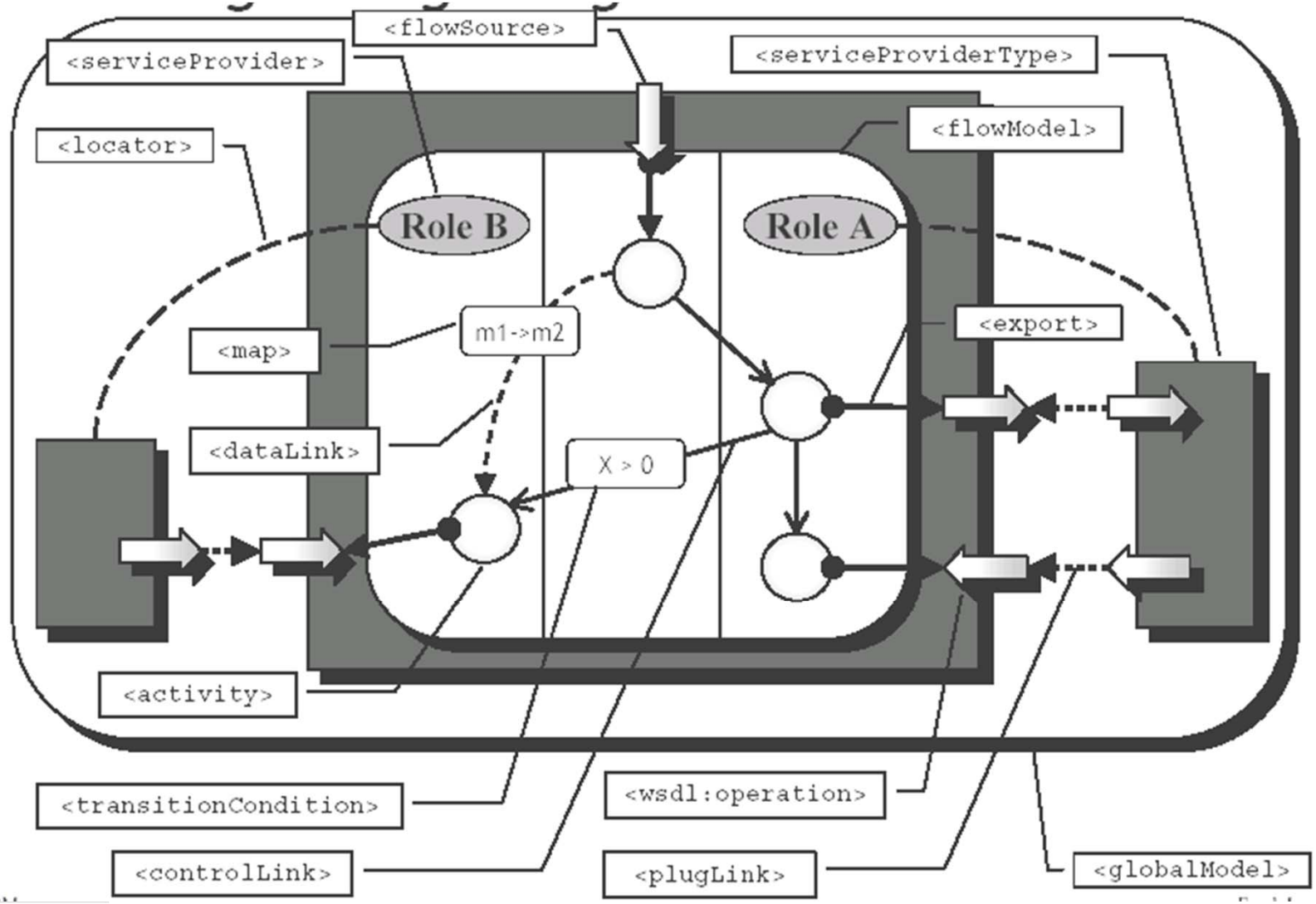


Beispiel eines globalen Modells



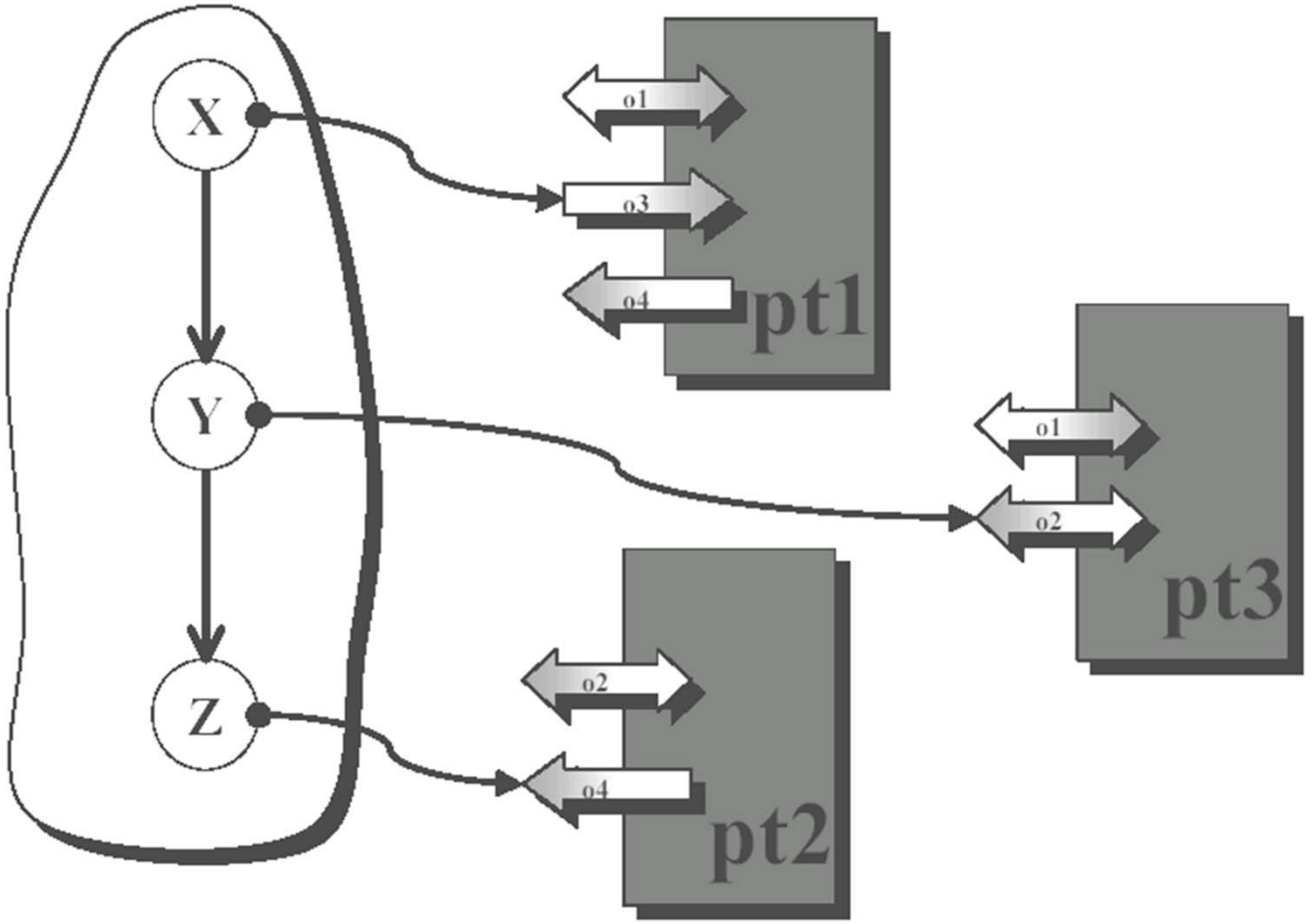
© IBM

Zusammenbringen dieser Beispiele: WSFL

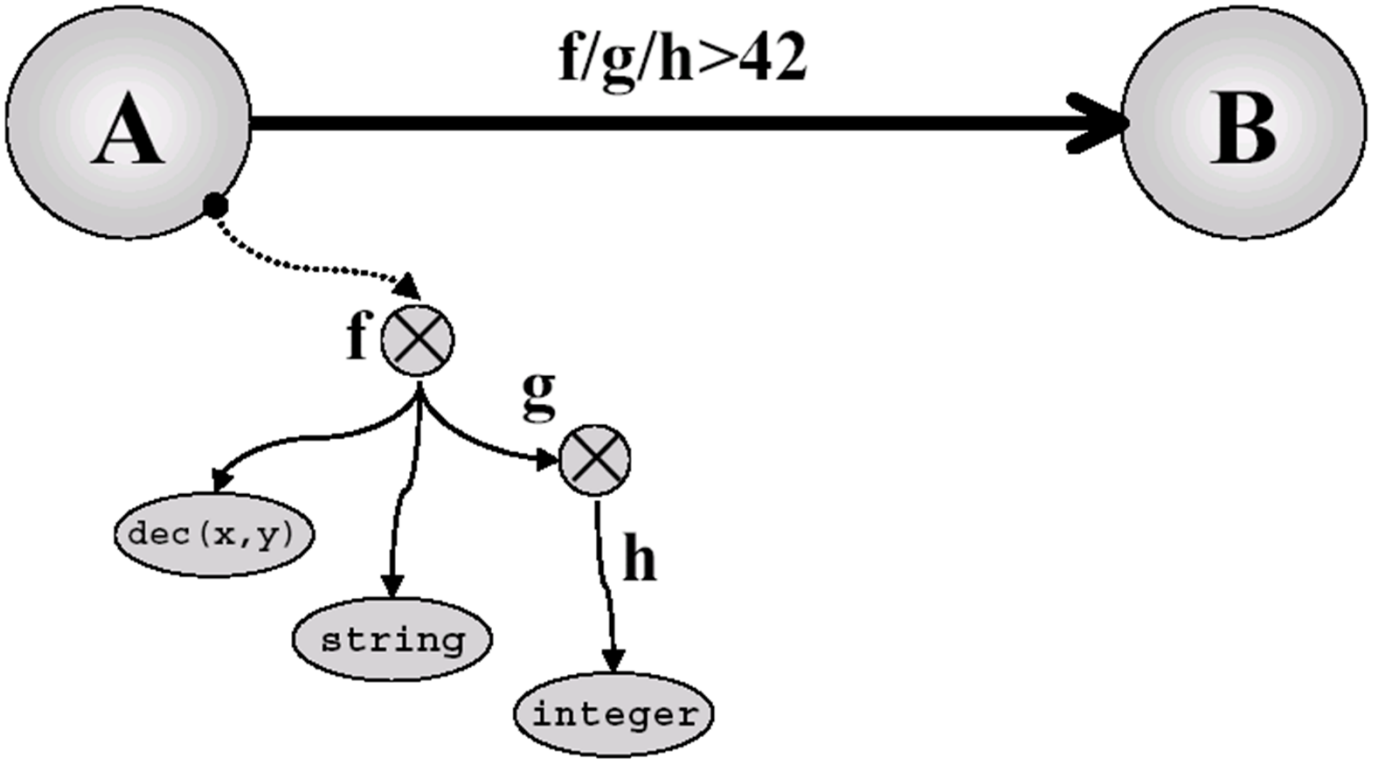


Einige Details über WSFL-Konzepte

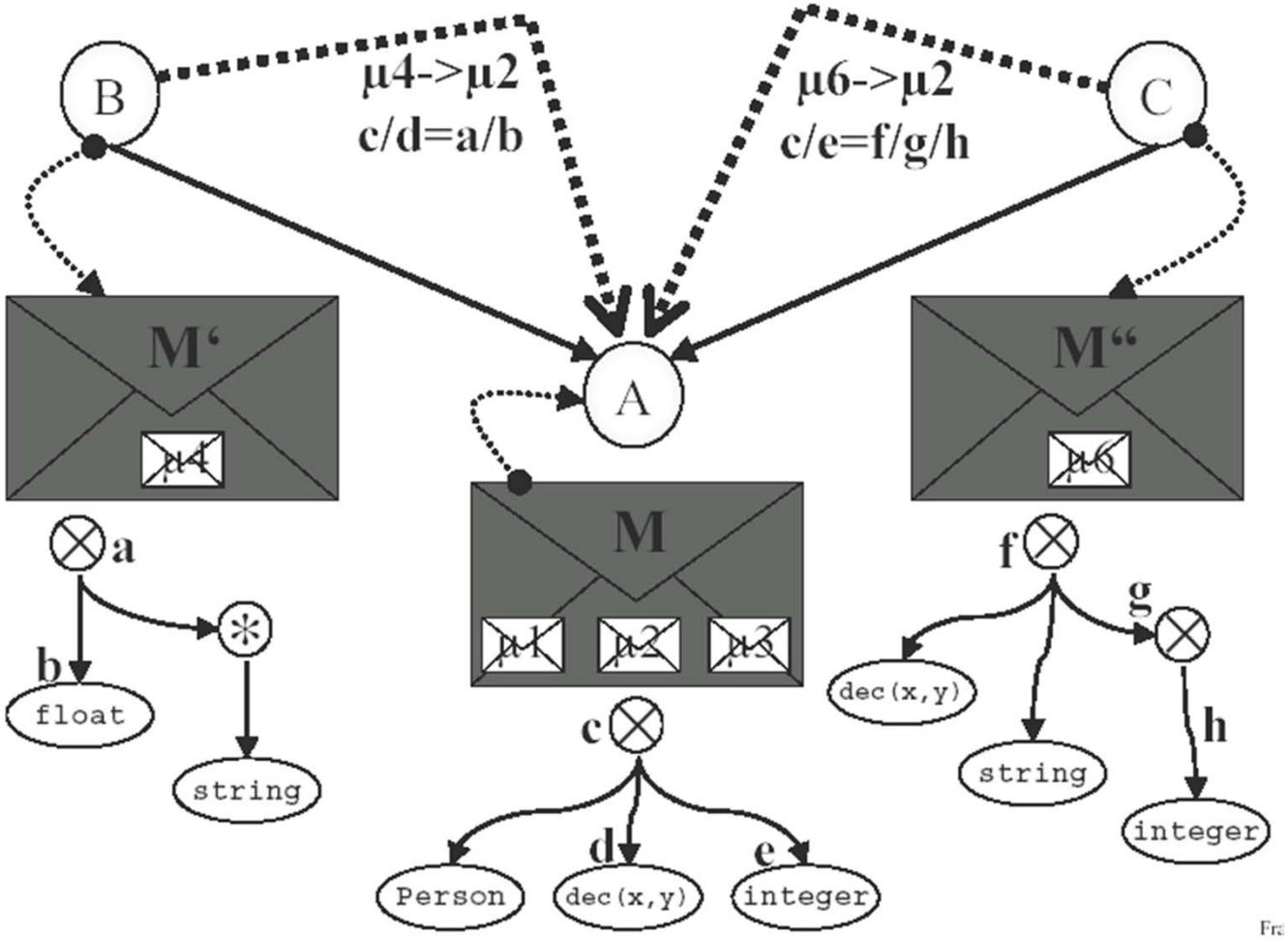
Nutzen von Port-Typen in Geschäftsprozessen



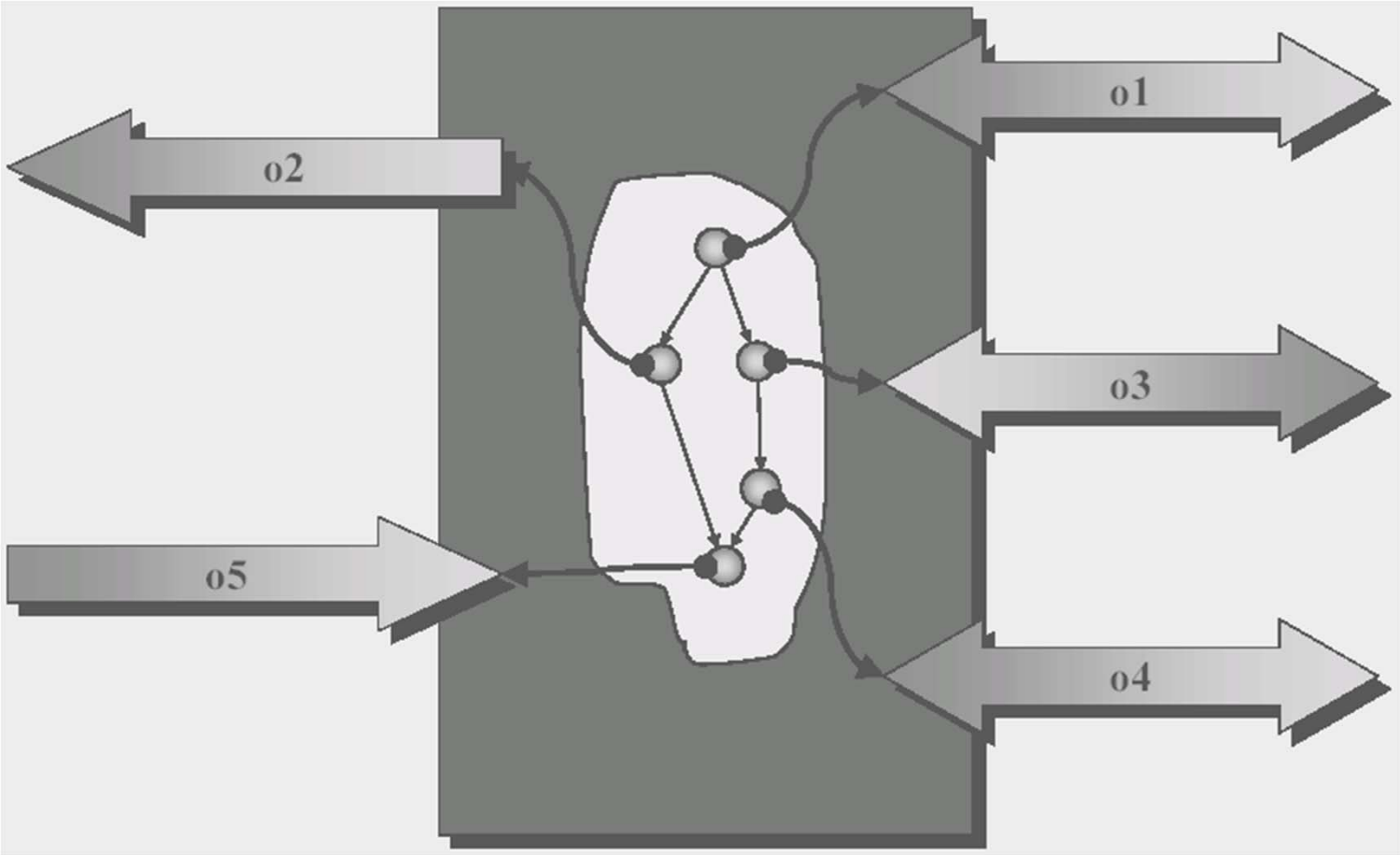
Control Links



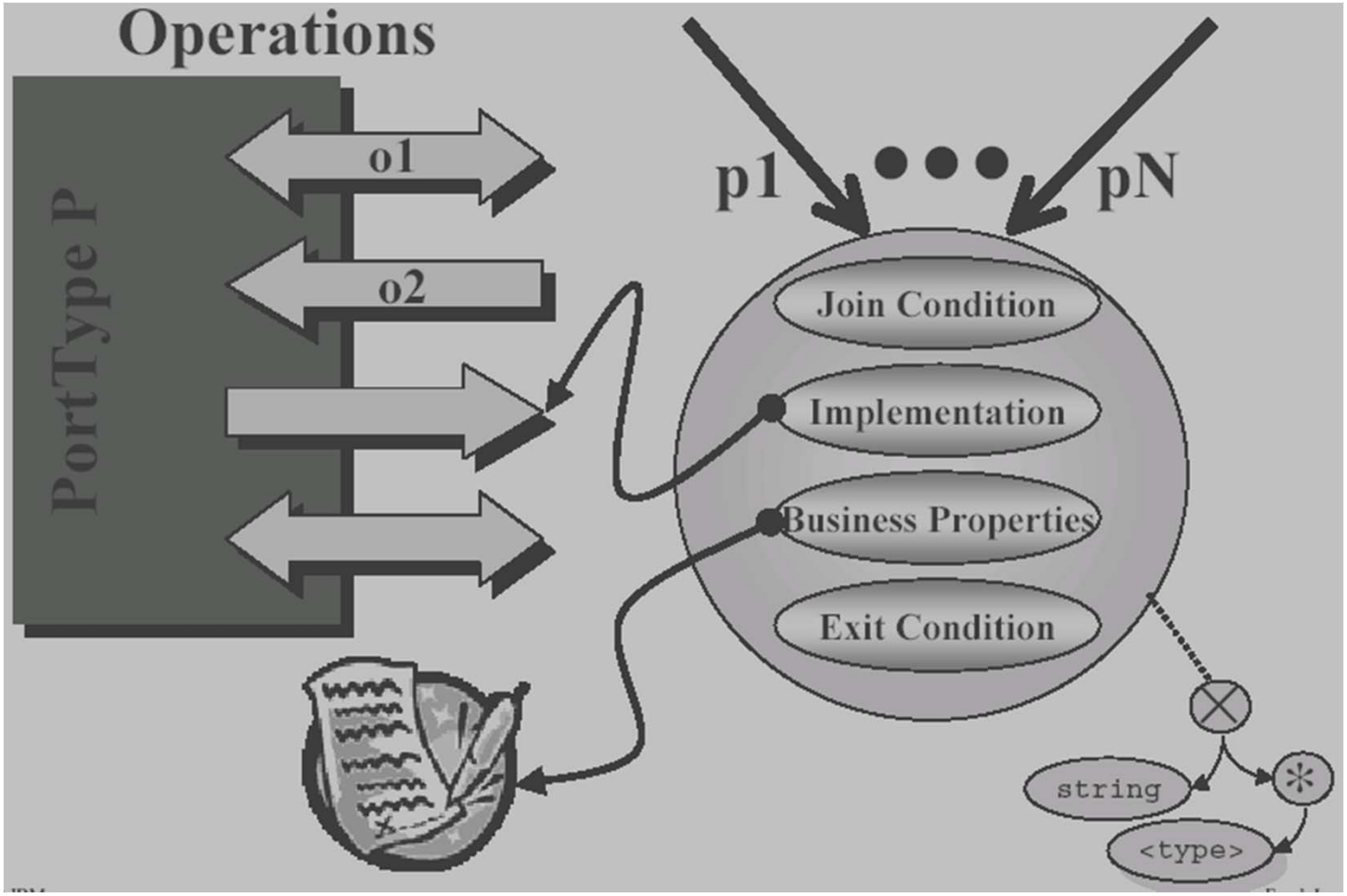
Daten Links


B
C
A
μ4->μ2
c/d=a/b
μ6->μ2
c/e=f/g/h
M‘
M“
M
μ4
μ6
μ1
μ2
μ3
a
b
float
string
c
Person
d
dec(x,y)
e
integer
f
dec(x,y)
g
string
h
integer

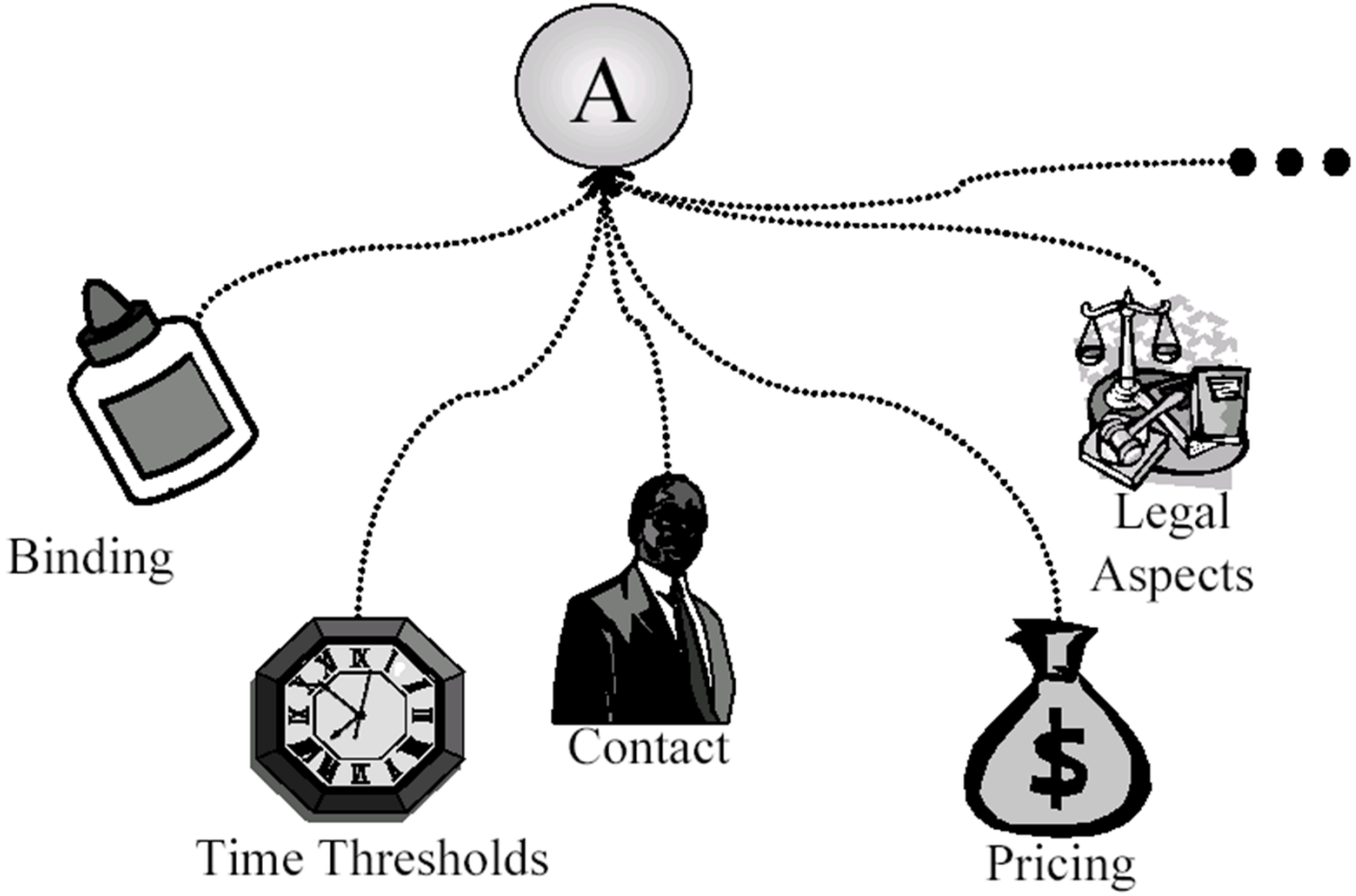
Einschränkung der Aufrufreihenfolge



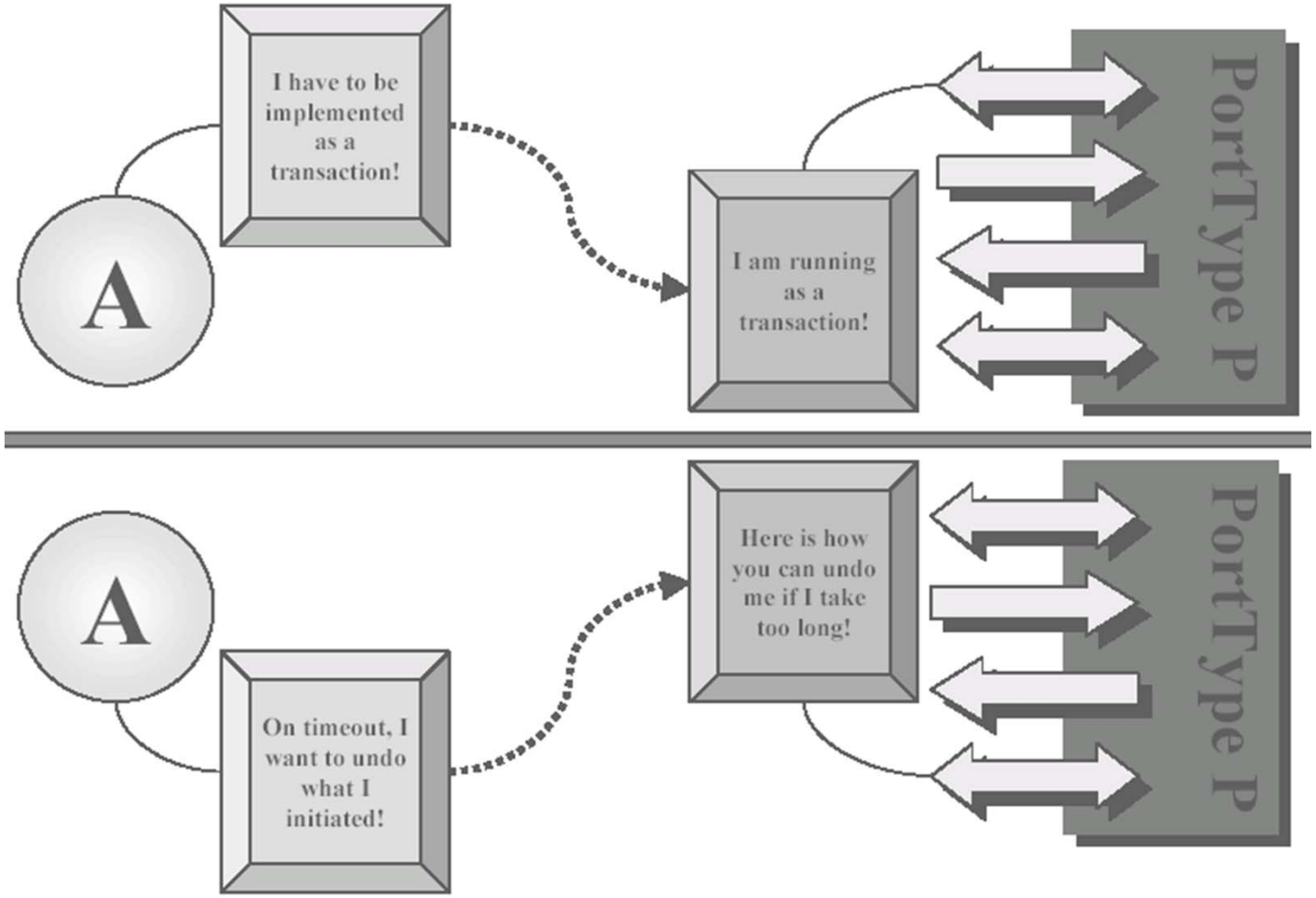
Feine Aktivitätenstruktur



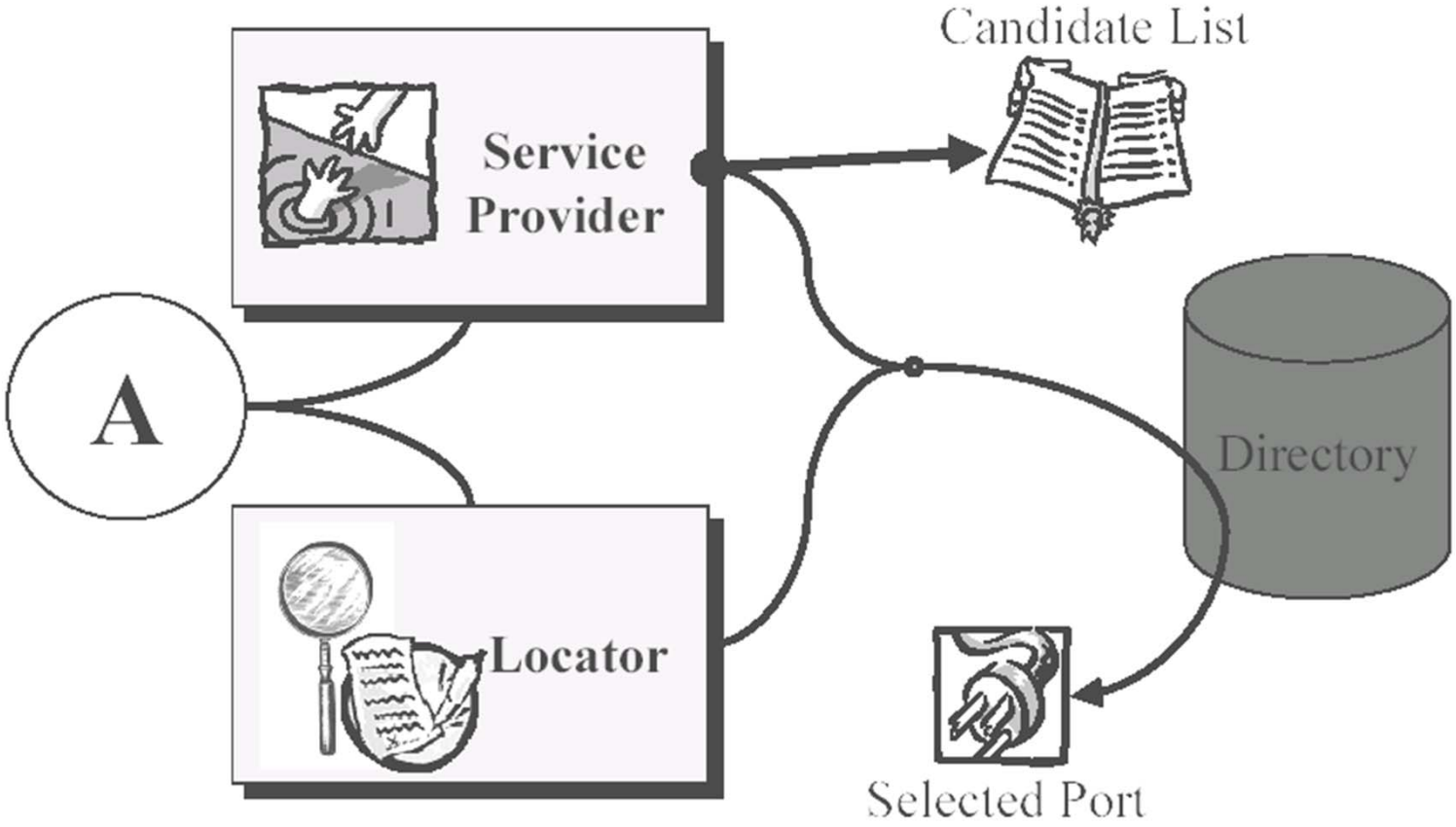
Geschäftseigenschaften einer Aktivität



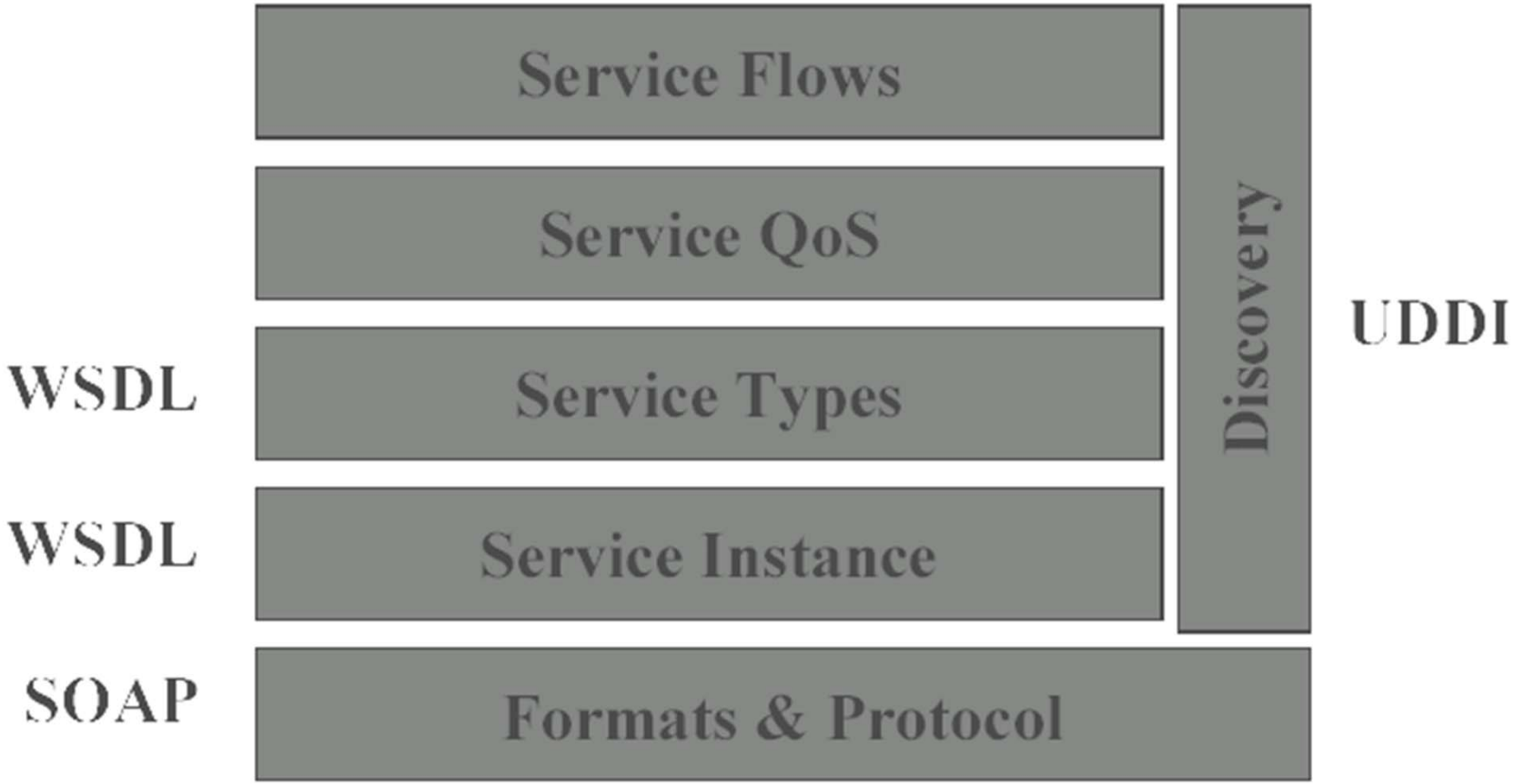
Qualität von Services



Passende Endpunkte

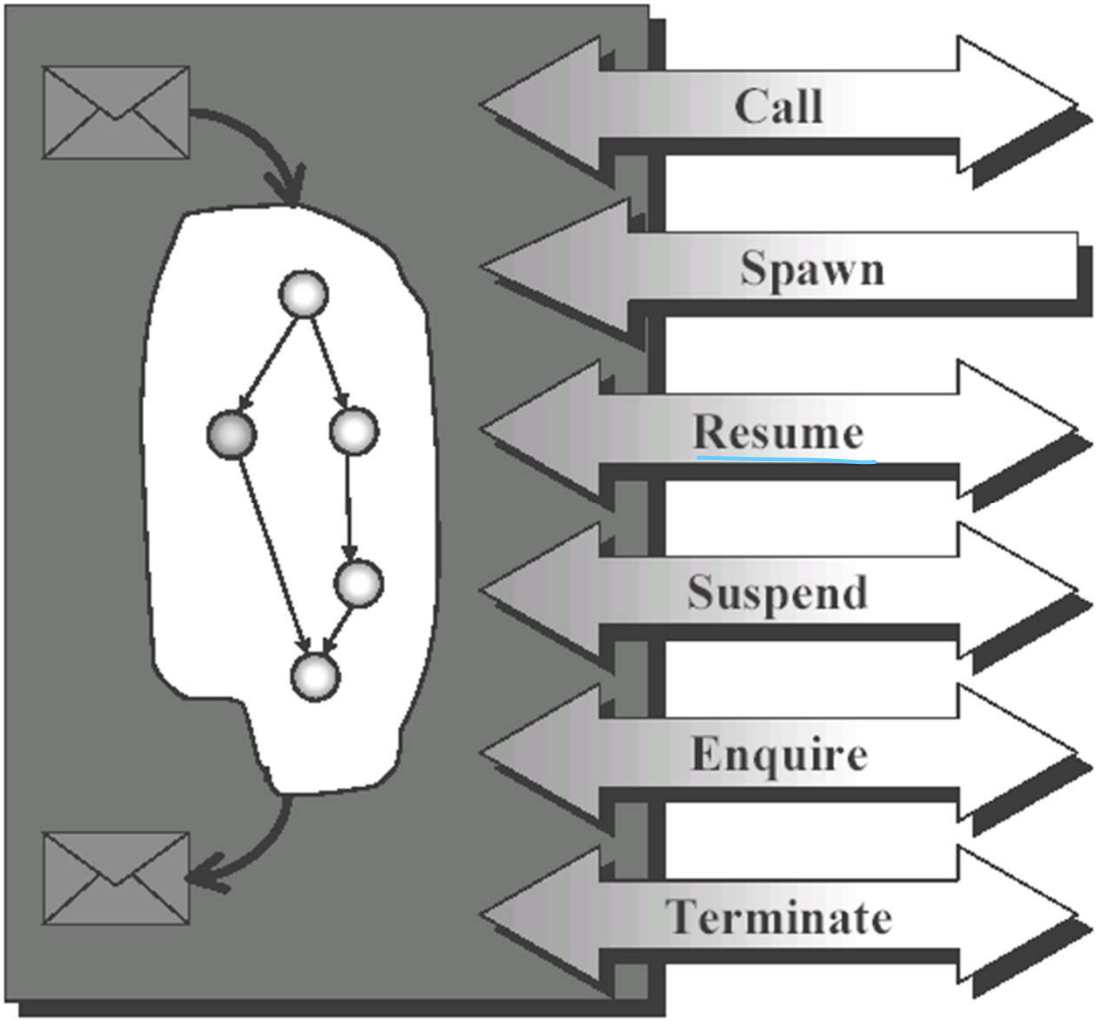


Der Web Services Stack

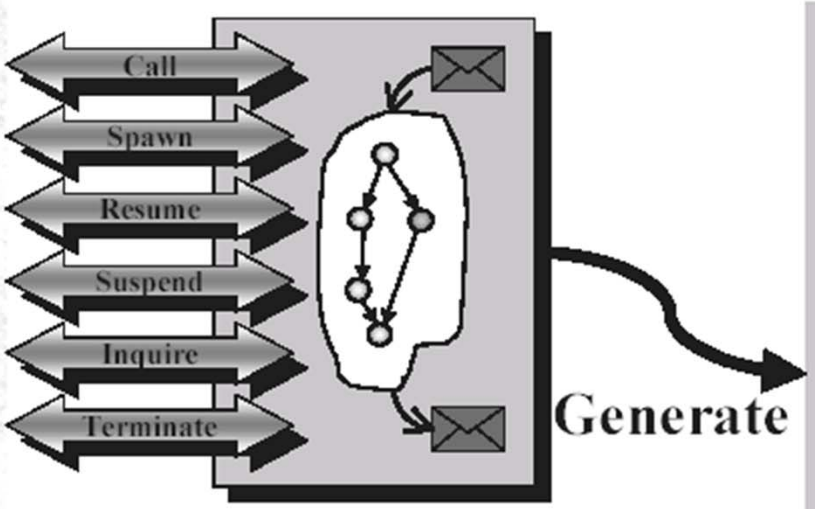


Workflows als Web Services

Workflows als Web Services

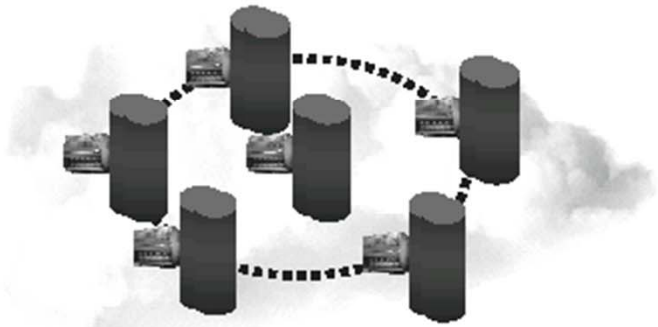


Workflow Lifecycle Services veröffentlichen



Generate

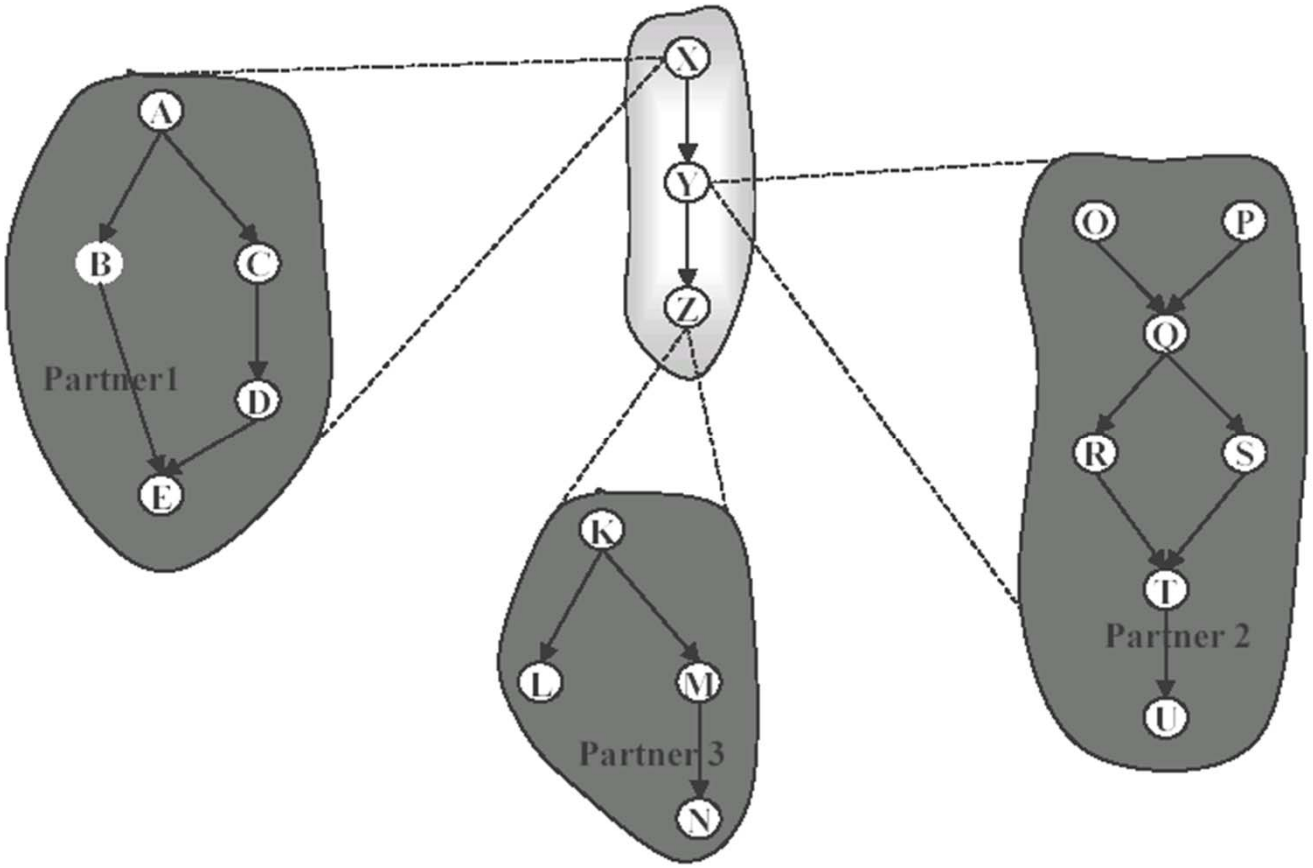
```
...
<message name="MortgageInput"> ...</message>
<message name="MortgageOutput">...</message>
...
 <portType name="MortgageLifeCyclePortType">
      <operation name="Call">
        <input message="tns:MortgageInput"/>
        <output message="tns:InstanceInformation"/>
      </operation>
      <operation name="Suspend">
        <input message="tns:InstanceInformation"/>
        ...
      </operation>
      ...
   </portType>
<binding name="MortgageSOAPBinding"
   type="tns:MortgageLifeCyclePortType">
...
</binding>
<service
name="MortgageLifeCycleService">...</service>
```



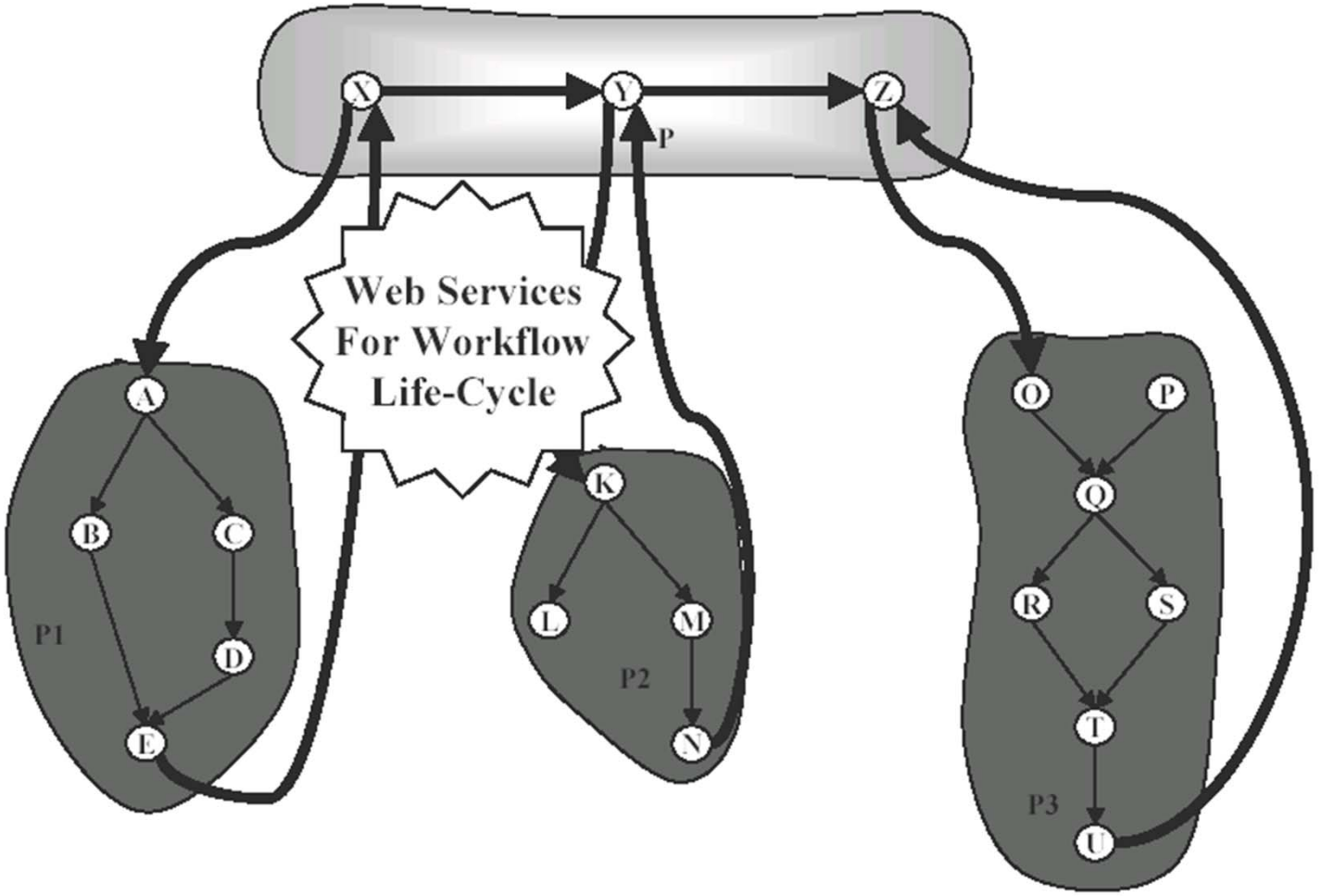
Publish

Web Services Choreographie: hierarchische Abläufe

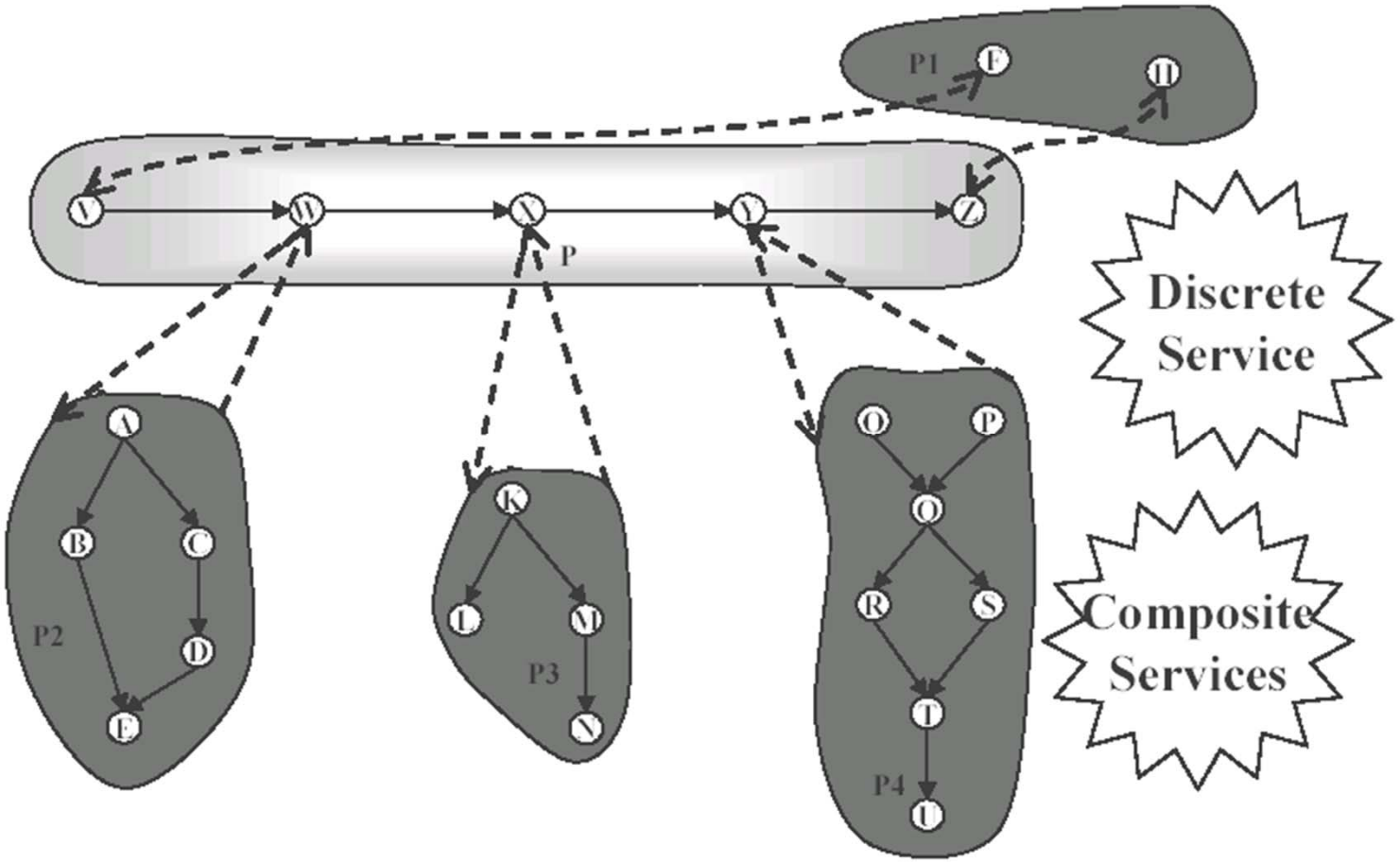
Multi-laterale Vereinbarung



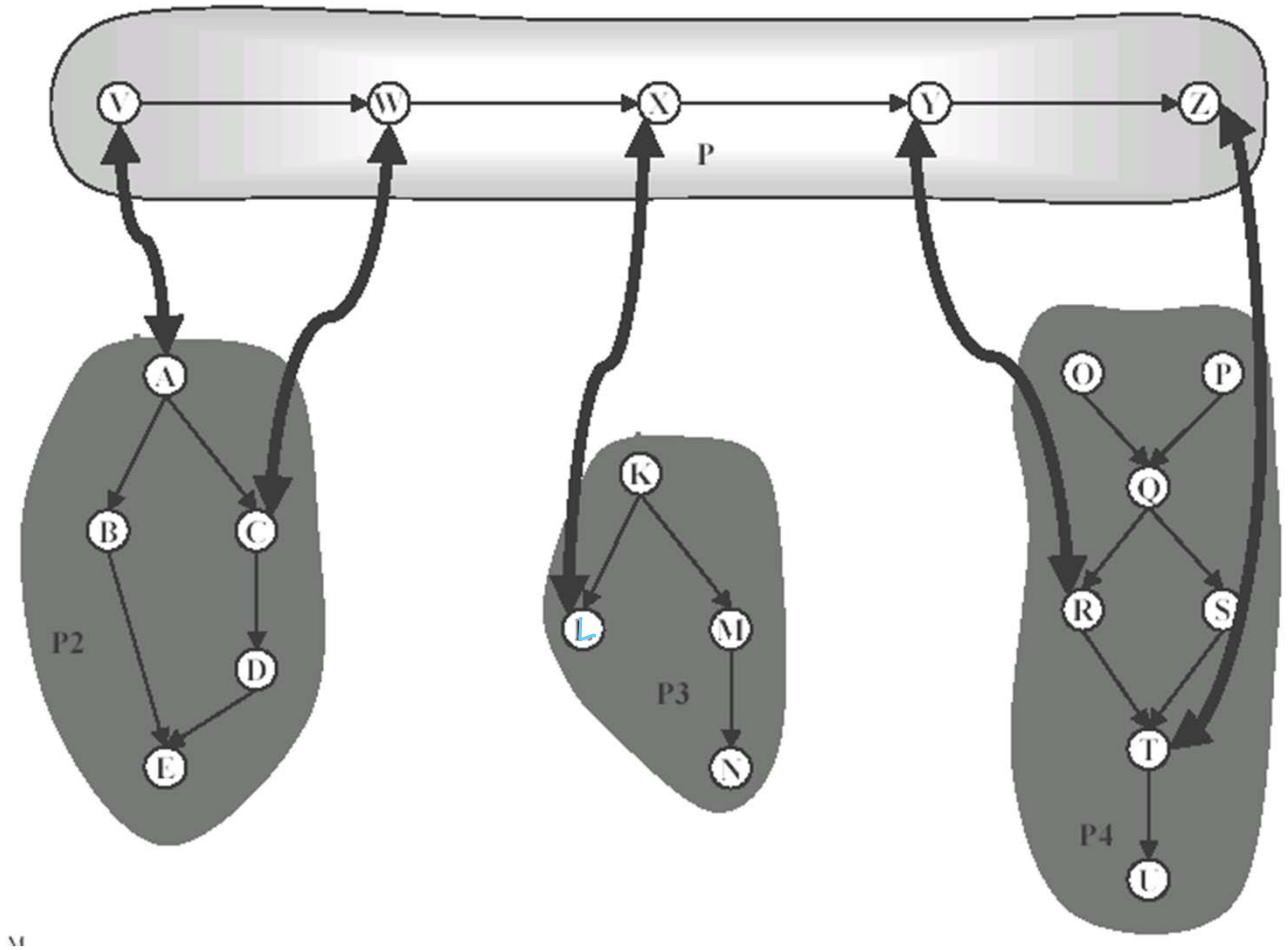
Öffentlicher Workflow



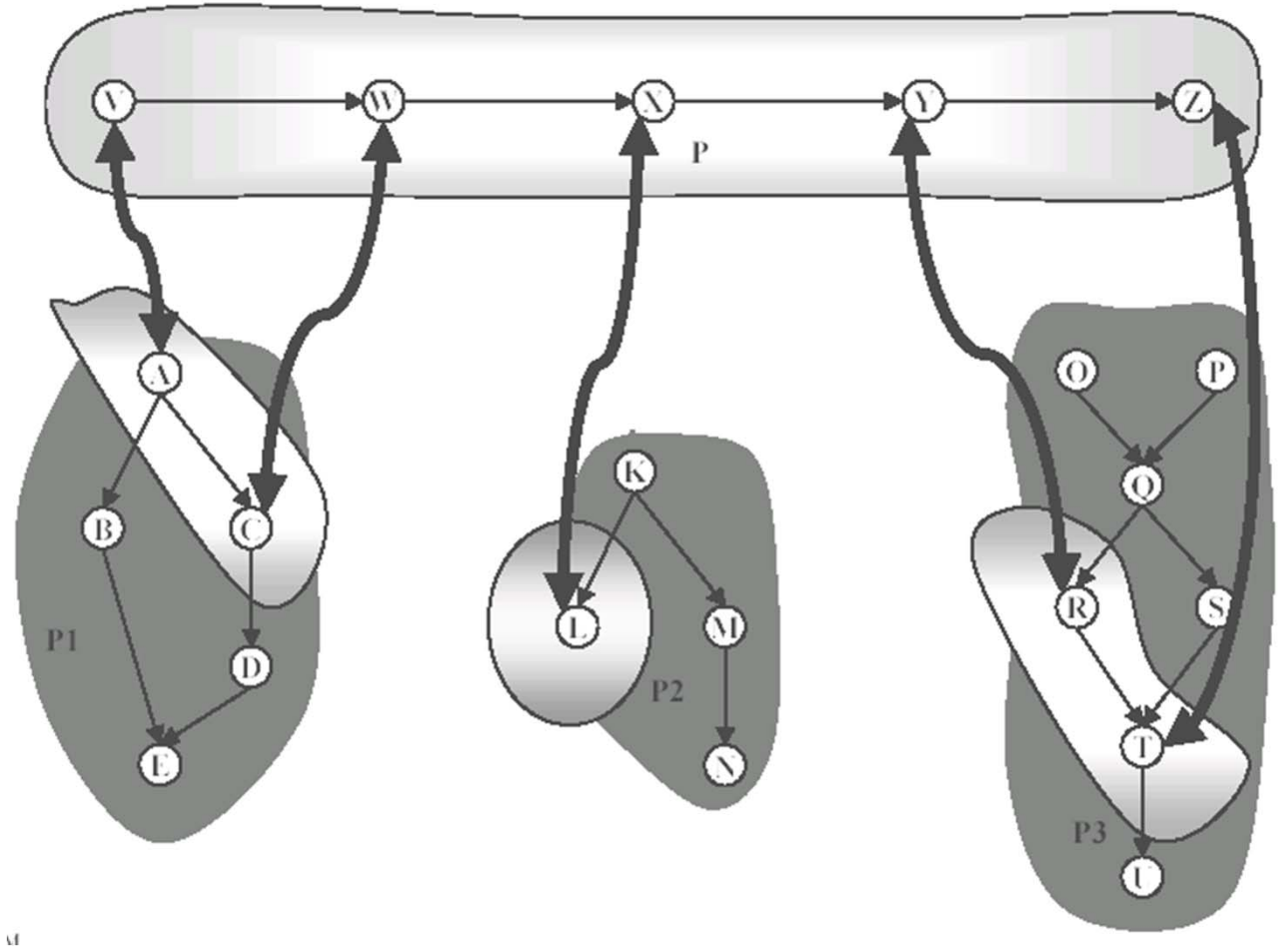
Hierarchisches B2B-Szenario



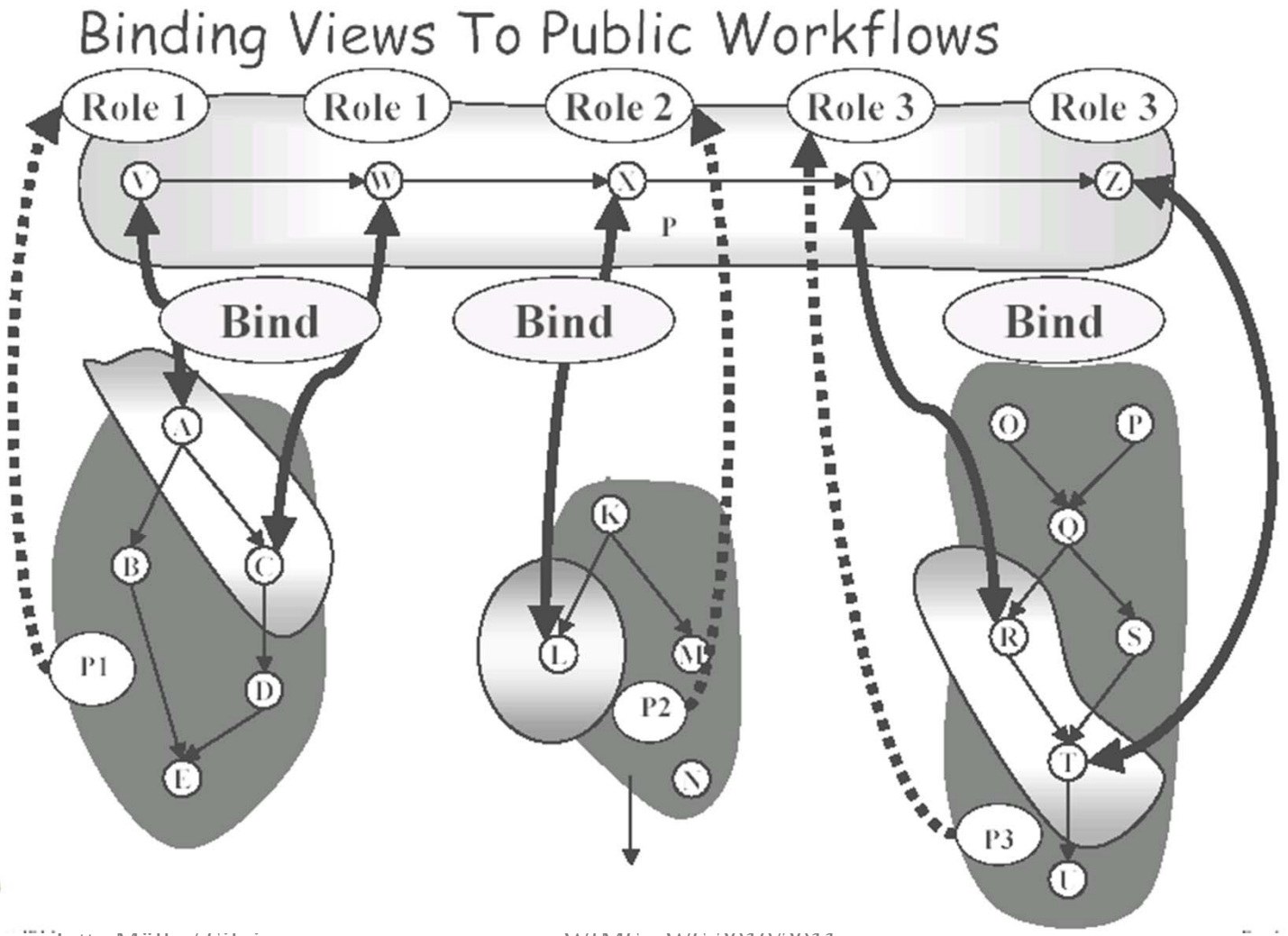
“Einzelne” Teilnahme an öffentlichen Workflows



Öffentliche Sichten auf privaten Workflows

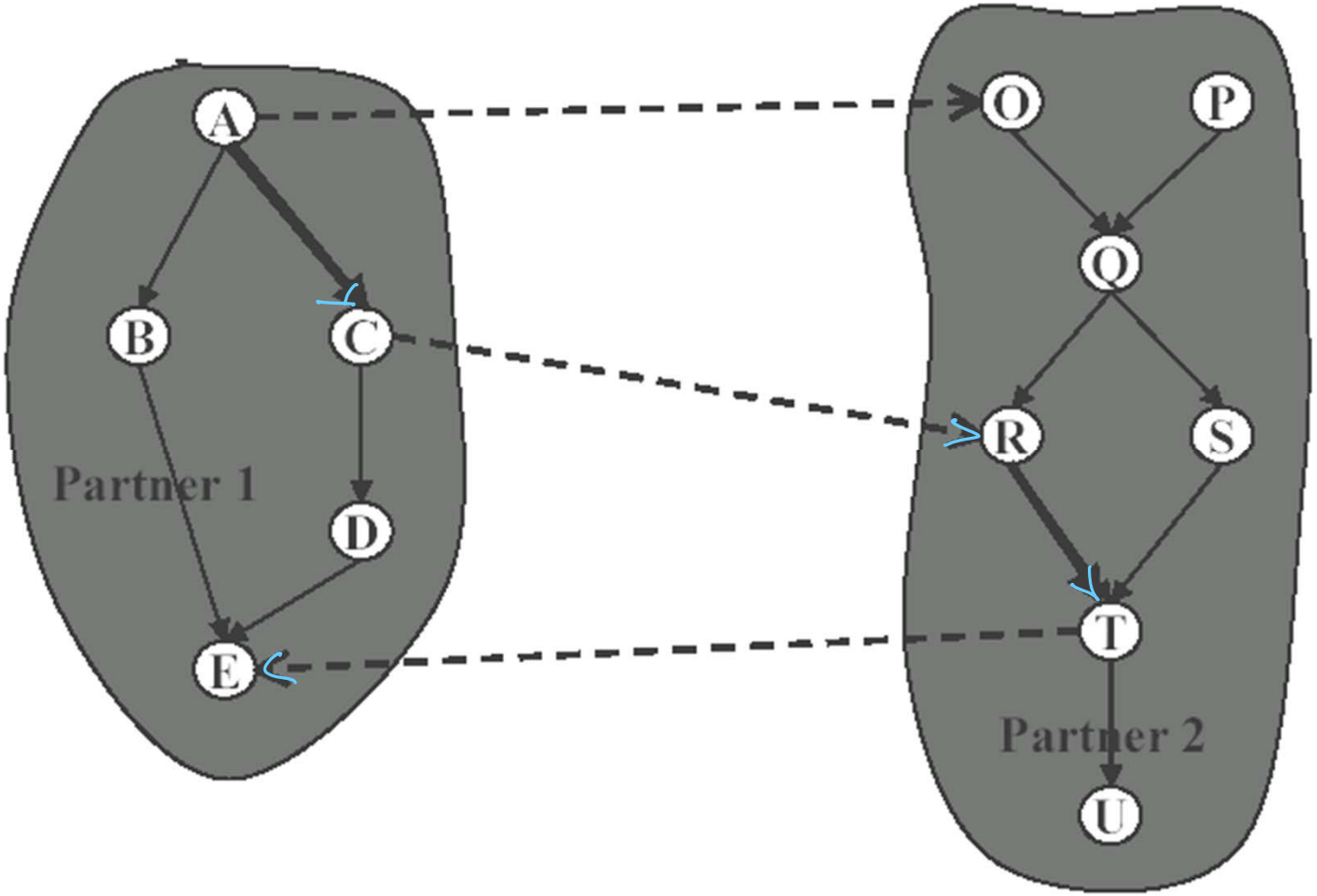


„Binding“ von Sichten an öffentliche Workflows

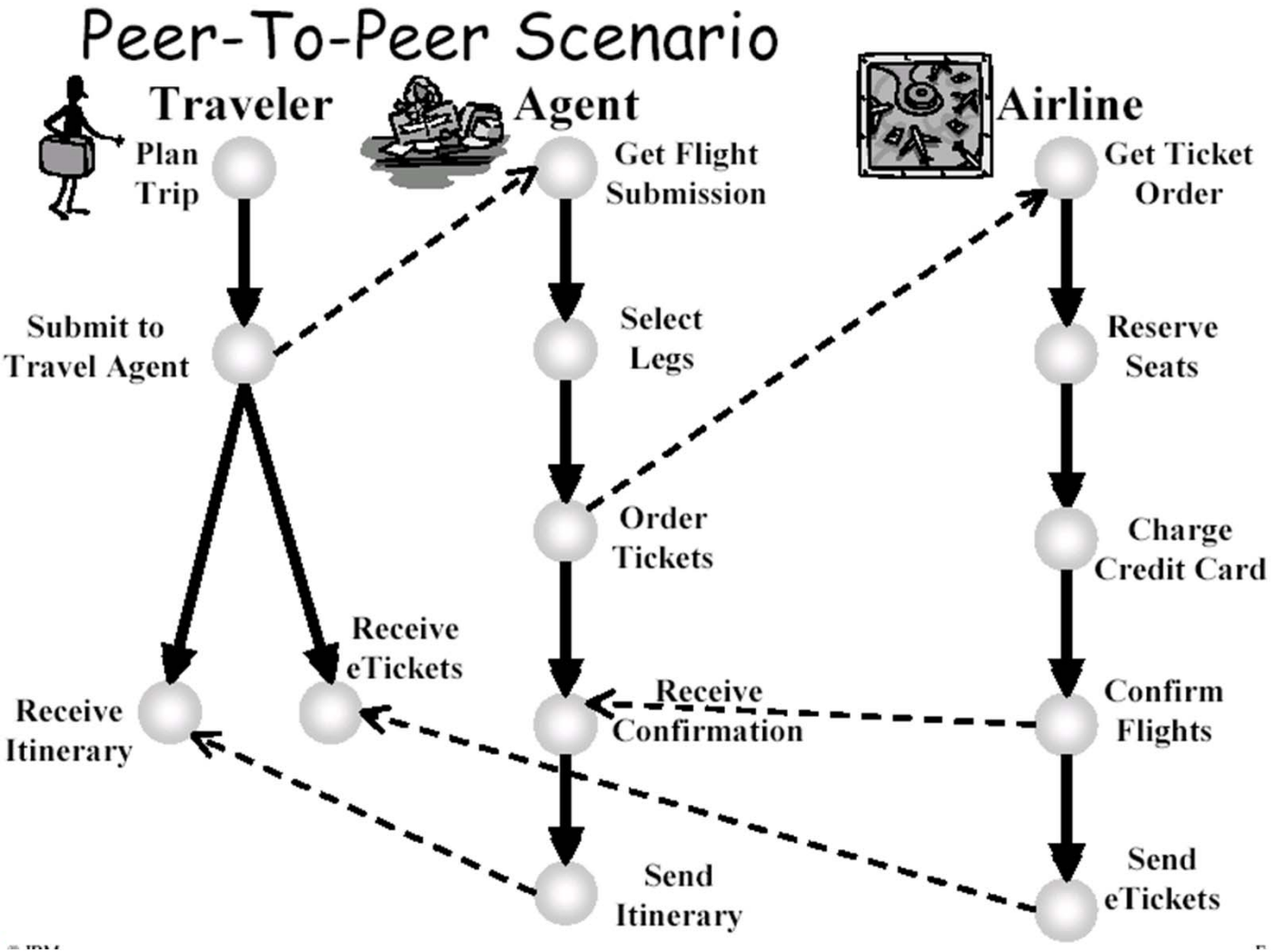


Web Services Choreographie: Peer-To-Peer Abläufe

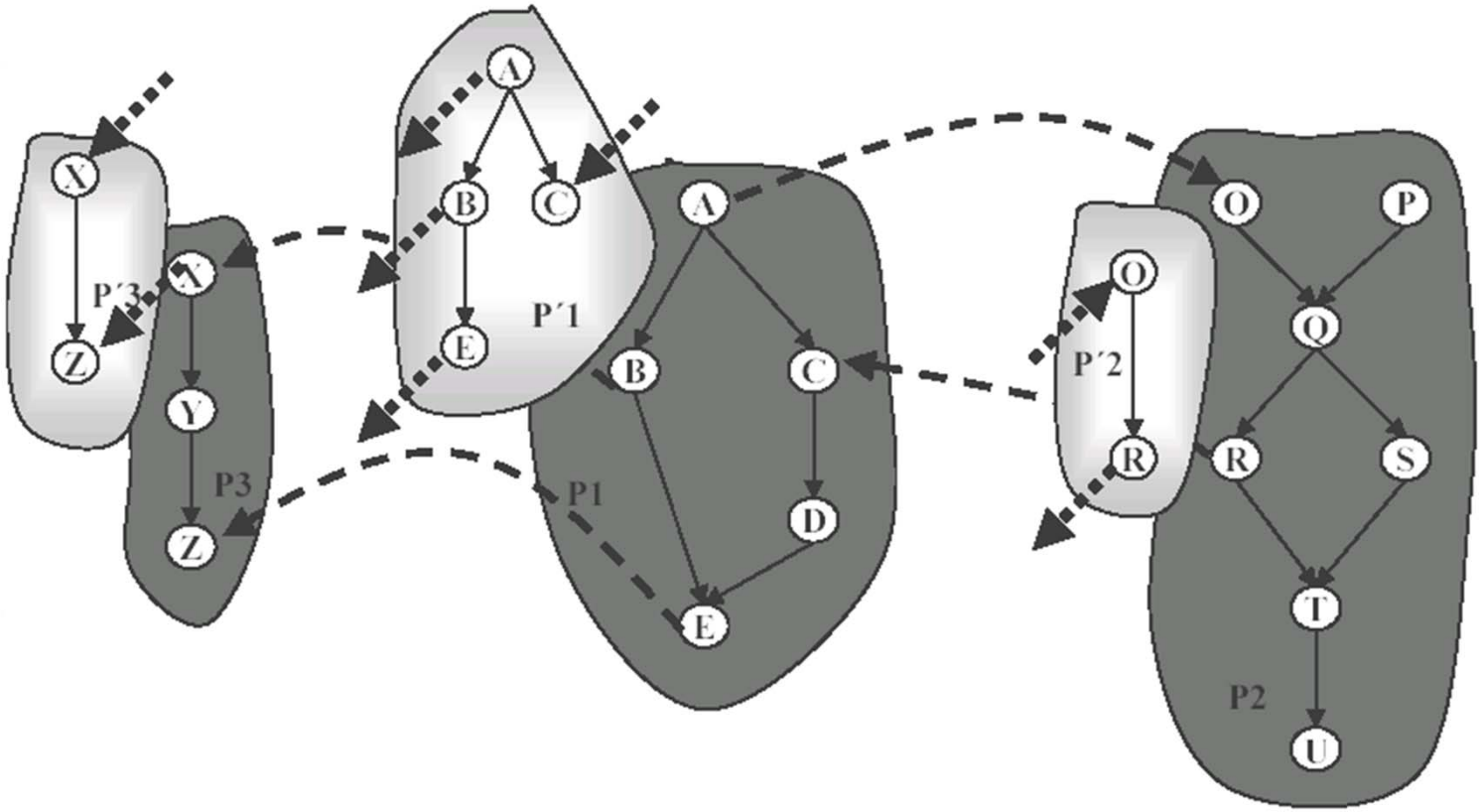
Peer-To-Peer B2B-Szenario



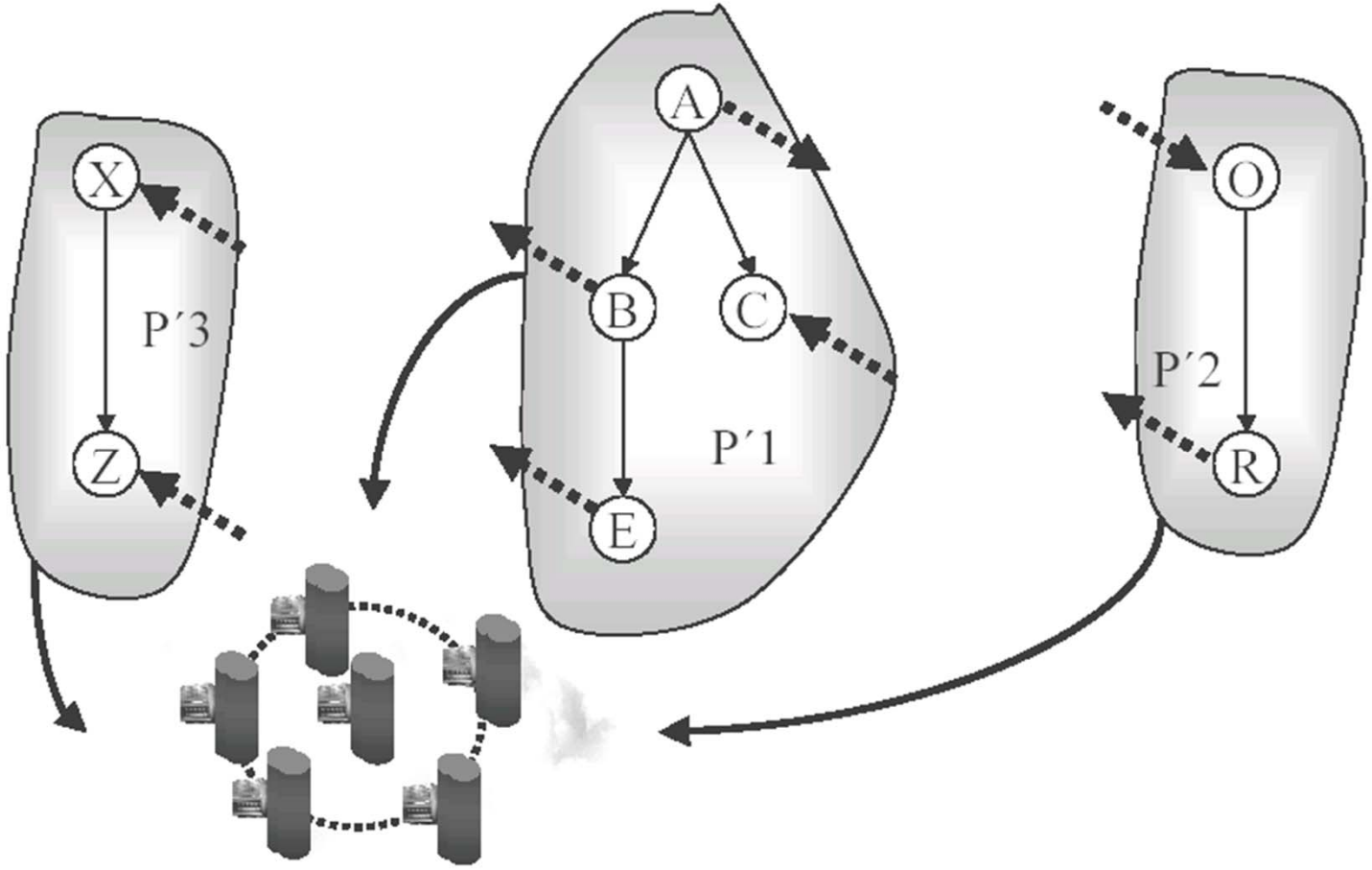
Peer-To-Peer Szenario



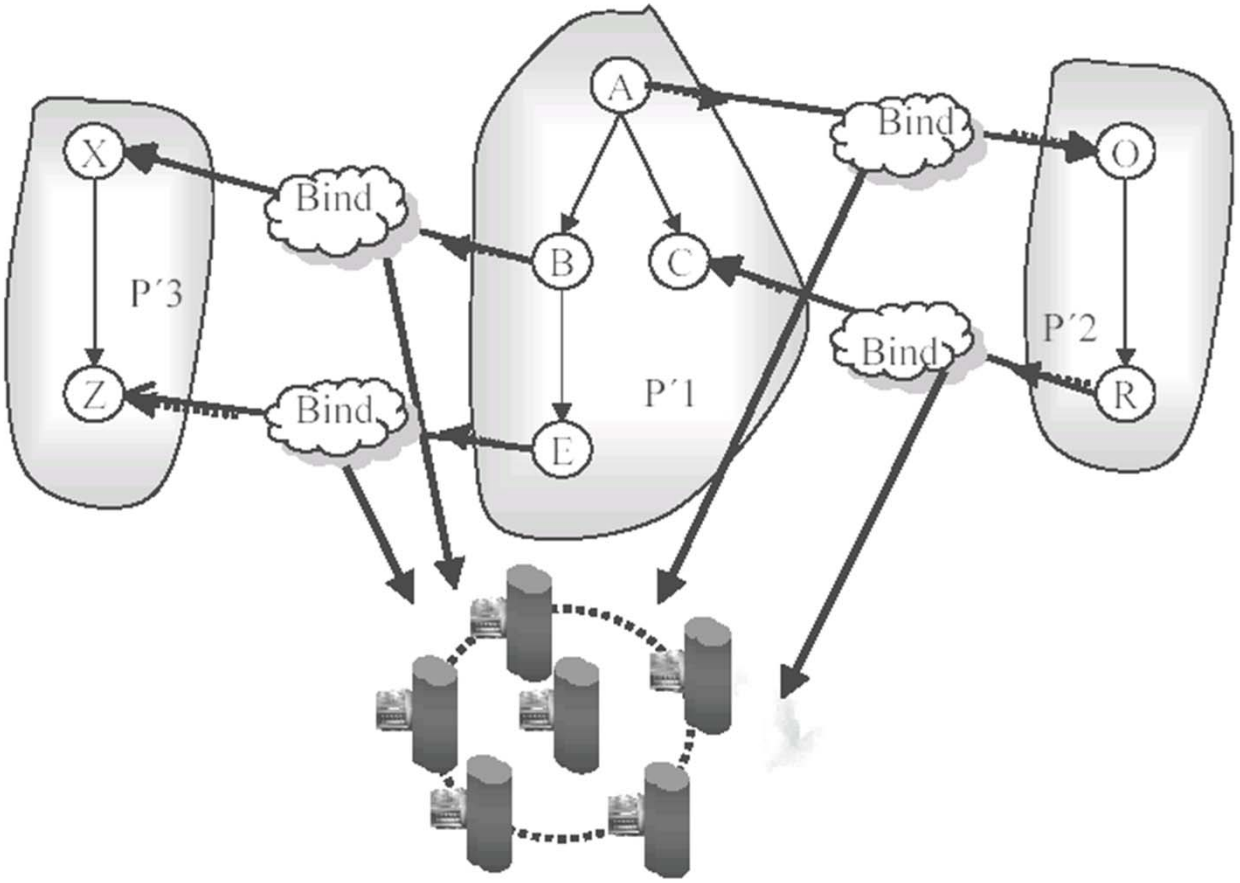
Sichtendefinition auf Workflows...



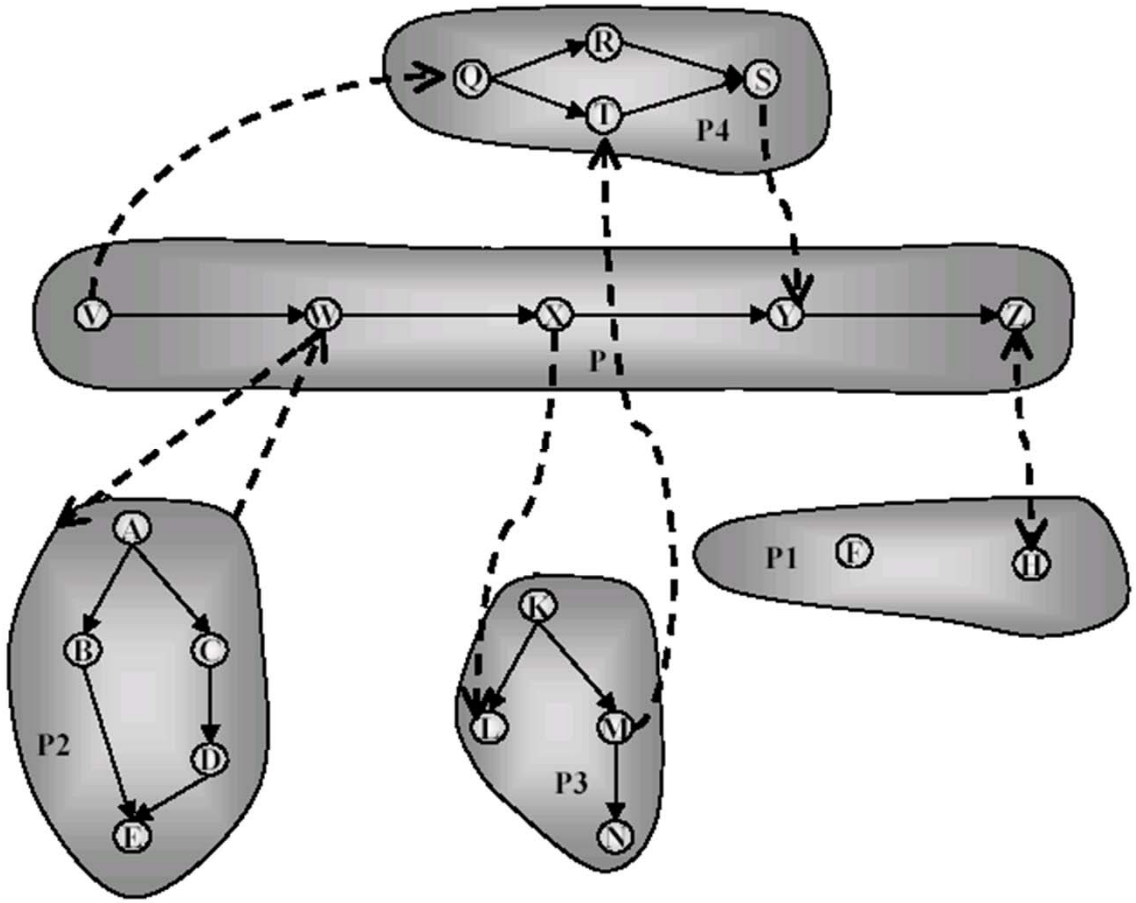
...und veröffentlichen der Sichten als Web Services



Dynamisches „Binding“zur Laufzeit

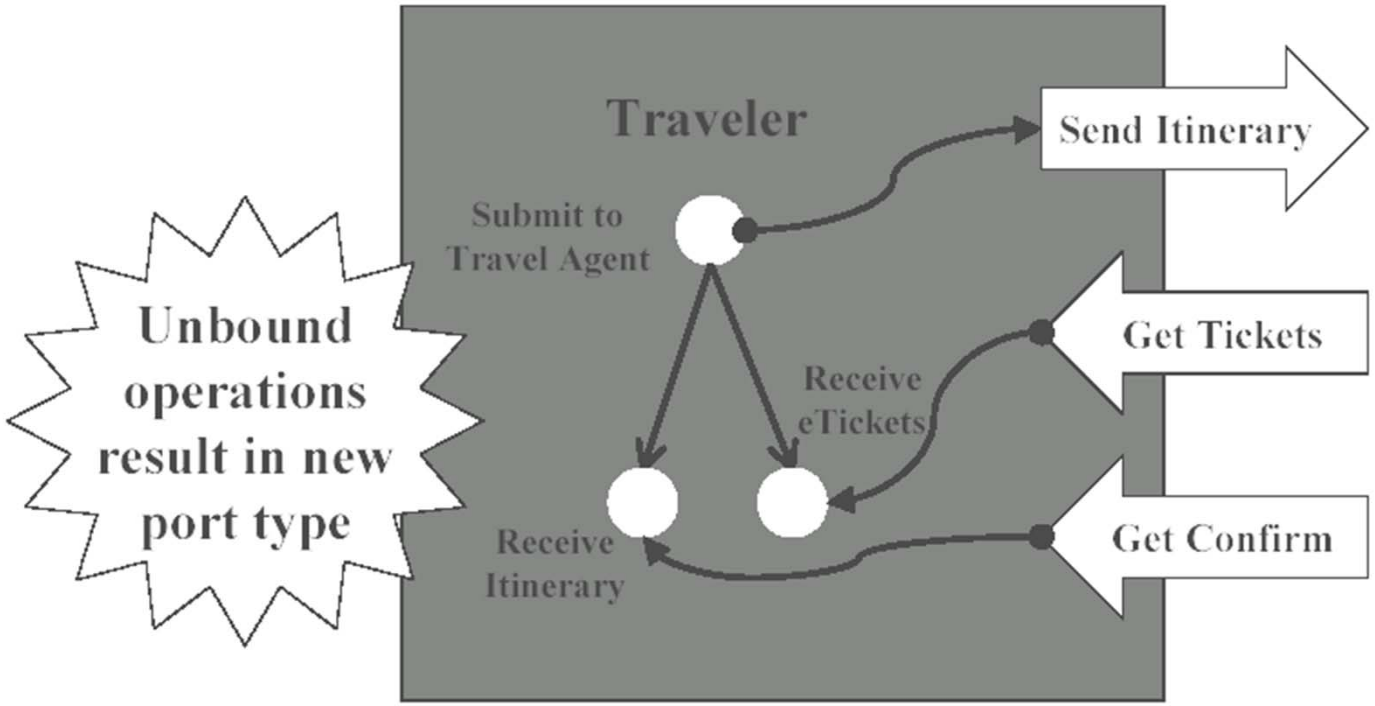


und schließlich: ein gemischtes Szenario

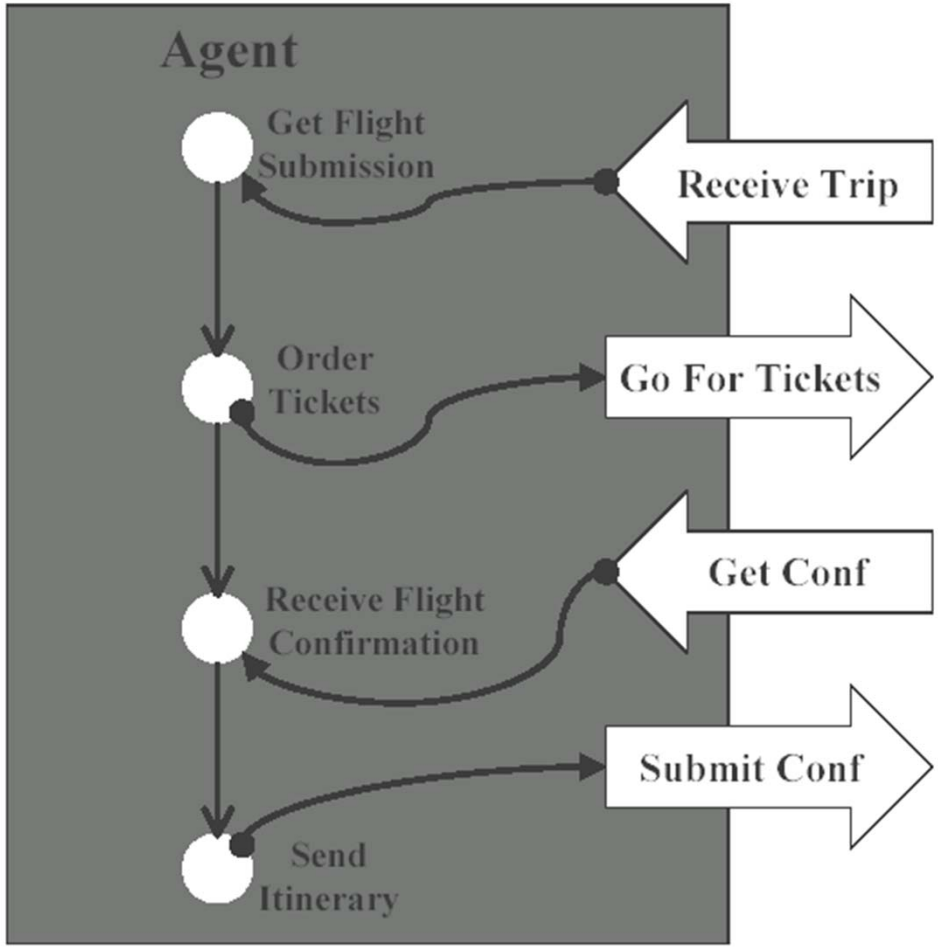


Web Services Choreographie: Aggregierung von Web Services

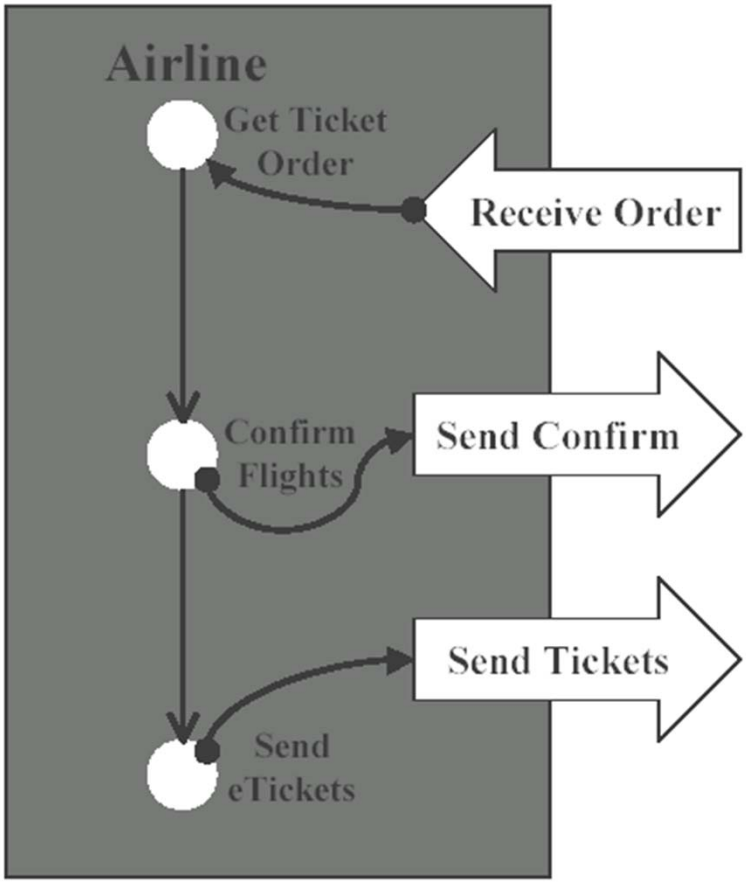
Reiseablauf als zusammengesetzter Service



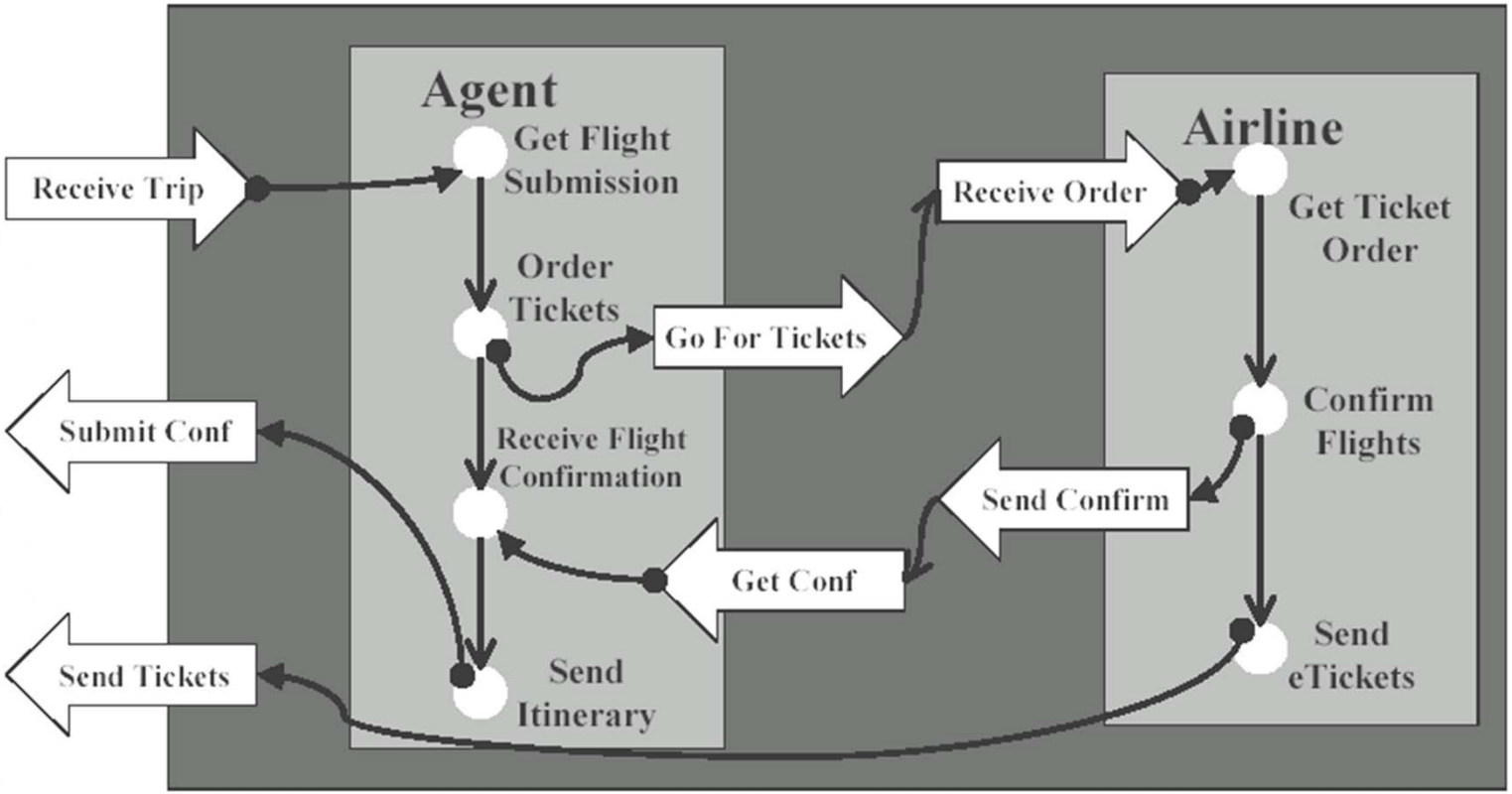
Agent als zusammengesetzter Service



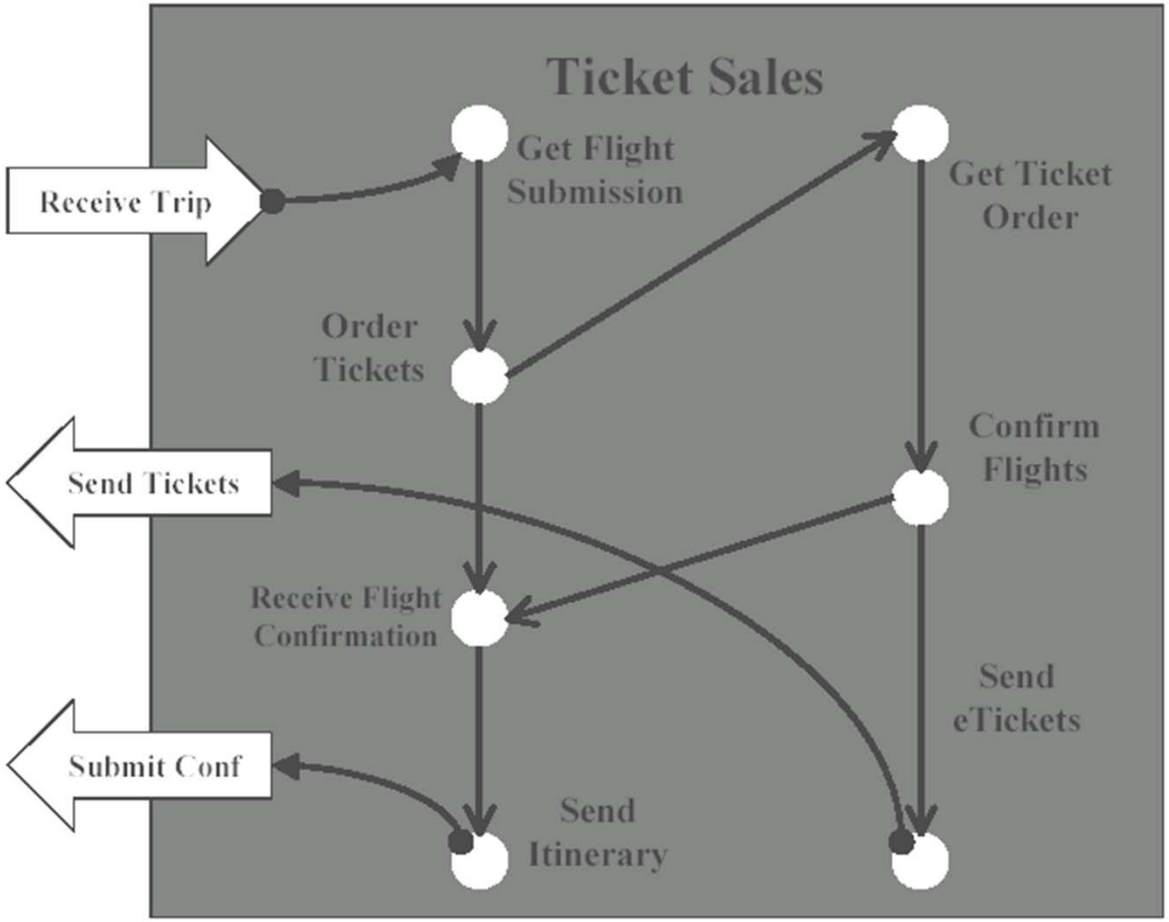
Fluggesellschaft als zusammengesetzter Service

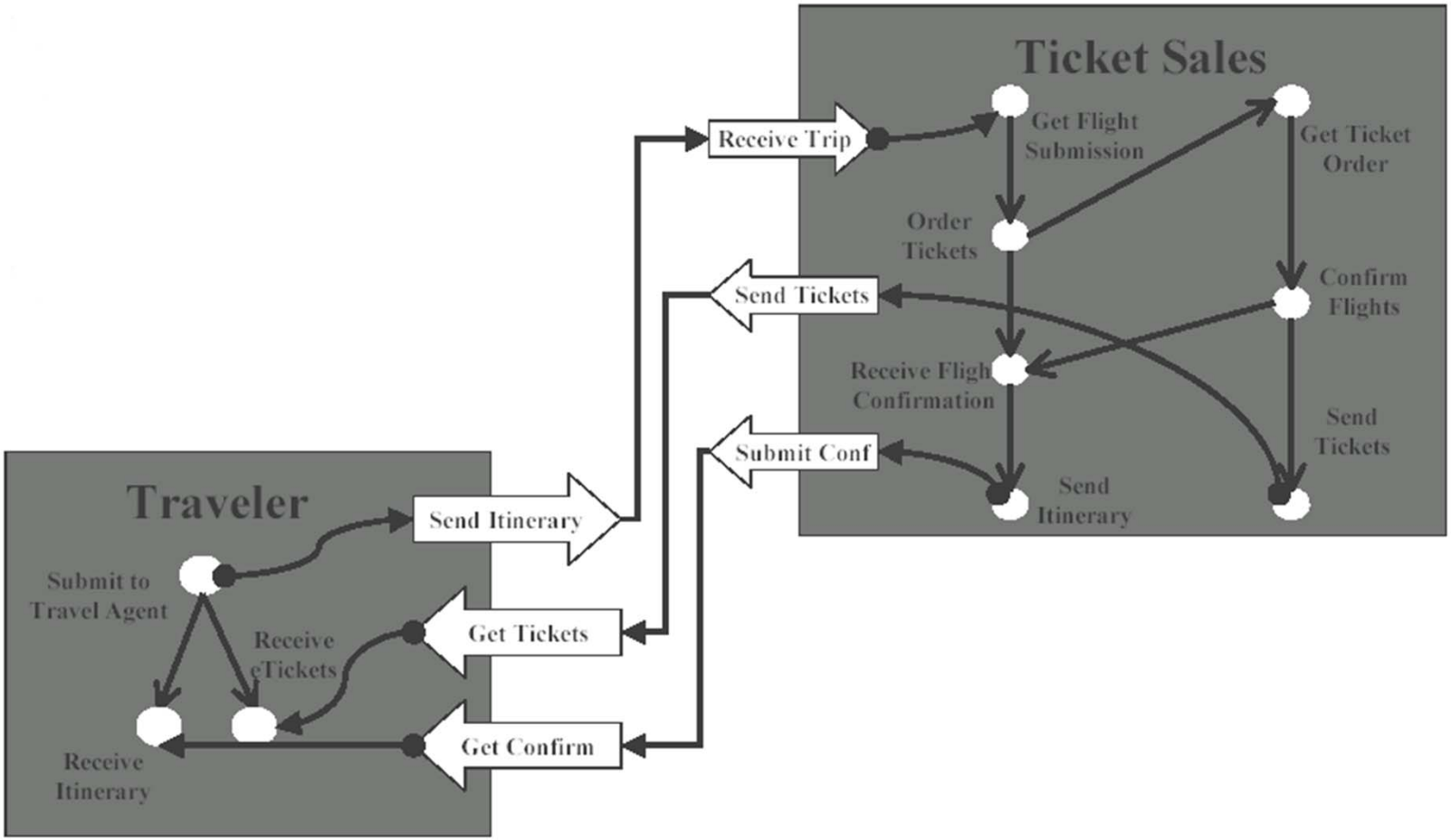


rekursive Komposition...

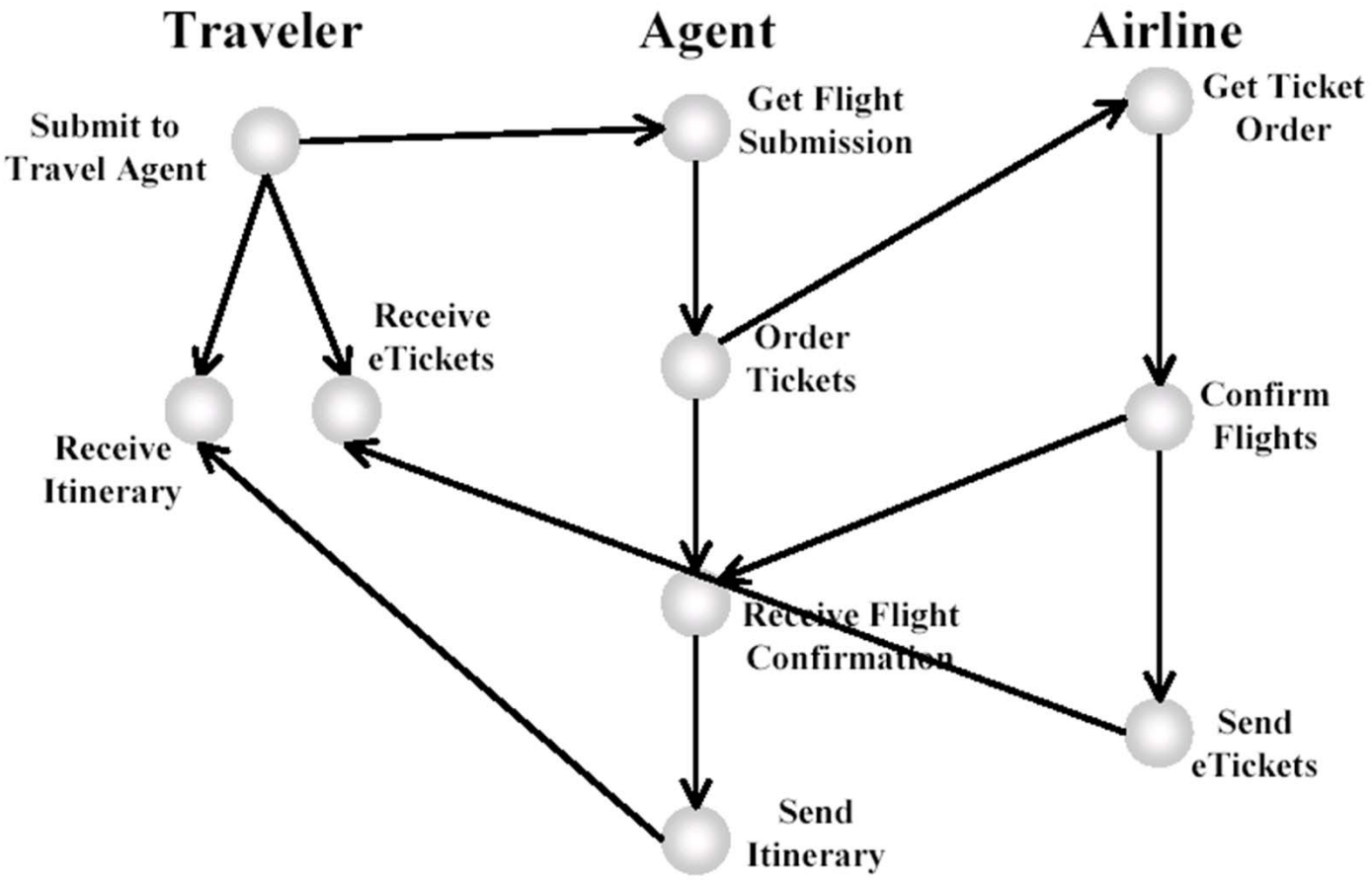


...und ist die Sicht von außerhalb



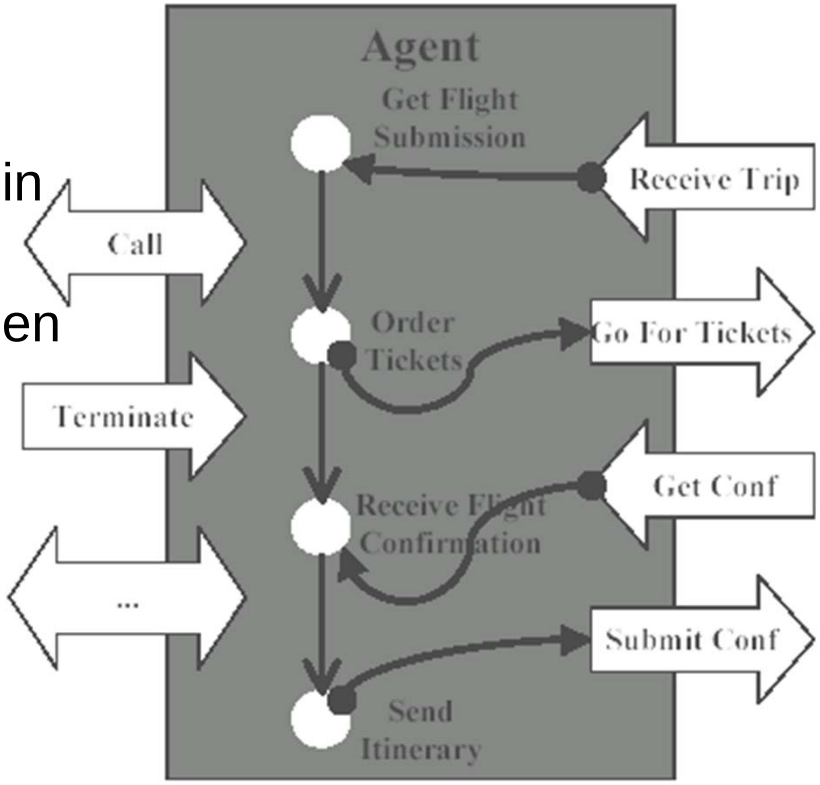

Ticket Sales
Receive Trip
Get Flight Submission
Get Ticket Order
Order Tickets
Send Tickets
Confirm Flights
Receive Fligh Confirmation
Submit Conf
Send Tickets
Send Itinerary
Traveler
Send Itinerary
Submit to Travel Agent
Get Tickets
Receive eTickets
Get Confirm
Receive Itinerary

...das kennt man schon!



Vervollständigung des Bildes

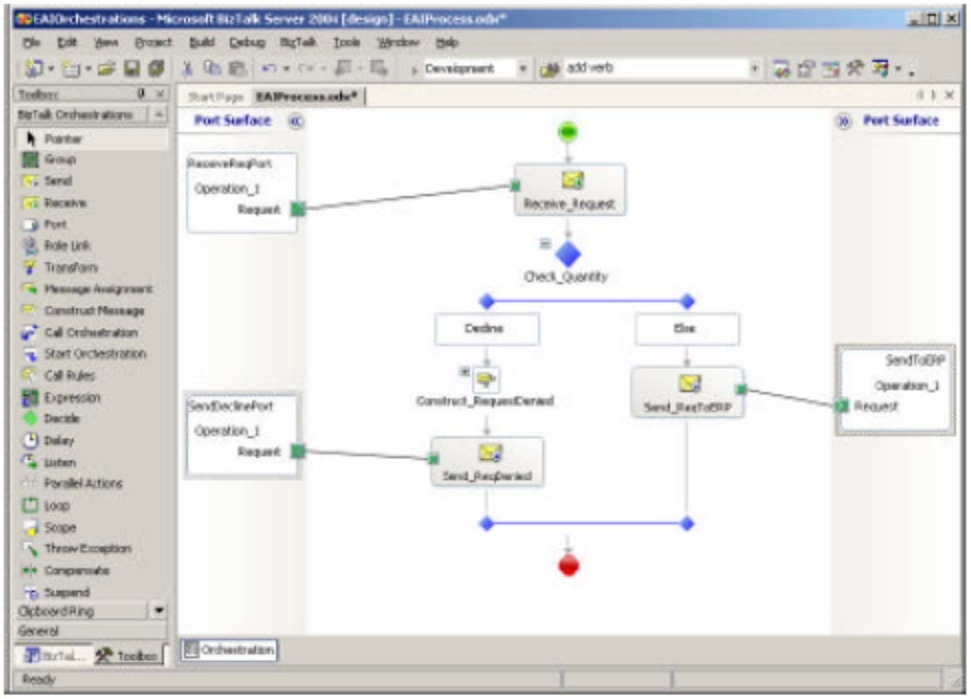
- Ein Web Service Ablauf sollte zwei Arten von Operationen anbieten
  - die exportierten offenen („dangling“) Operationen, die ein Binding erfordern
  - die Lifecycle-Operationen für den Ablauf als Ganzes



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)

Orchestrierung mit Microsoft Biztalk



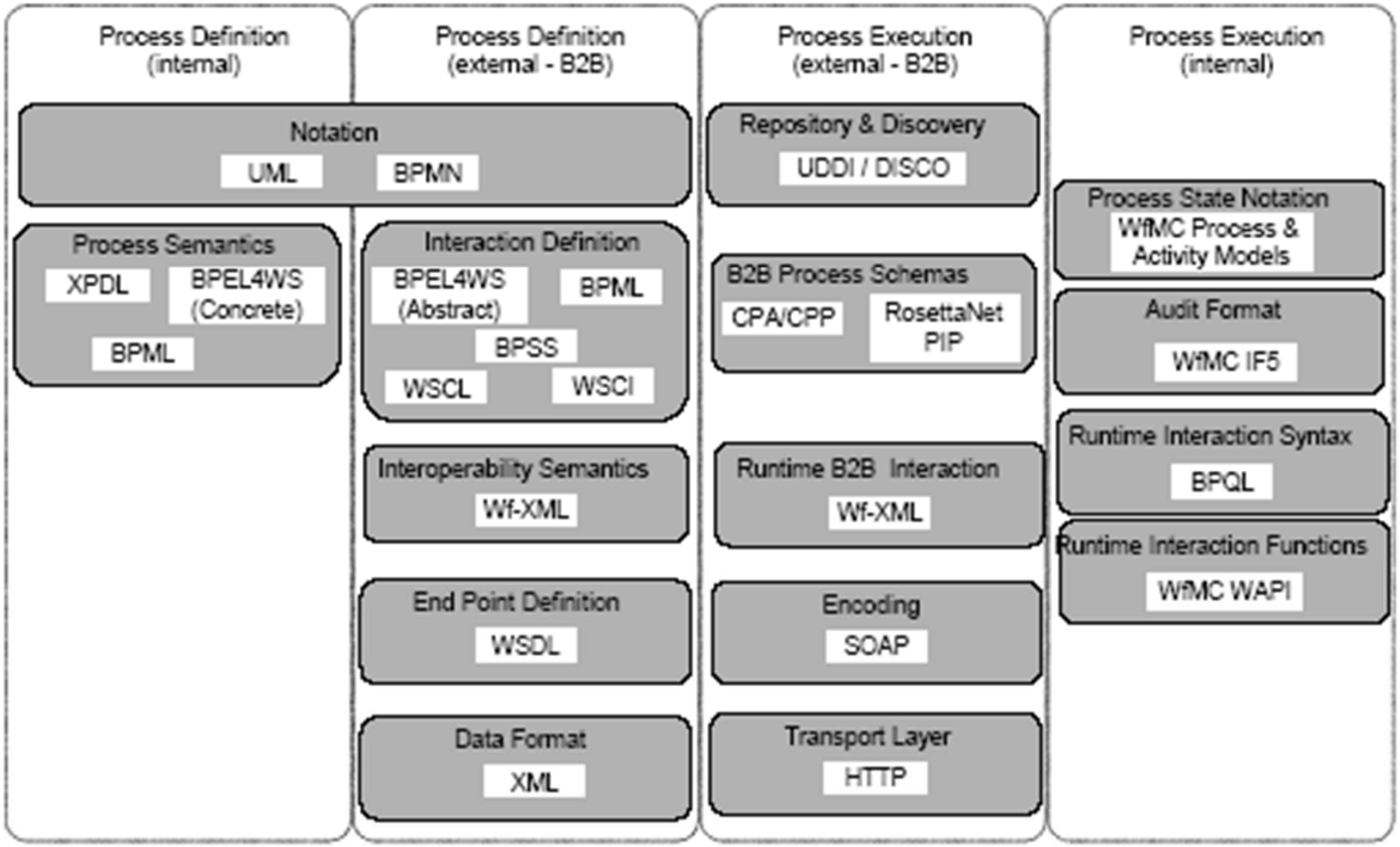
Exemplarische Fragen – Kapitel 6 (1)

- Was versteht man unter Web Services?
- Was sind die Grundkomponten 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‘.

Designelemente

- Es gibt Bullets
  - Auf verschiedenen
    - Ebenen
    - Wie man
      - Sehen kann
        - Ist das sehr
        - Fein gegliedert :.)
  - Und man kann
  - Auch relativ viel Text
  - Schreiben
- Denn die Schrift ist nicht allzu gross
- Aber noch leserlich

Standards im Workflow-Bereich (WFMC)