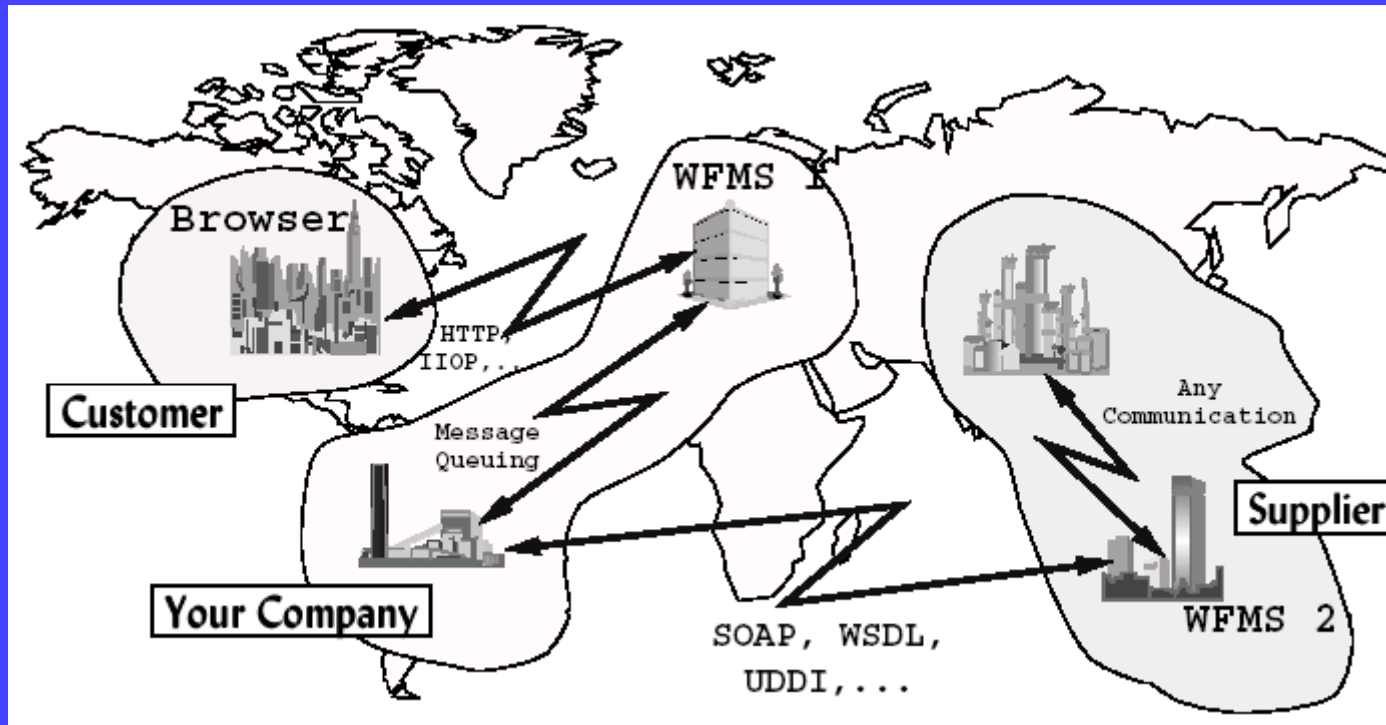


Kapitel 6: Workflows im Internet



Virtual Enterprise: Scenario

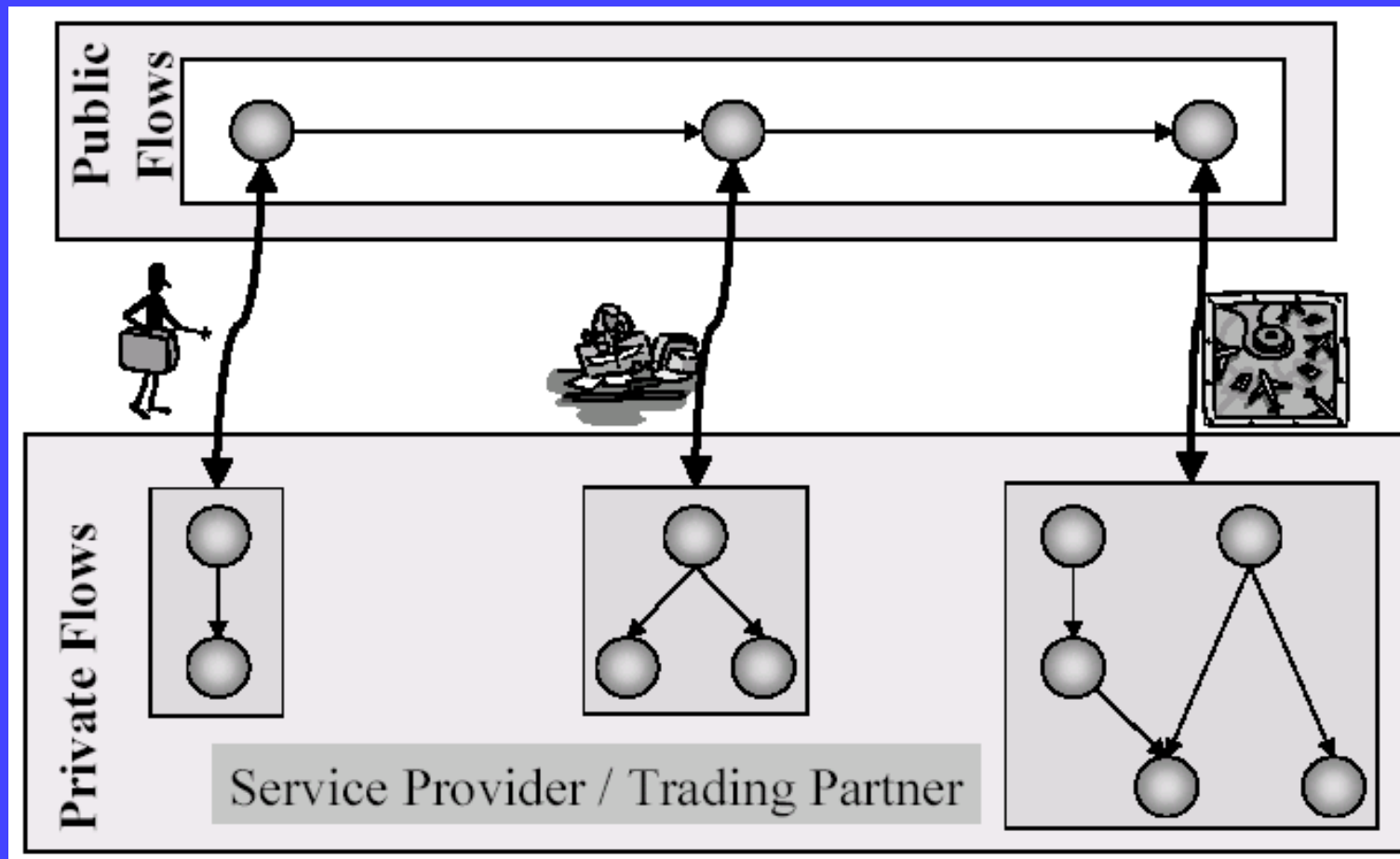


Workflows zwischen Partnern

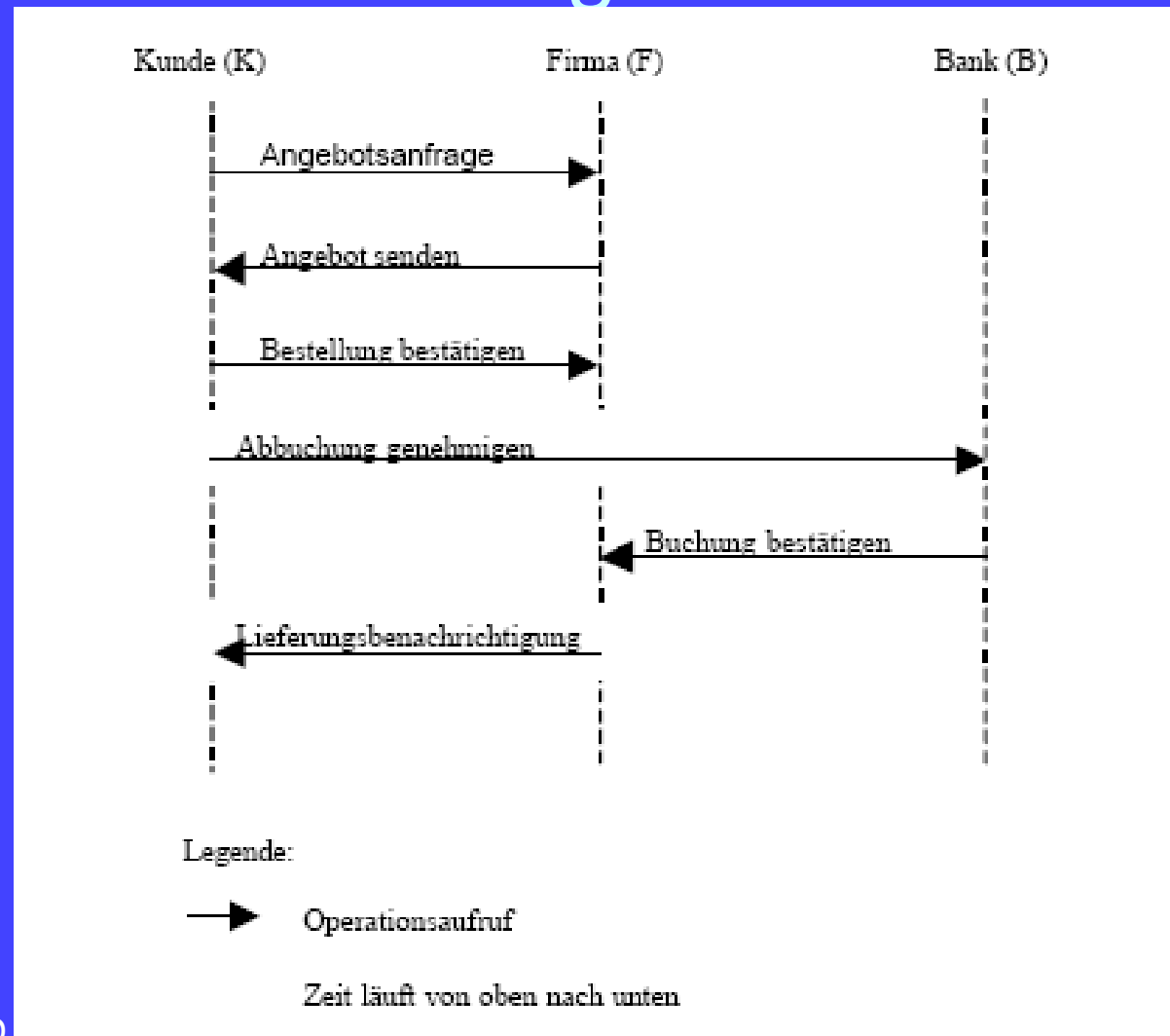
- ◆ Es ist aufwändig, manuell die Verarbeitung bei verschiedenen Partnern zu überwachen und zu protokollieren und auf Ausnahmen zu reagieren.
- ◆ Workflow-Technologie kann helfen!
 - ◆ Modelliere reguläre Partner-Interaktionen
 - ◆ „Conversations“ sind unterschiedlich zum nicht verteilten Fall (!!!)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



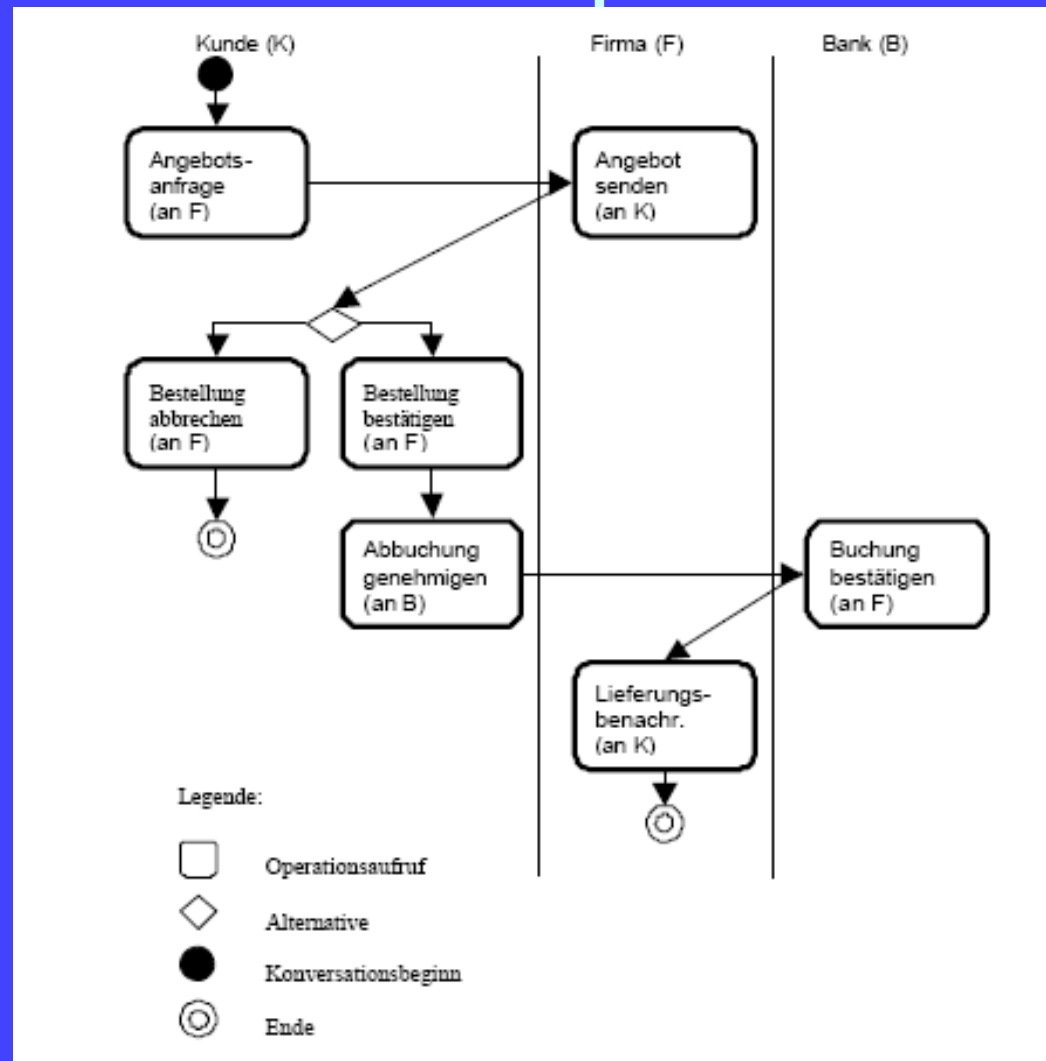
Workflow-Typen



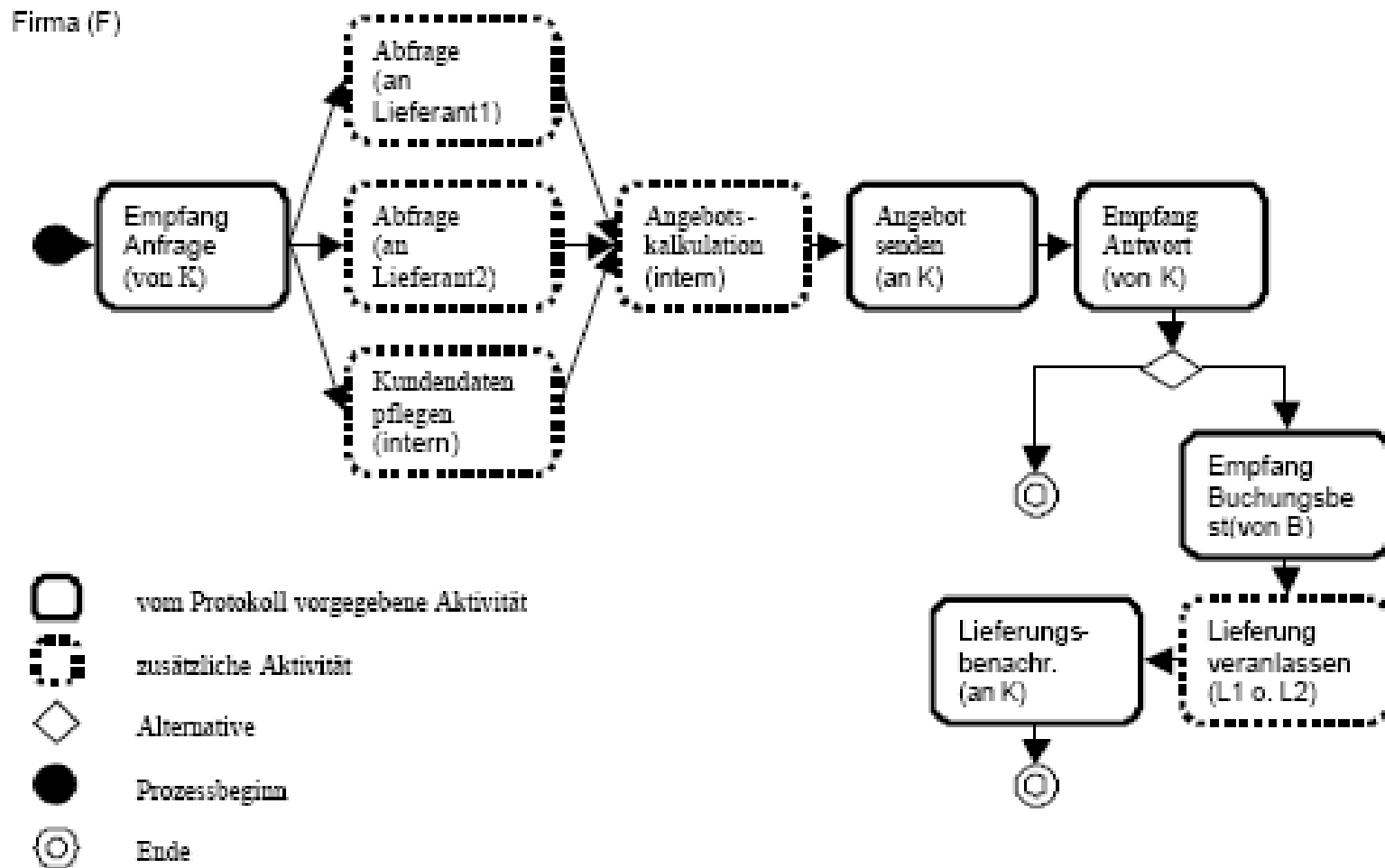
Konversation – Beispiel: Nachrichtenabfolge mit 3 Partnern



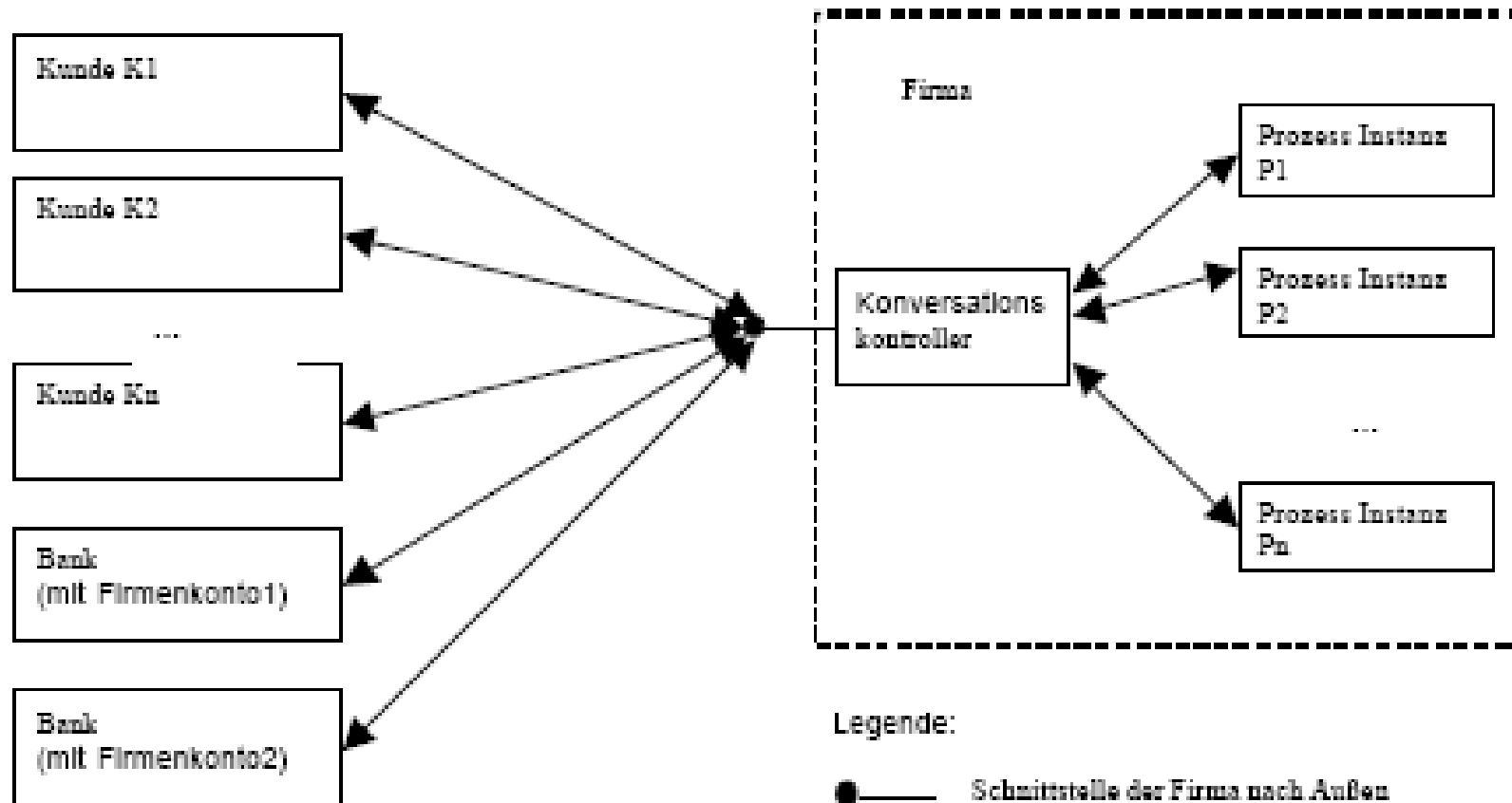
Choreographie – Beispiel Koordinationsprotokoll



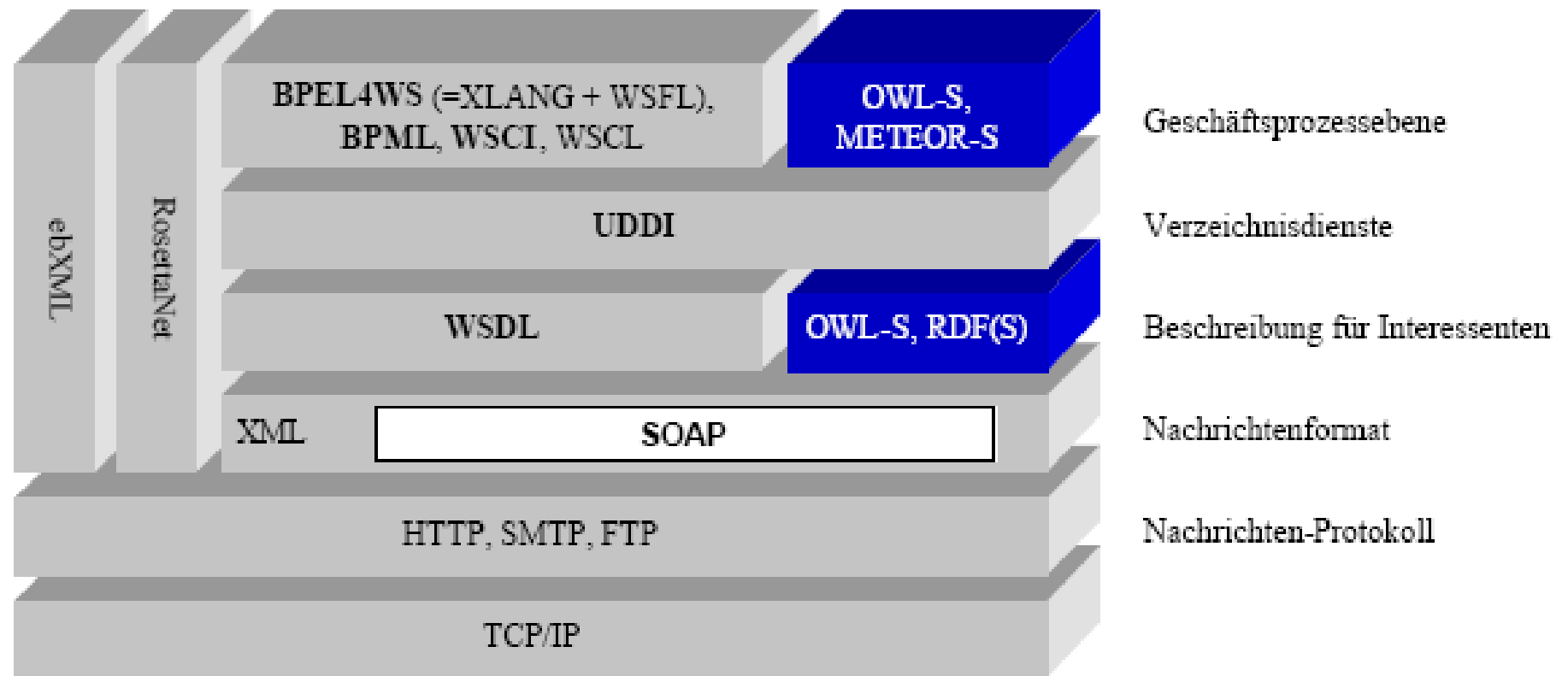
Orchestrierung von Web Services



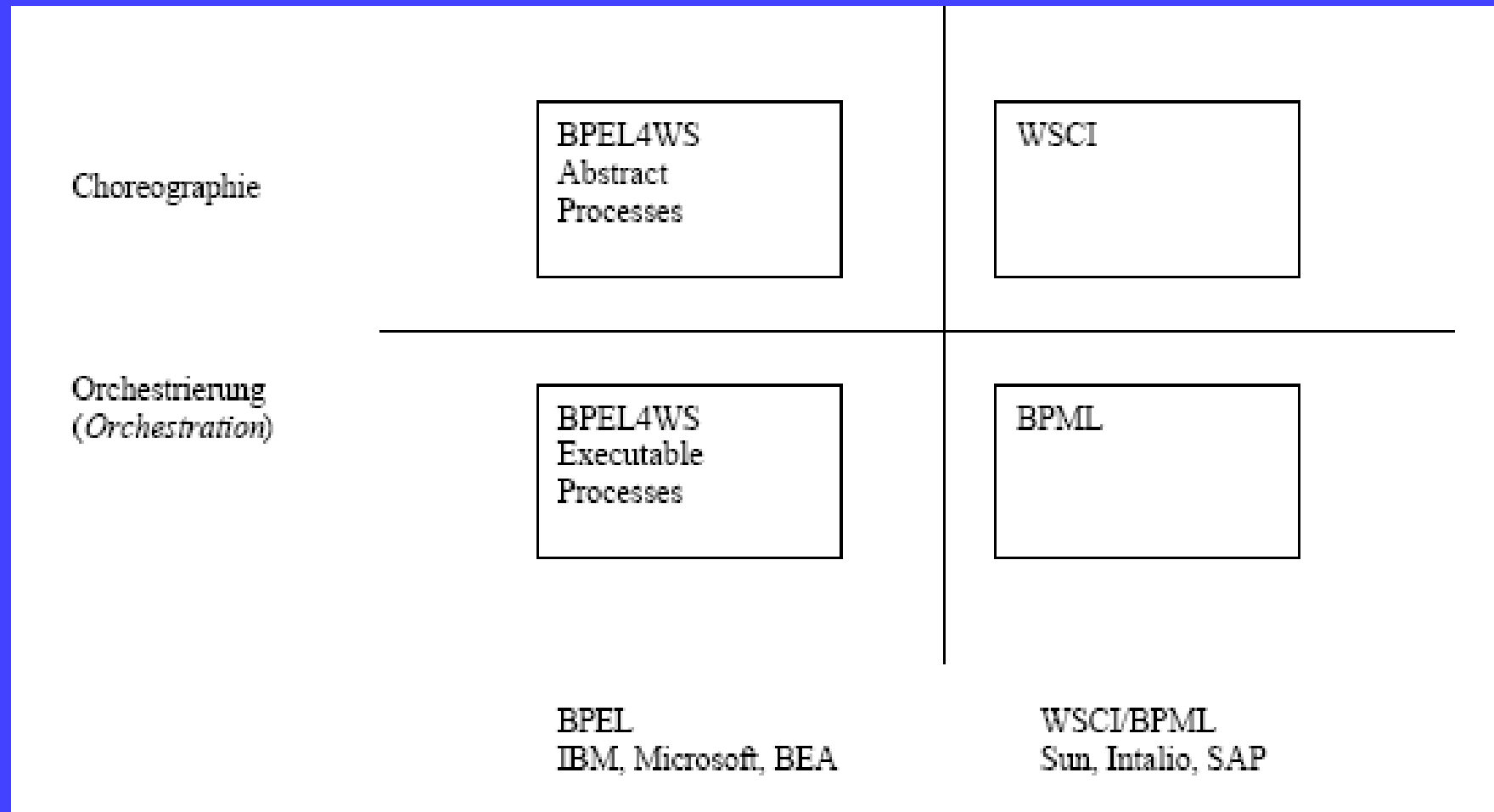
Konversationsunterstützung



Web Service Stack

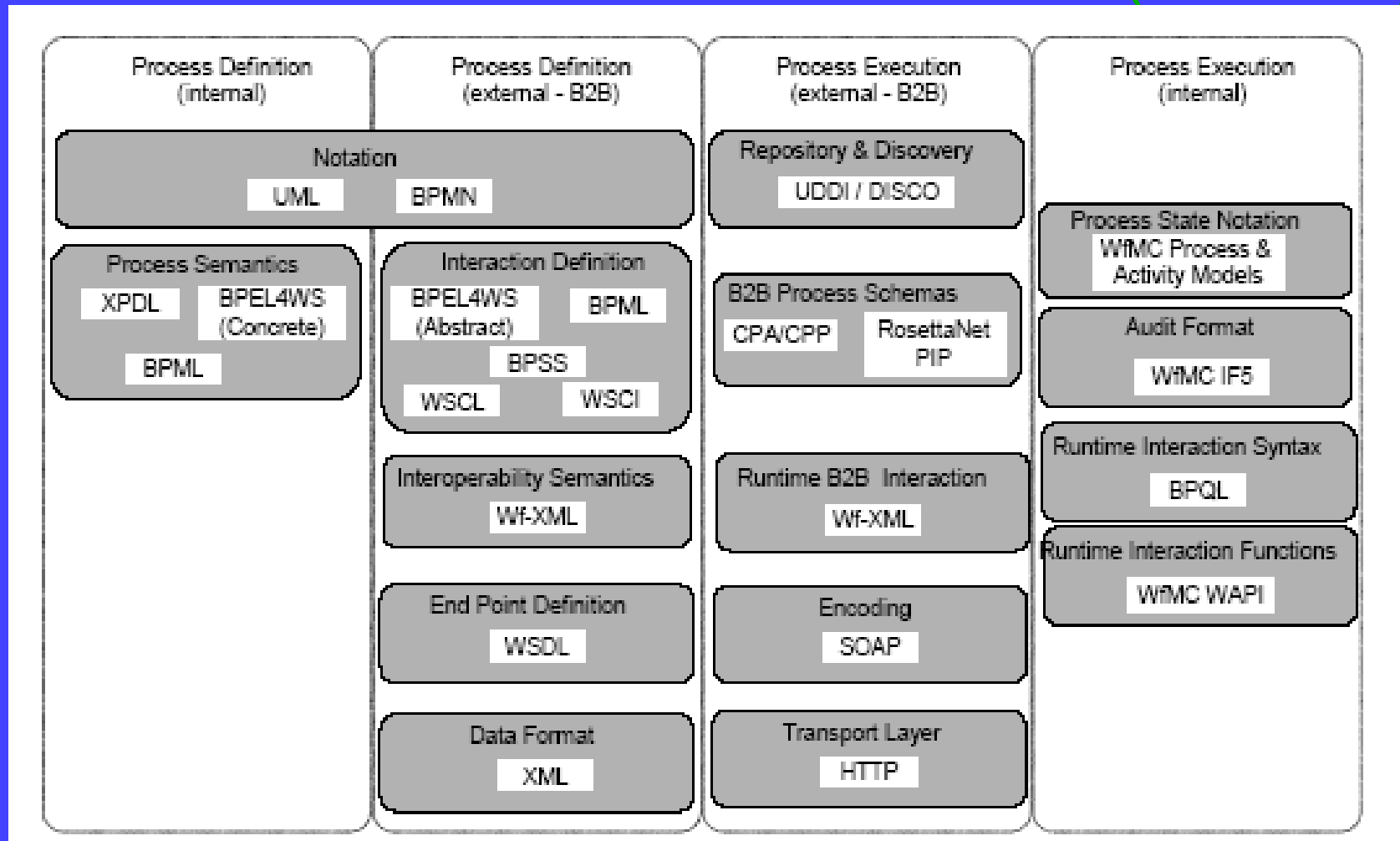


Standardisierungssprachen für Choreographie und Orchestrierung

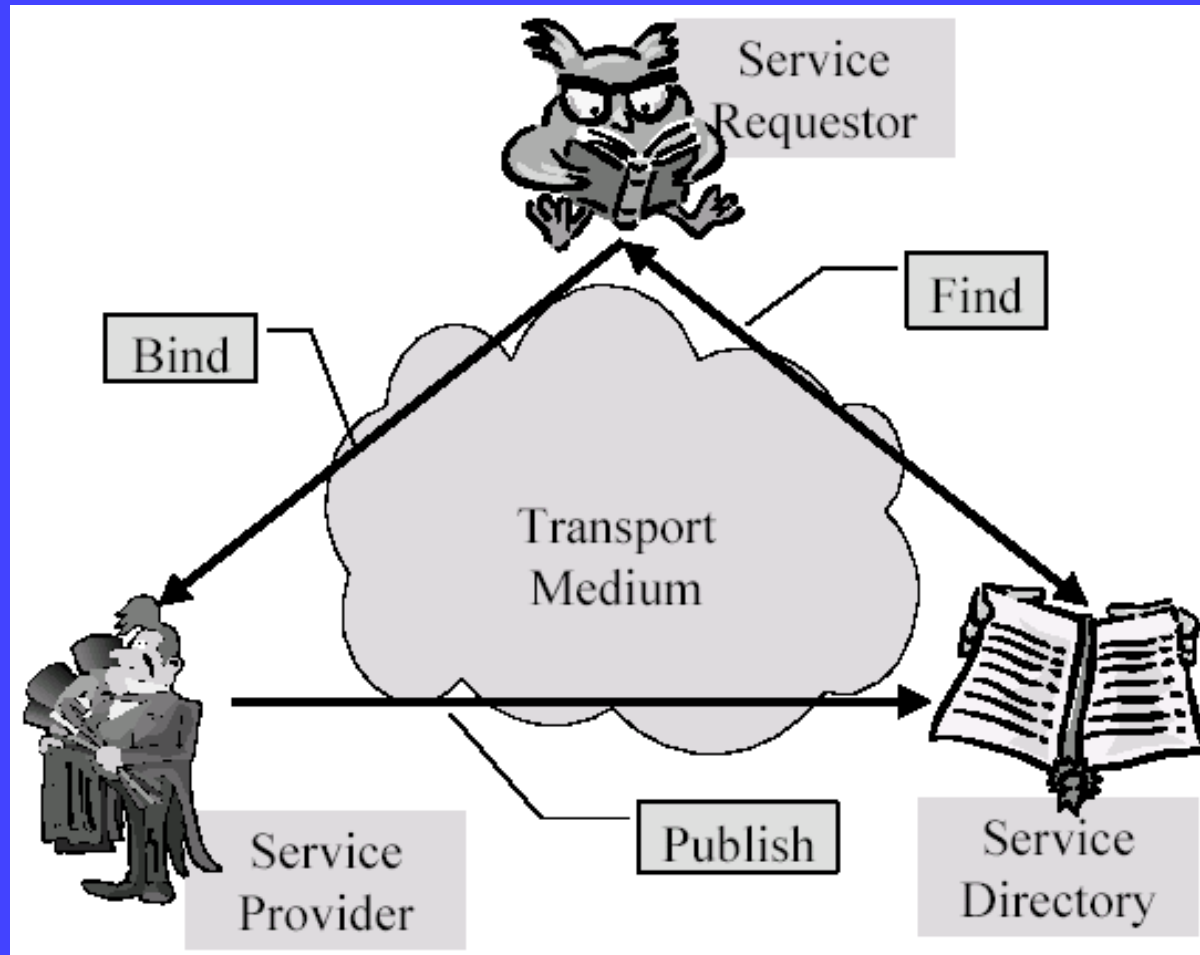


(Wiederholung aus Kapitel 5)

Standards im Workflow-Bereich (WFMC)



Service-orientierte Architektur (SOA): beteiligte Stellen



Schlussfolgerungen

Man kann...

- ◆ ...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)

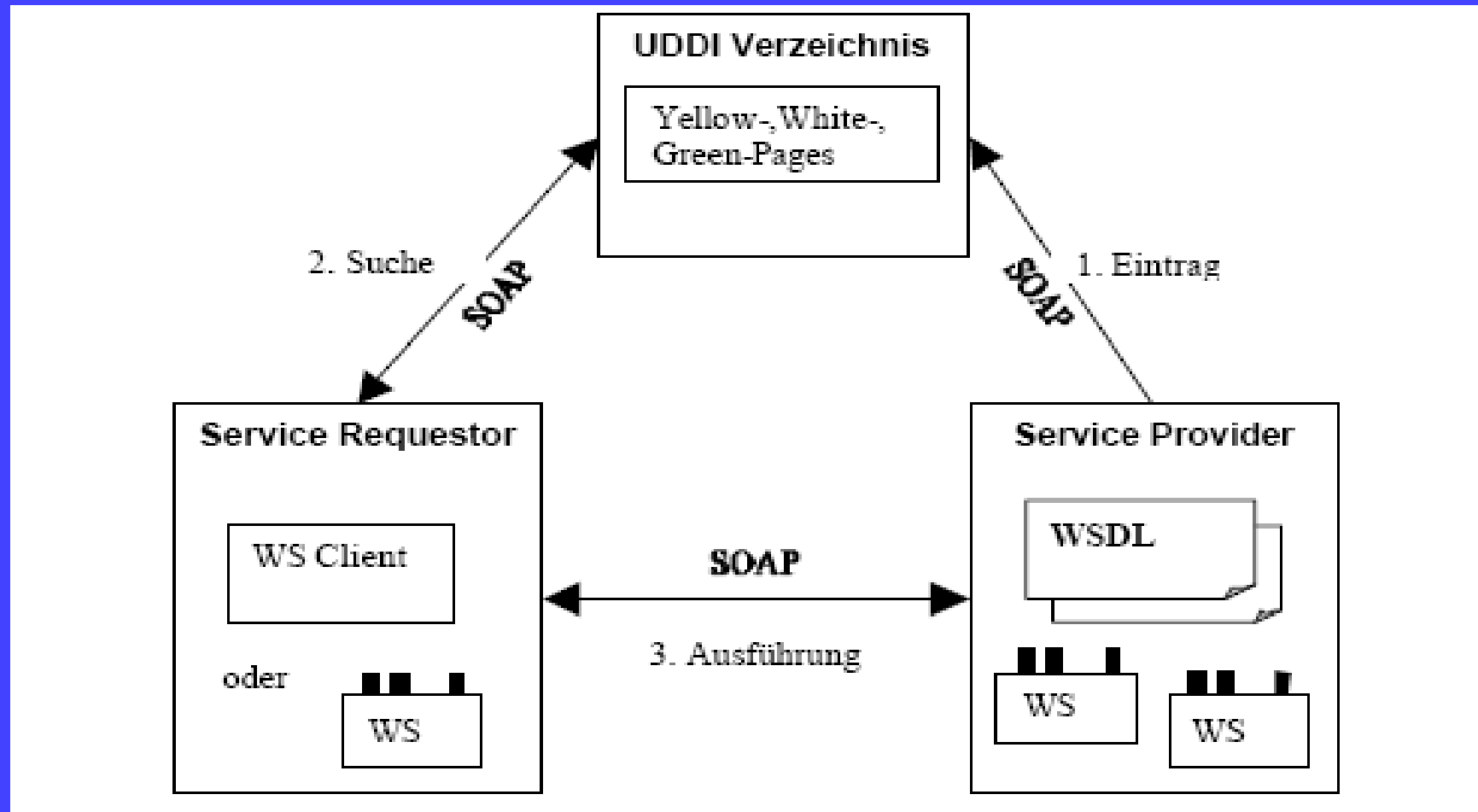
Diese Dienste umfassen

- ◆ ...einfache, fein granulare Funktionen...
- ◆ ...bis zu kompletten Geschäftsprozessen, d.h. grob granulare Funktionen

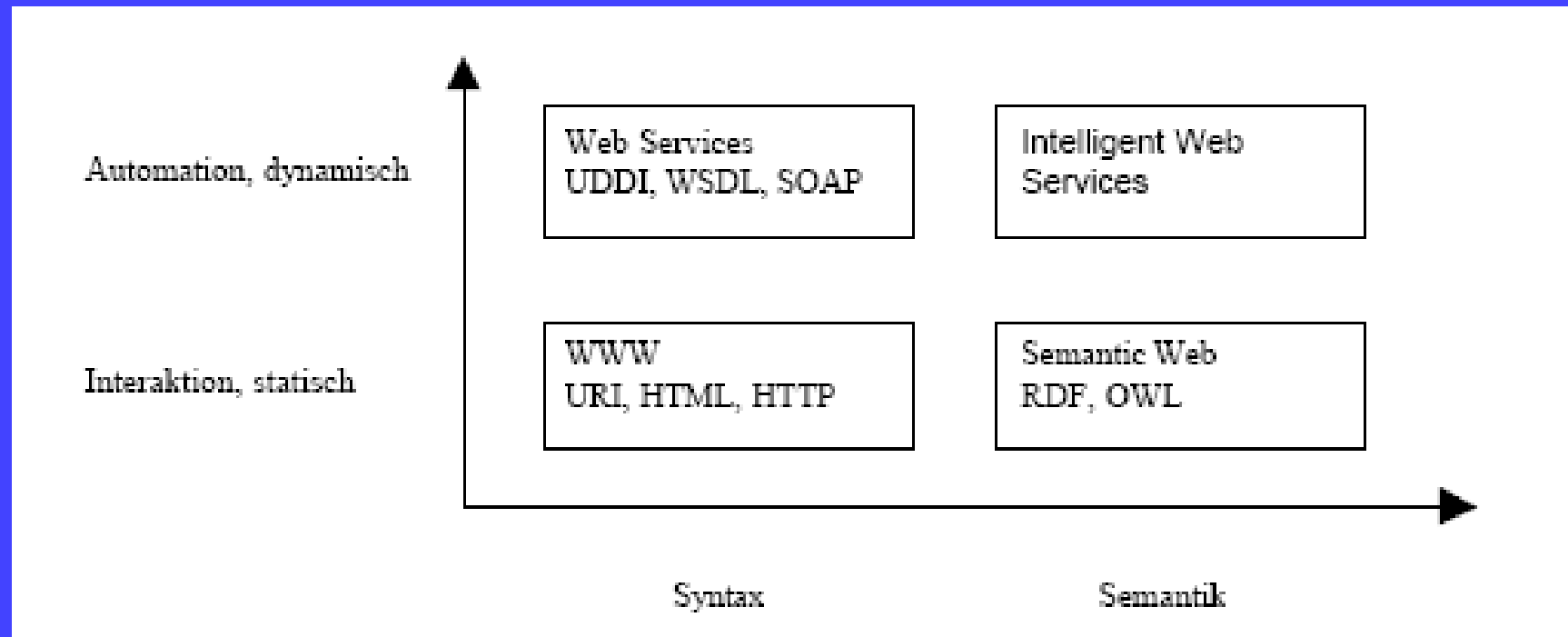
Workflow Technologie ist das Herz davon!



Service-orientierte Architektur (SOA) mit Web Services



Einordnung von Semantic Web



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



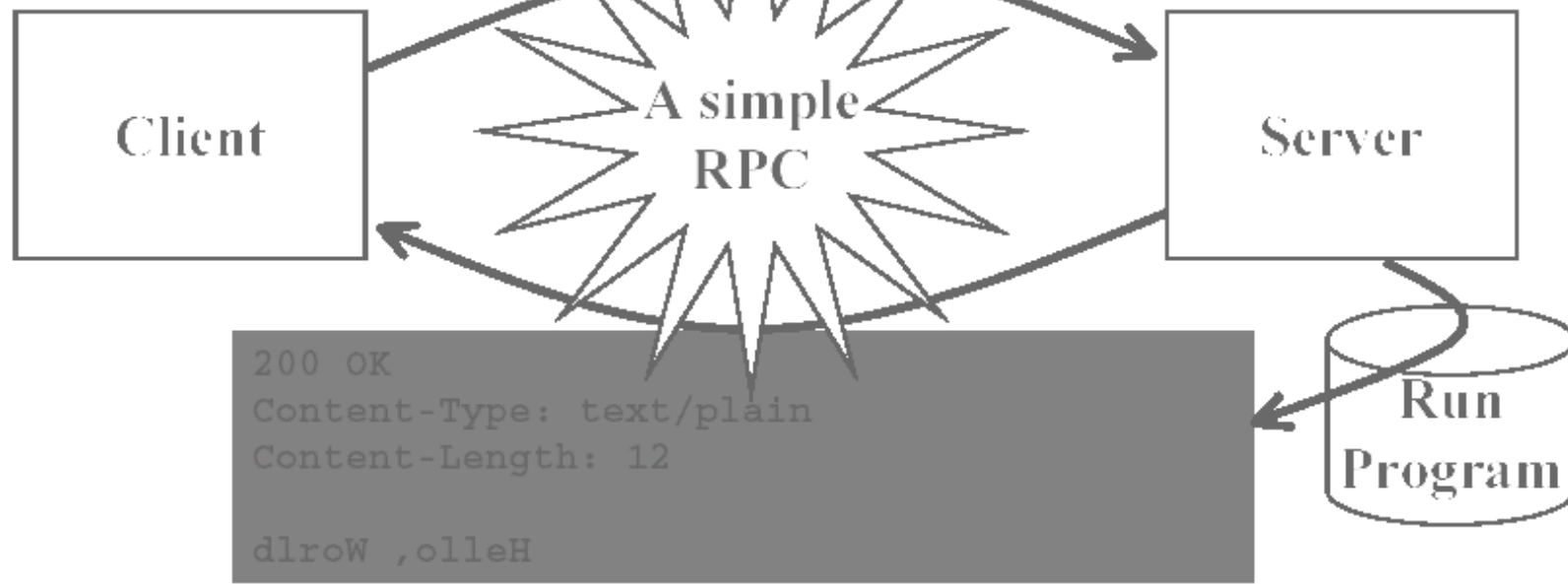
SOAP



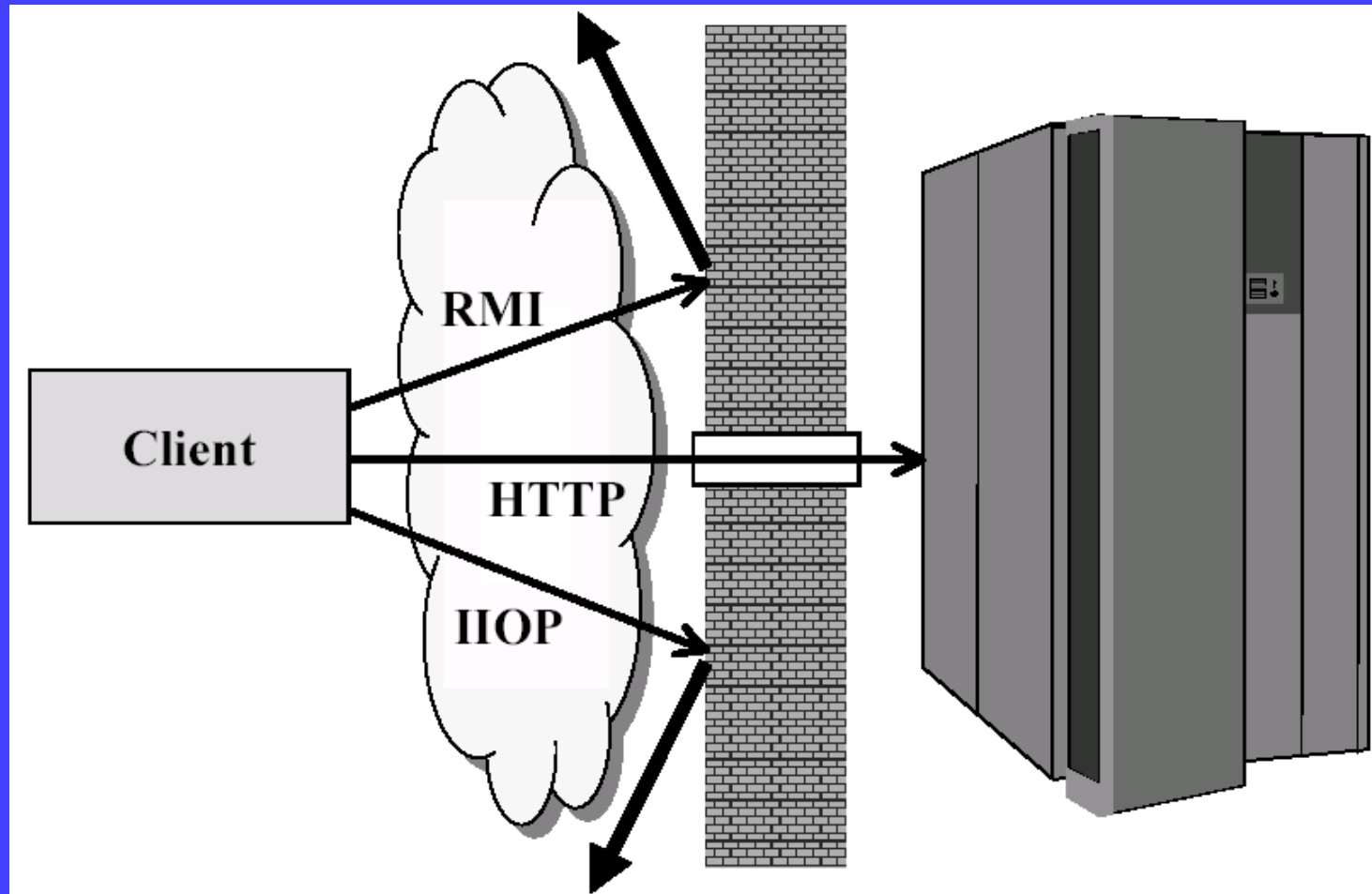
HTTP POST: Aufruf-Code

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

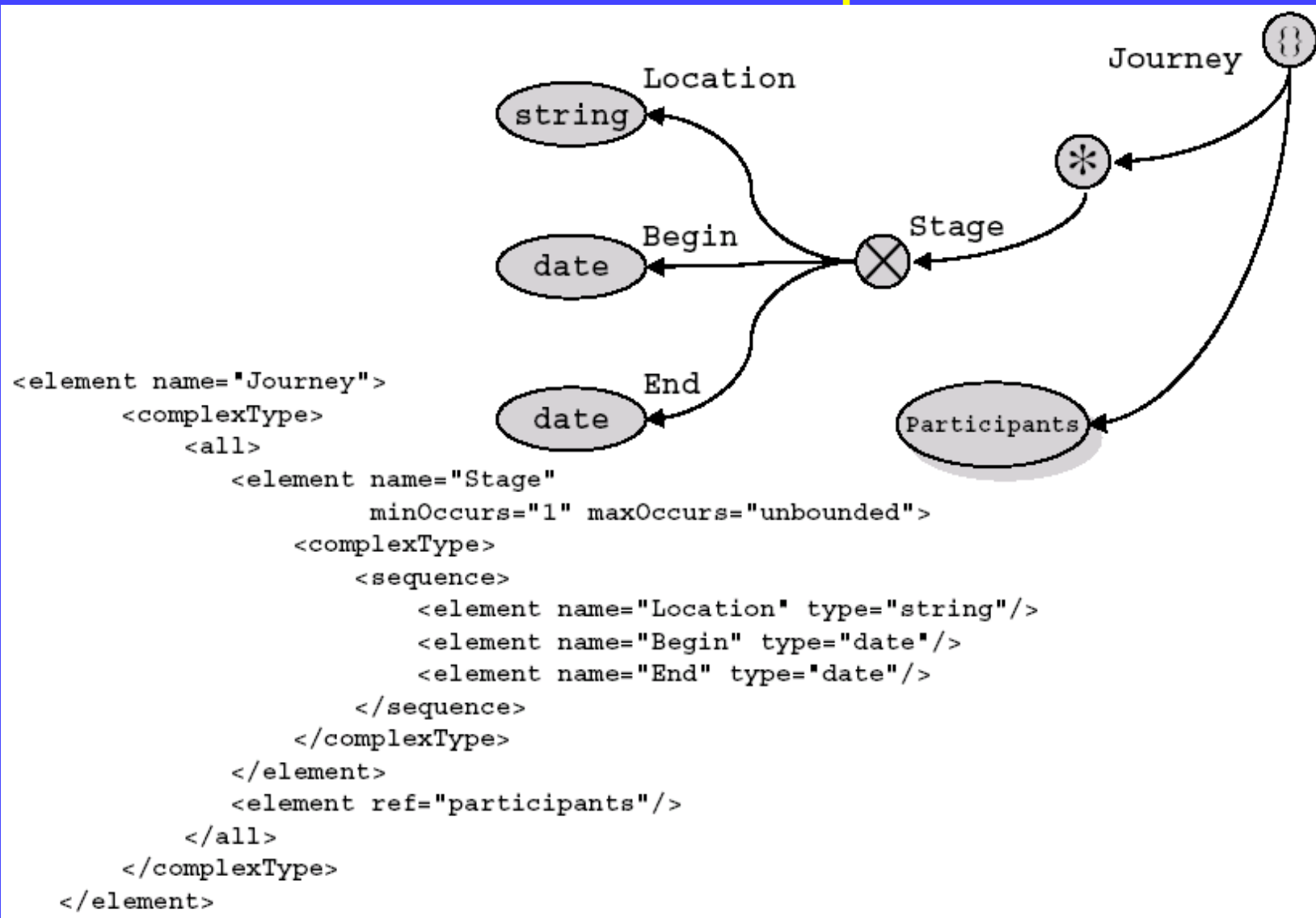
Hello, World
```



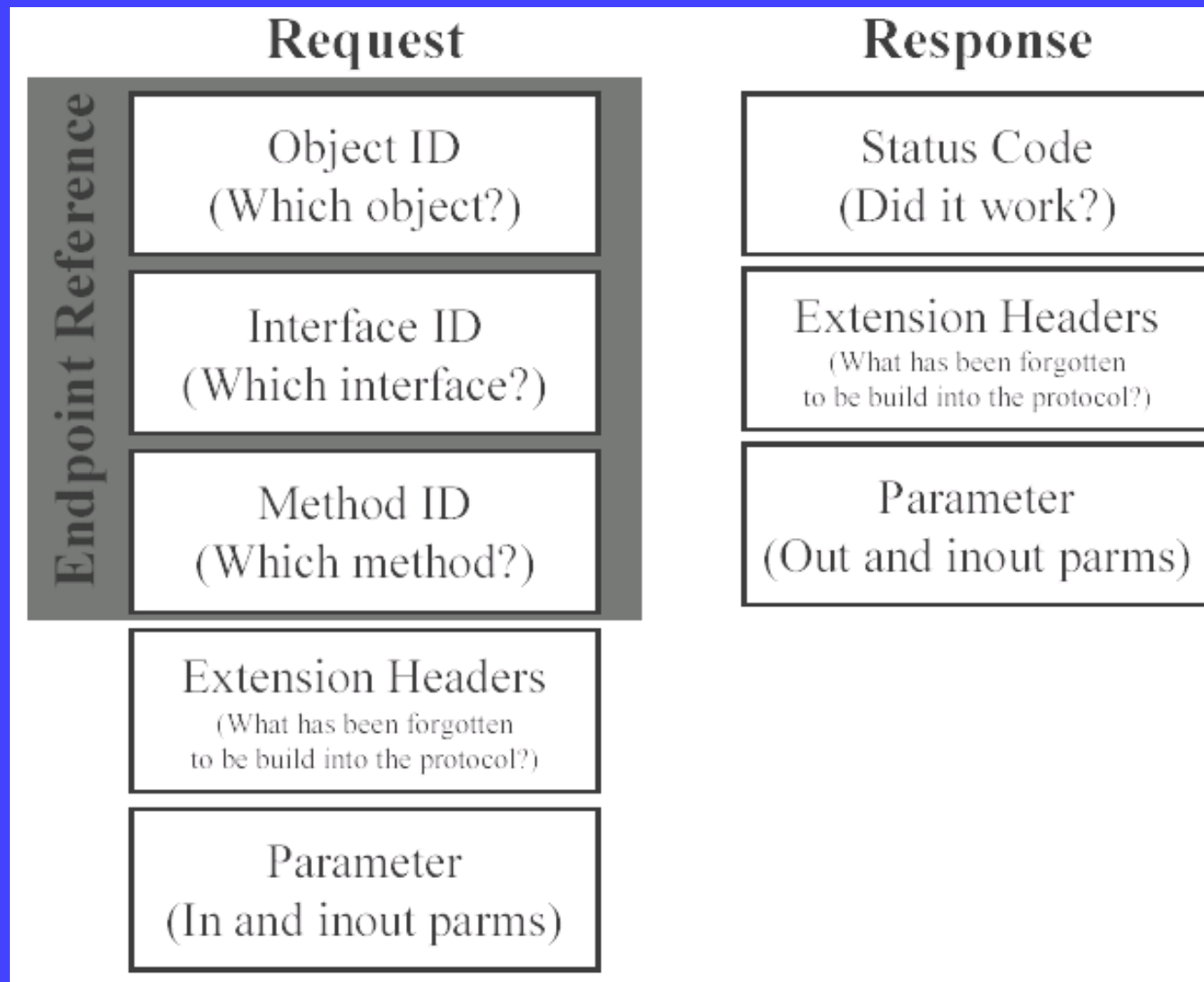
Die Firewall „durchtunneln“



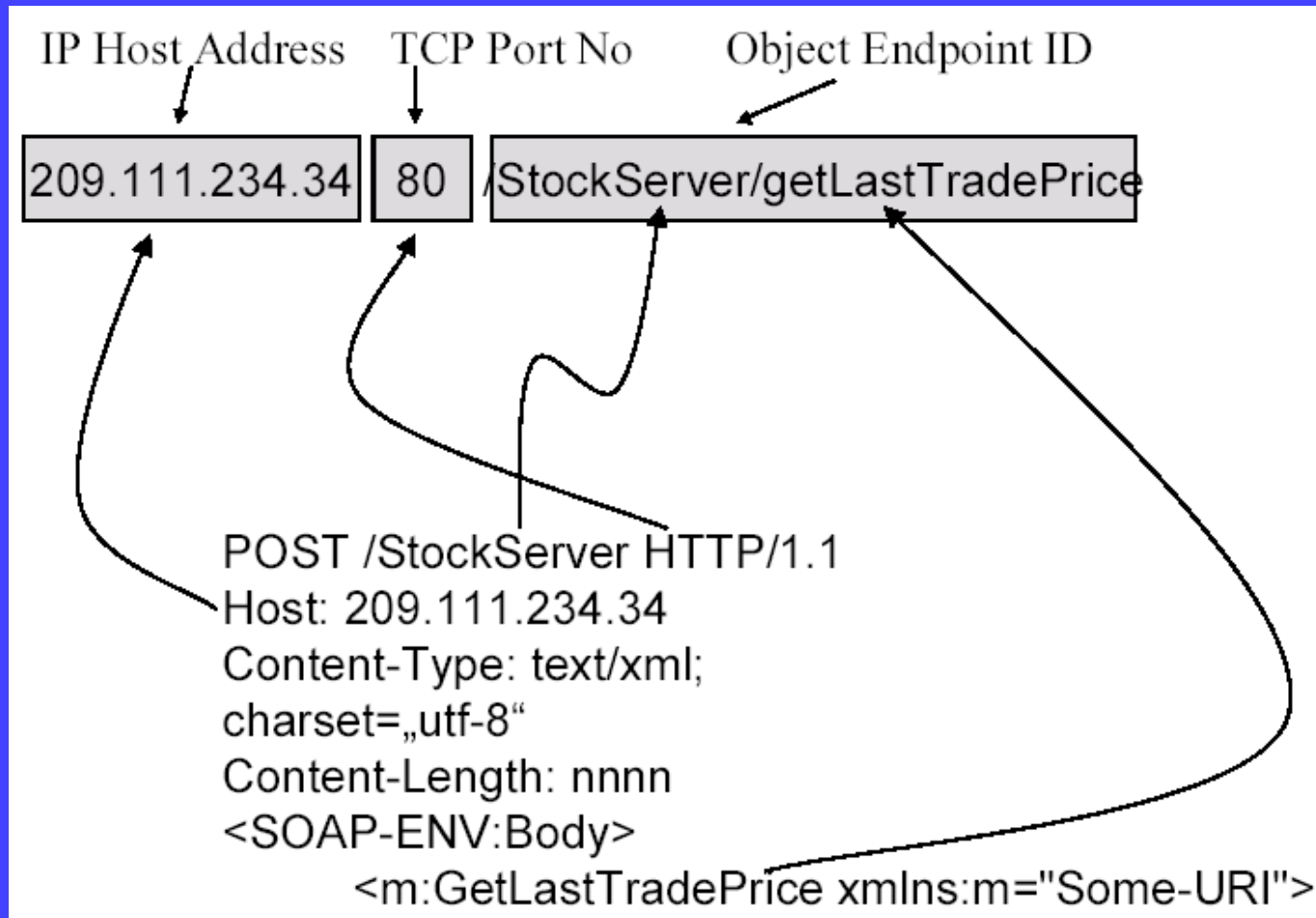
Ein XSD-Beispiel



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

```
<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>
```

Method
Name

Input



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>
```

Standard
Suffix



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**



UDDI



Wie findet man Dienste?

Wir brauchen ein global verfügbares Verzeichnis!

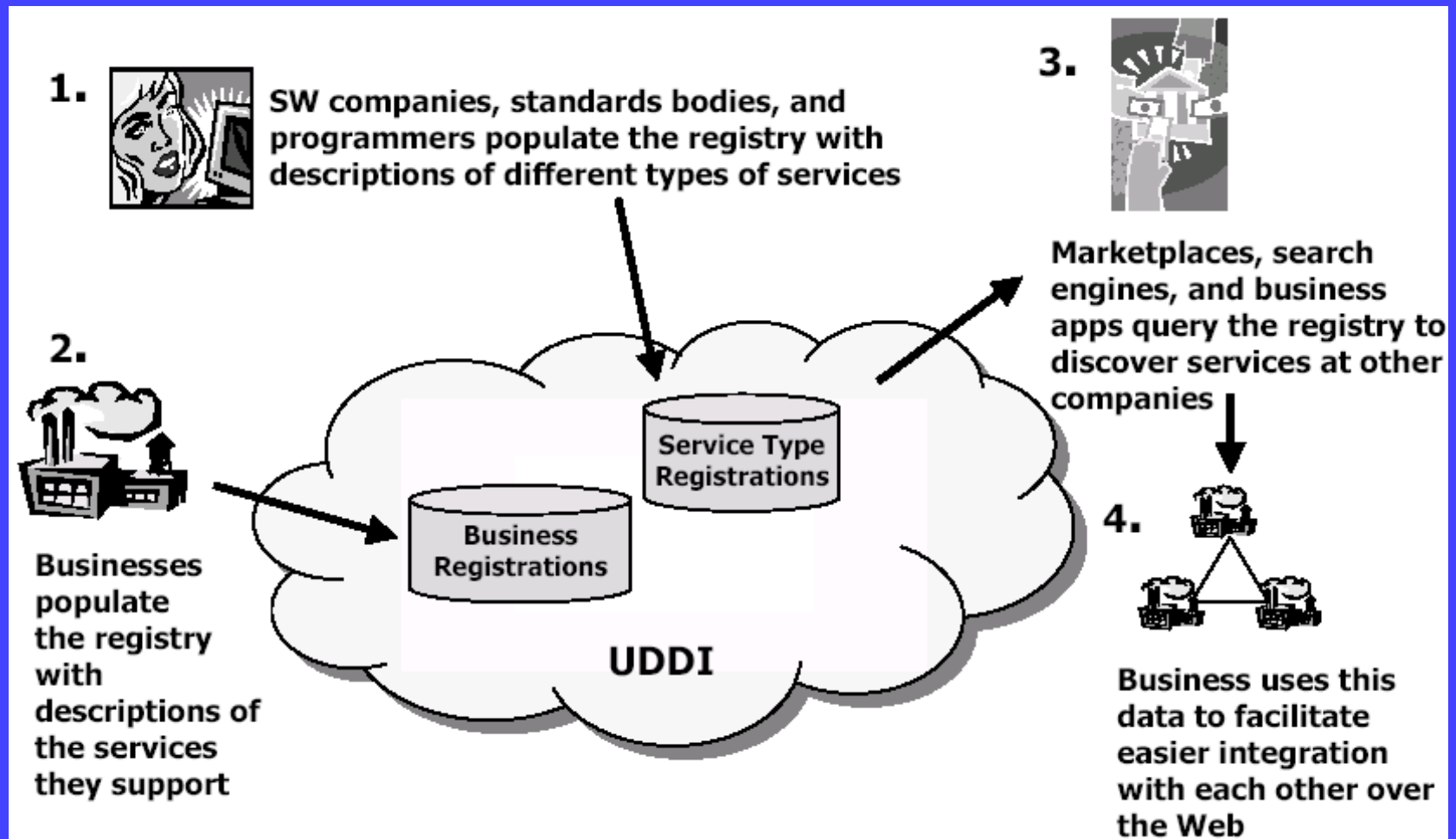
- ◆ ...um Dienste basierend auf öffentlichen Anfragen von Service-Providern zu katalogisieren
- ◆ ...um Taxonomien zur Unterstützung der Suche von geeigneten Dienste zu unterstützen
- ◆ ...um Dienste durch menschliche Benutzer oder Programme suchen zu können

UDDI

(Universal Description, Discovery & Integration)



Wie arbeitet UDDI



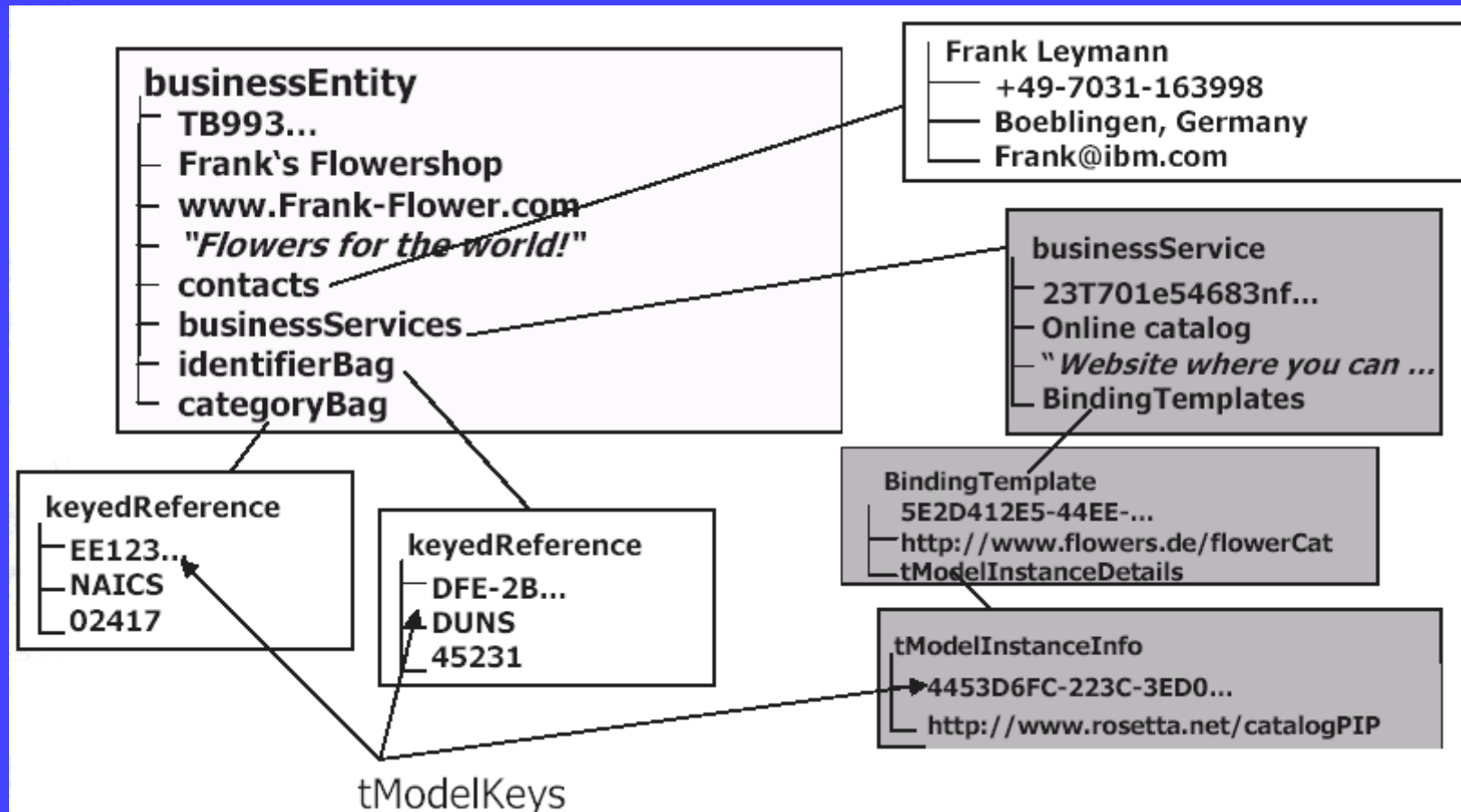
Registry Daten

- ◆ Firmen registrieren öffentliche Informationen über sich

- ◆ Standardisierungsstellen, Programmierer, Firmen registrieren Informationen über ihre Service Typen („tModels“)

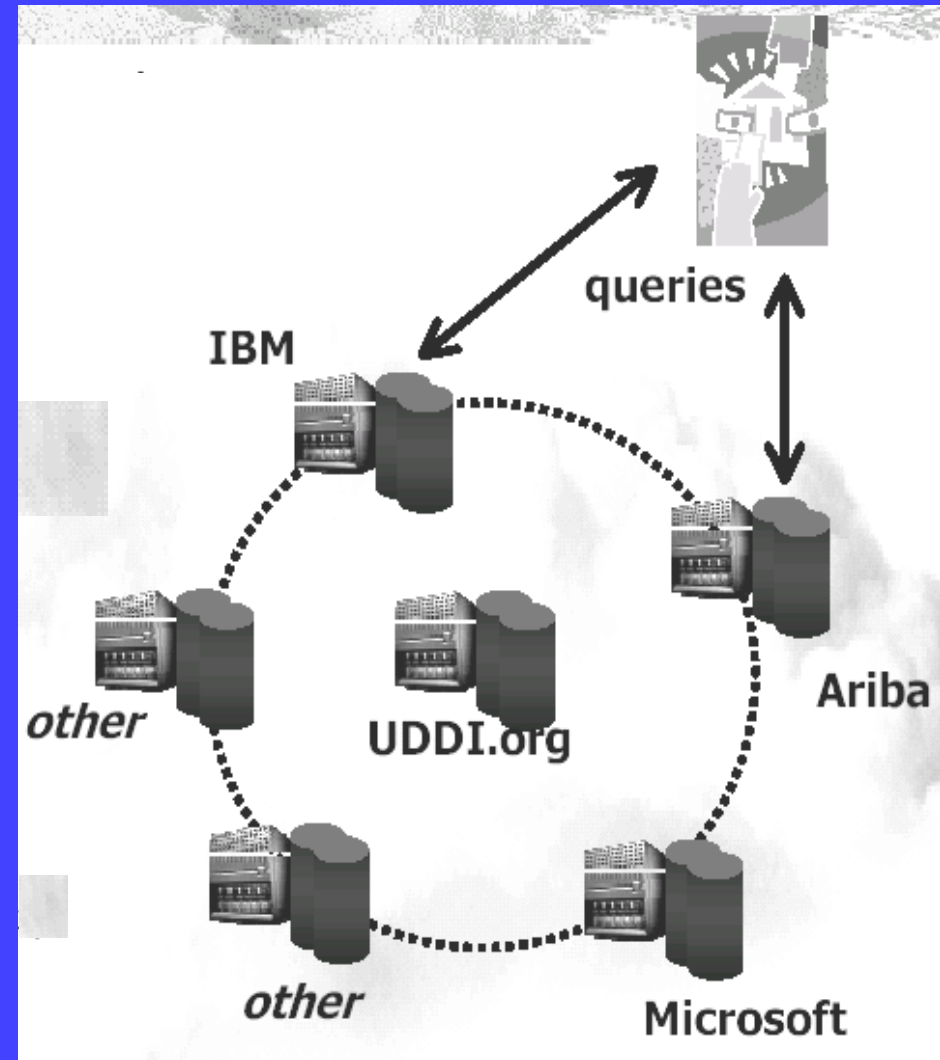


Beispiel einer Registrierung



Registrierungsoperation

- ◆ Peer-Knoten (Websites)
- ◆ 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?

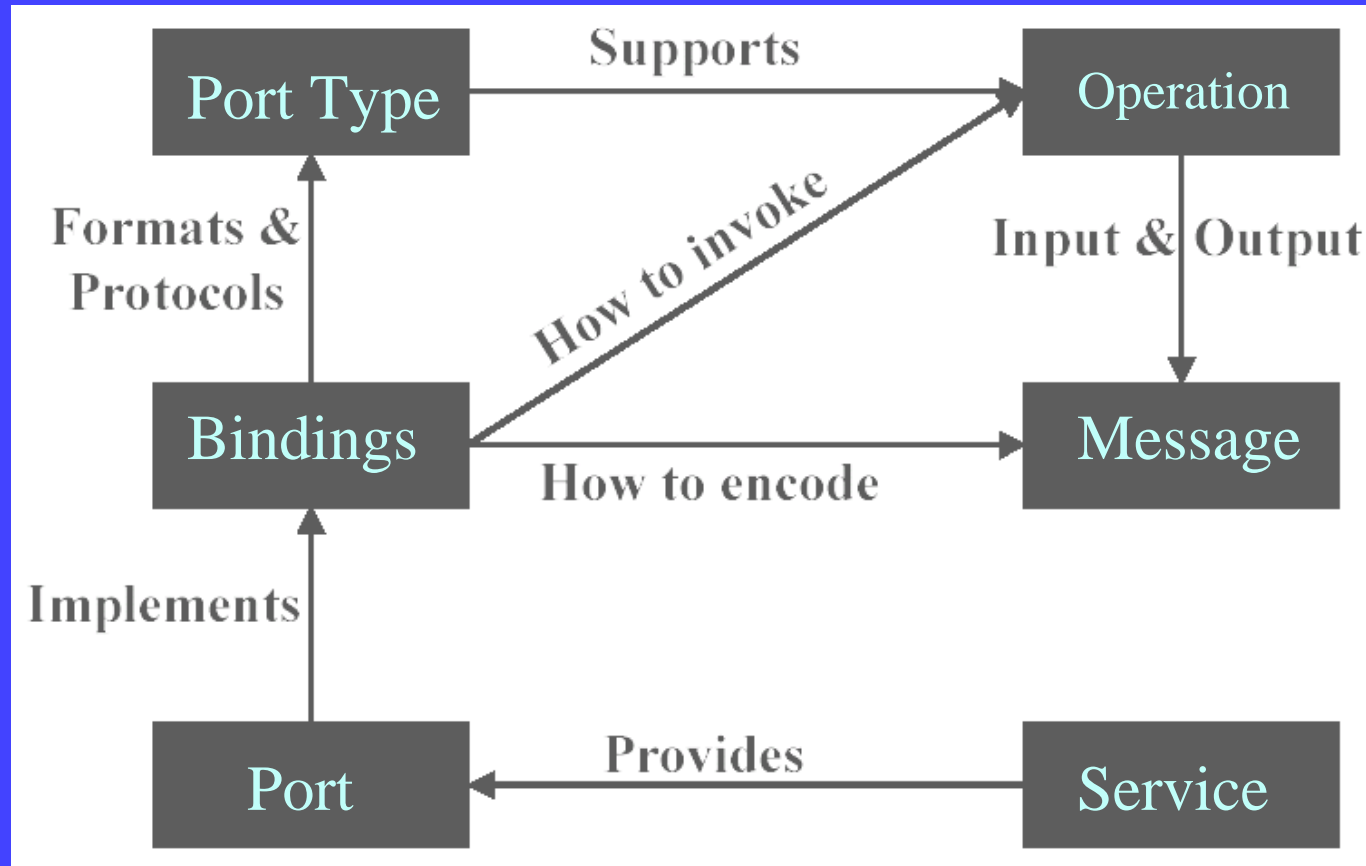
- ◆ Es ist gut Web Services zu nutzen, die andere definiert haben und über die man sich geeinigt hat (= tModels)
- ◆ ...aber: was, wenn man seine eigenen Web Services veröffentlichen will
 - ◆ z.B. einfache Funktionen, die auch für andere nützlich sind
 - ◆ oder komplette Geschäftsprozesse, die man anbieten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

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

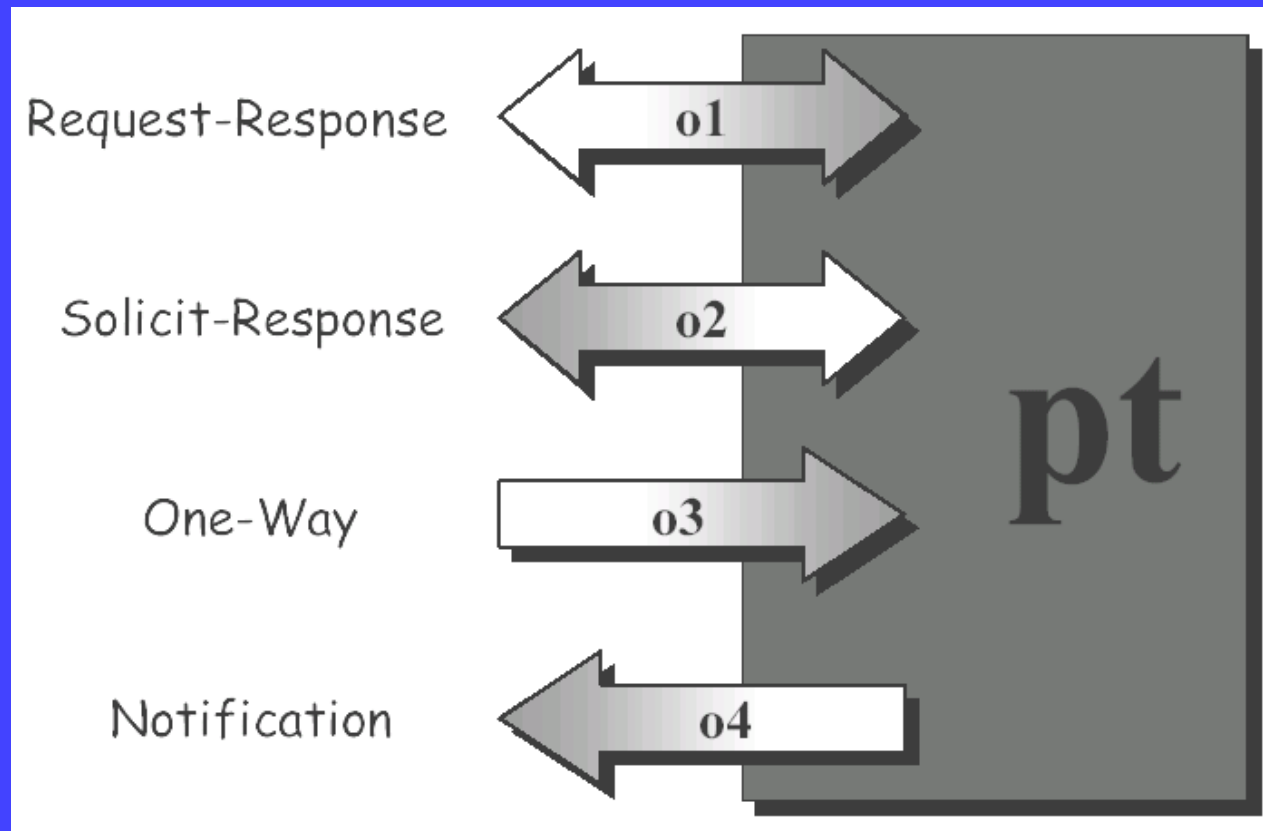
WSDL (Web Services Description Language)



WSDL Bestandteile



Port Typen: graphische Darstellung



Beispiel

```
<?xml version="1.0"?>
<definitions name="StockQuote"
xmlns:tns="http://leymann.com/stockquote.wsdl"
xmlns:xsd="http://leymann.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:timeInstant"/>
  </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

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

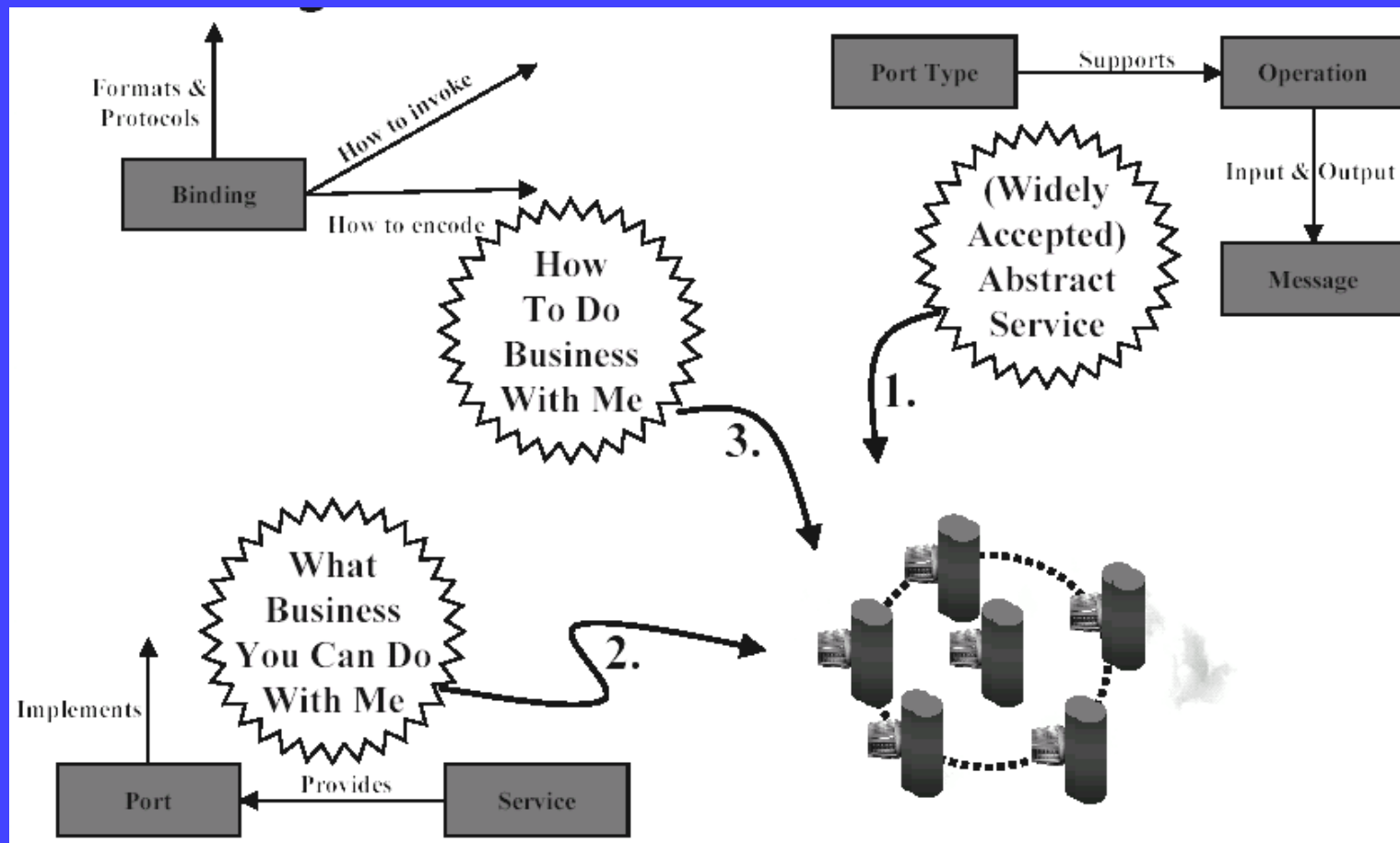


Beispiel

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



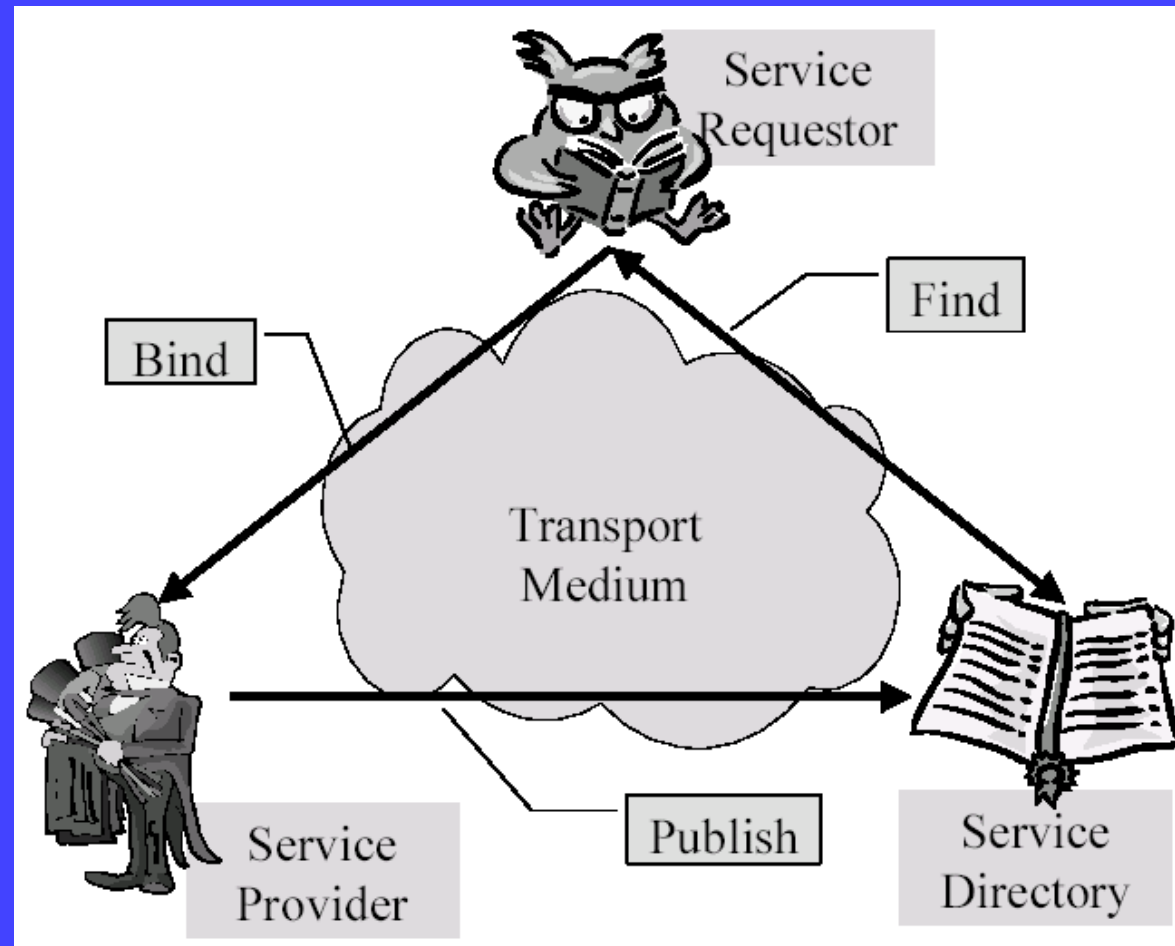
Wie macht man Dienste verfügbar



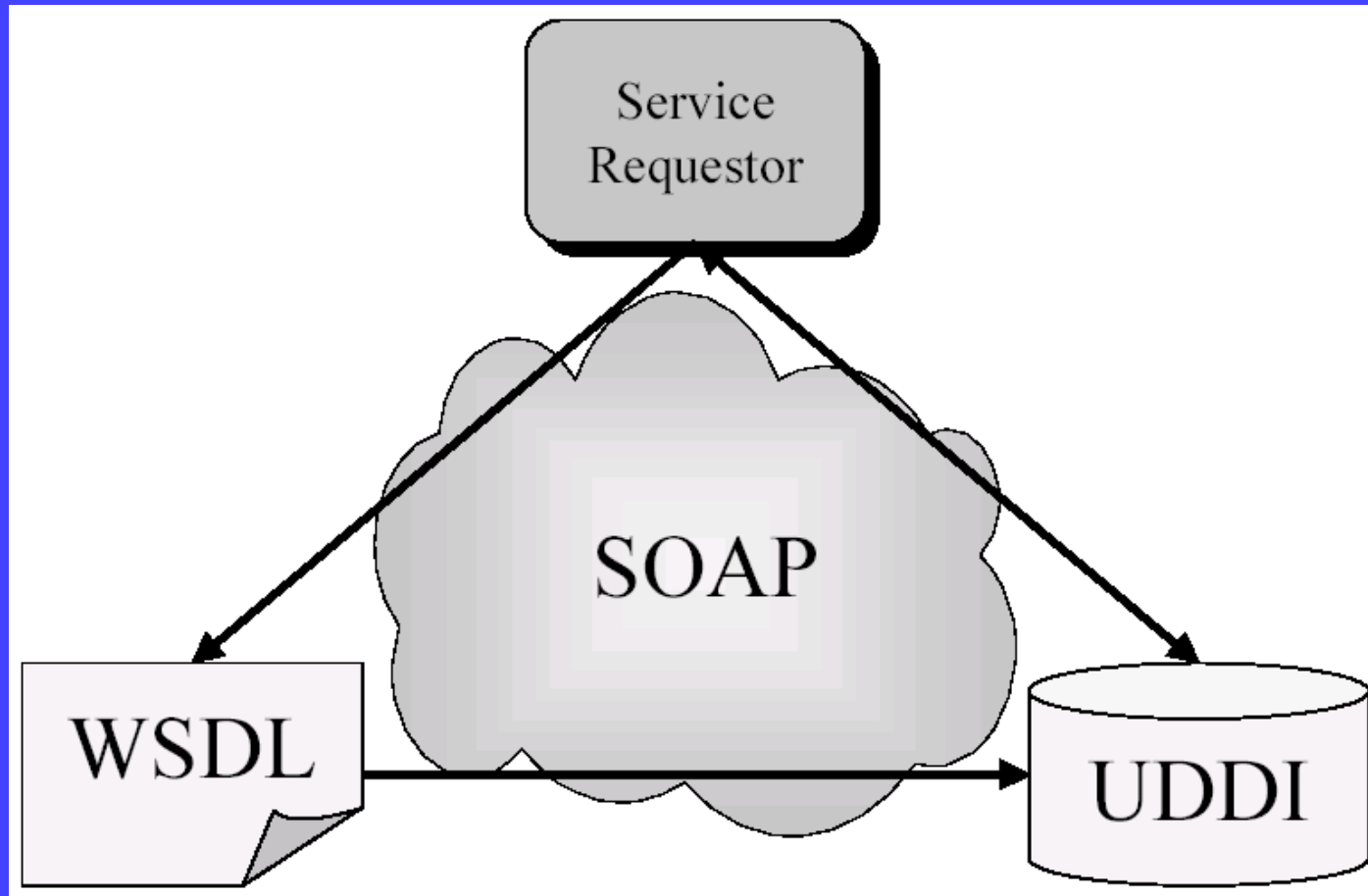
Zusammenhang mit der Service-Orientierten Architektur



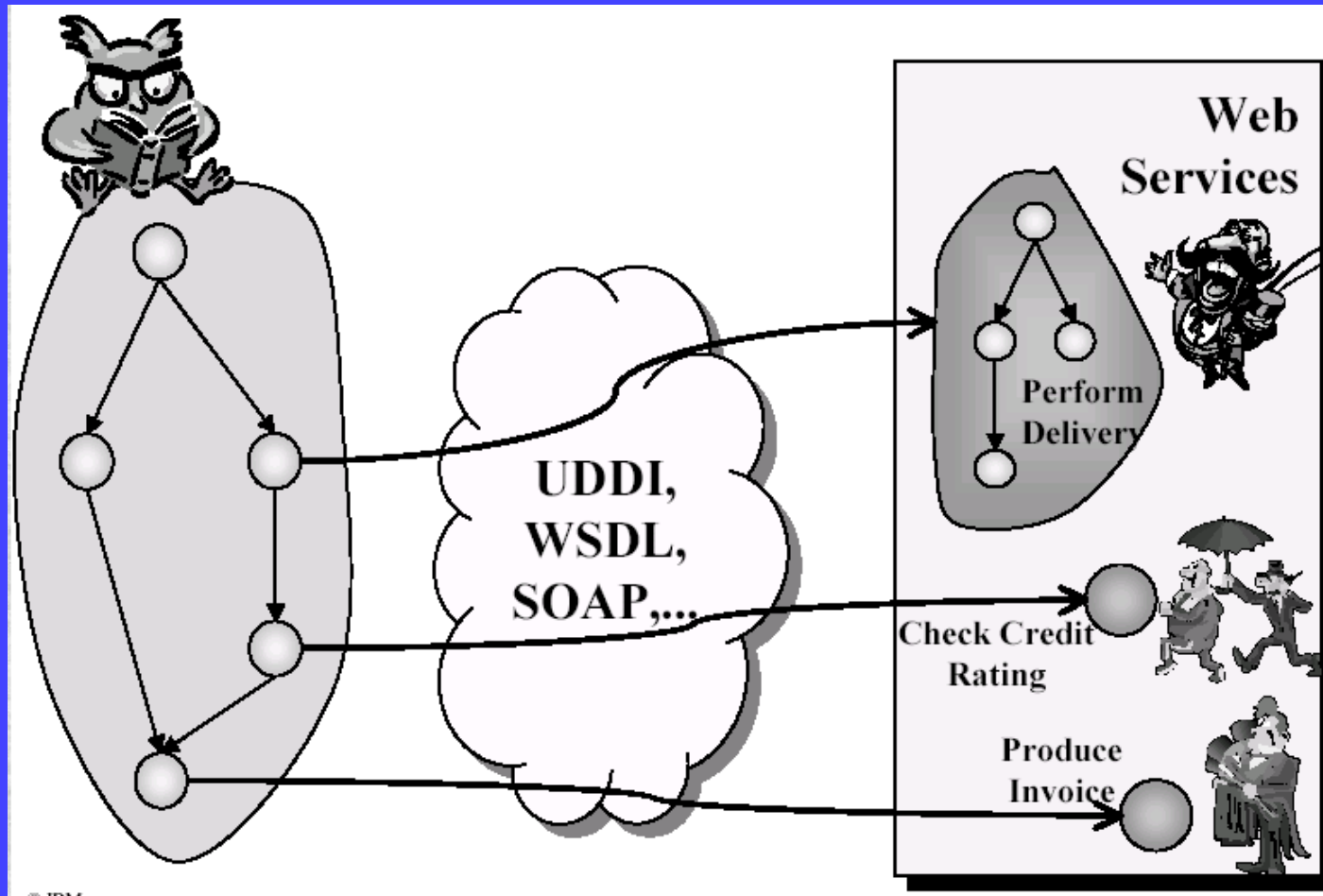
SOA - Wiederholung



...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

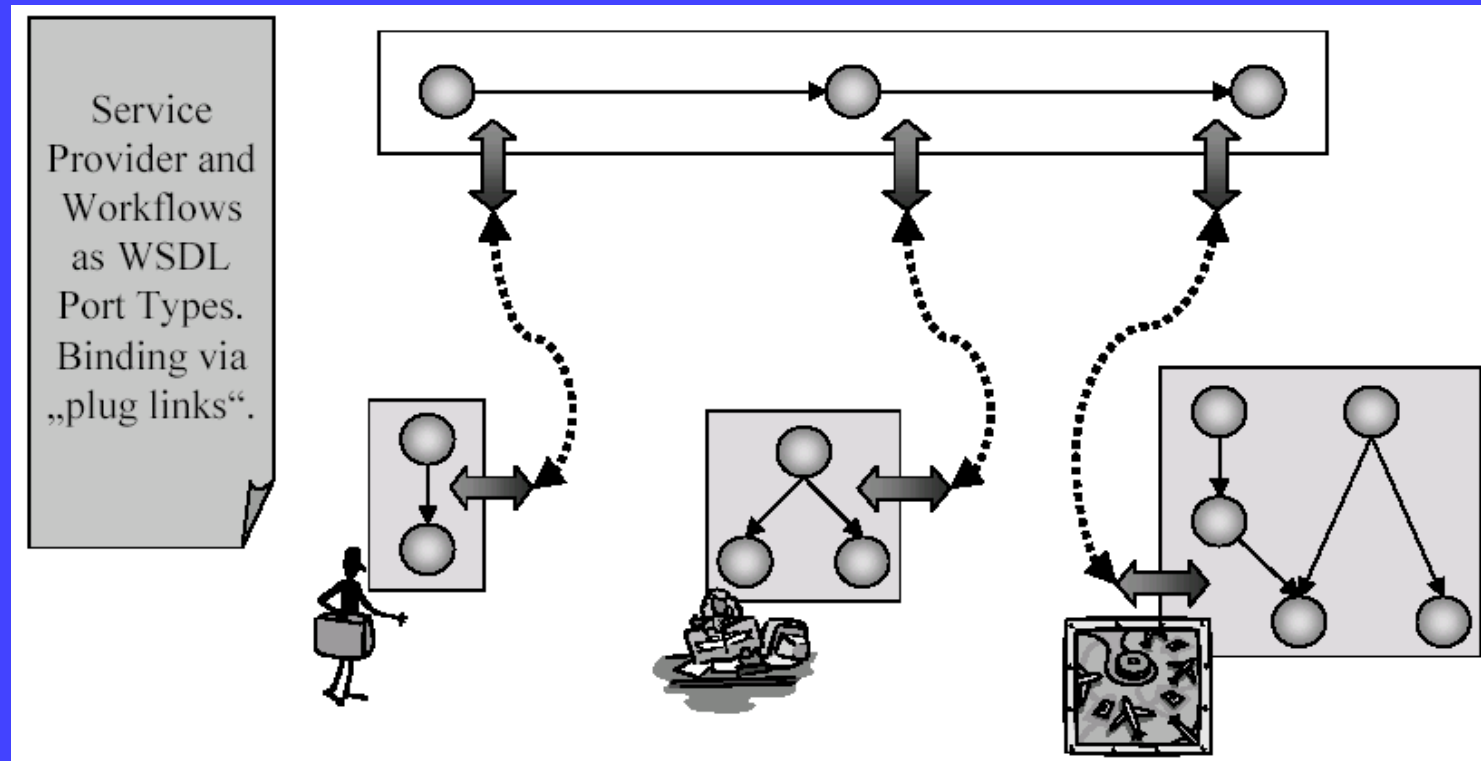
Web Services Choreographie

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.



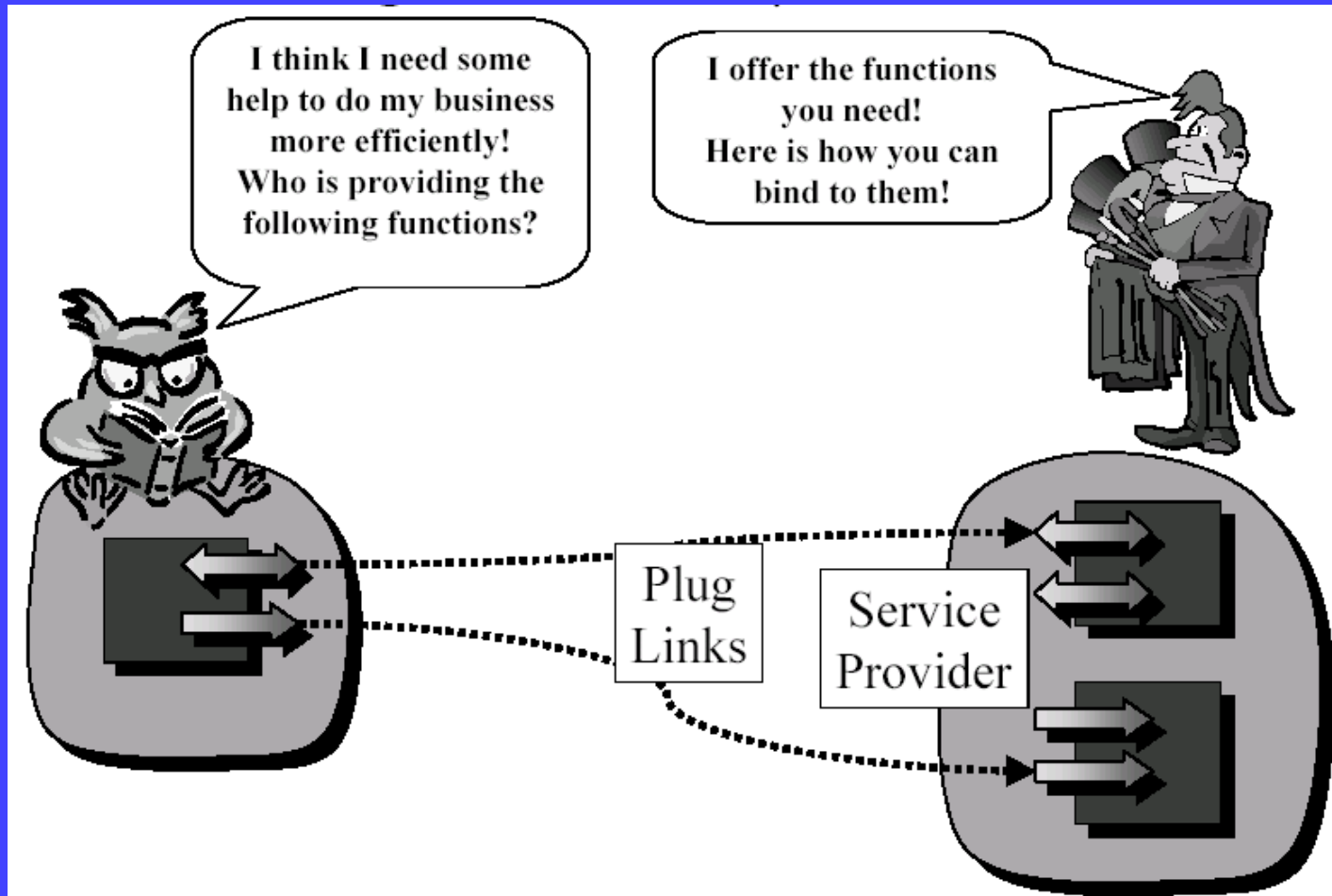
Beziehung zur Service-orientierten Architektur



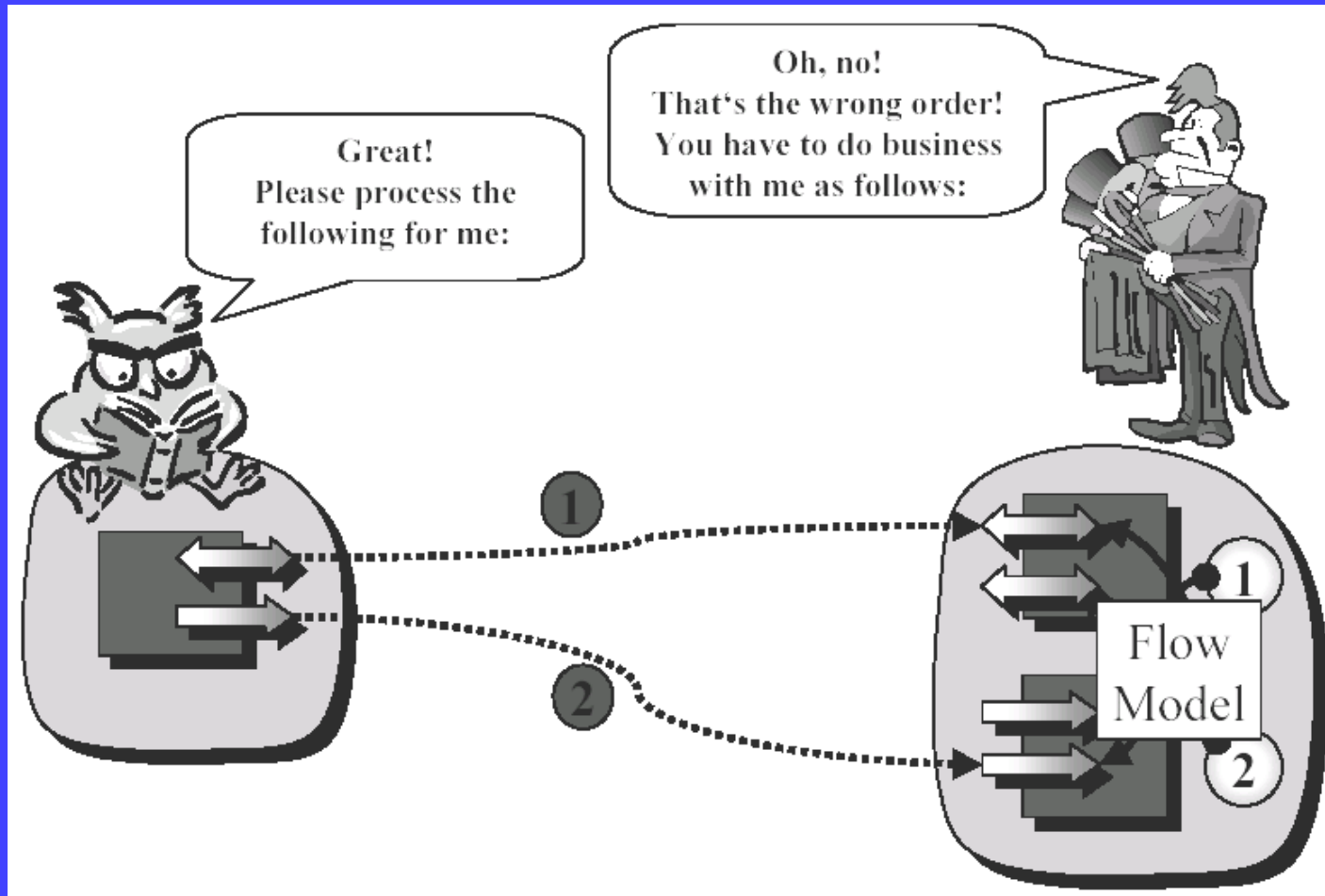
Eine abstraktere Sicht: High-Level Szenarios



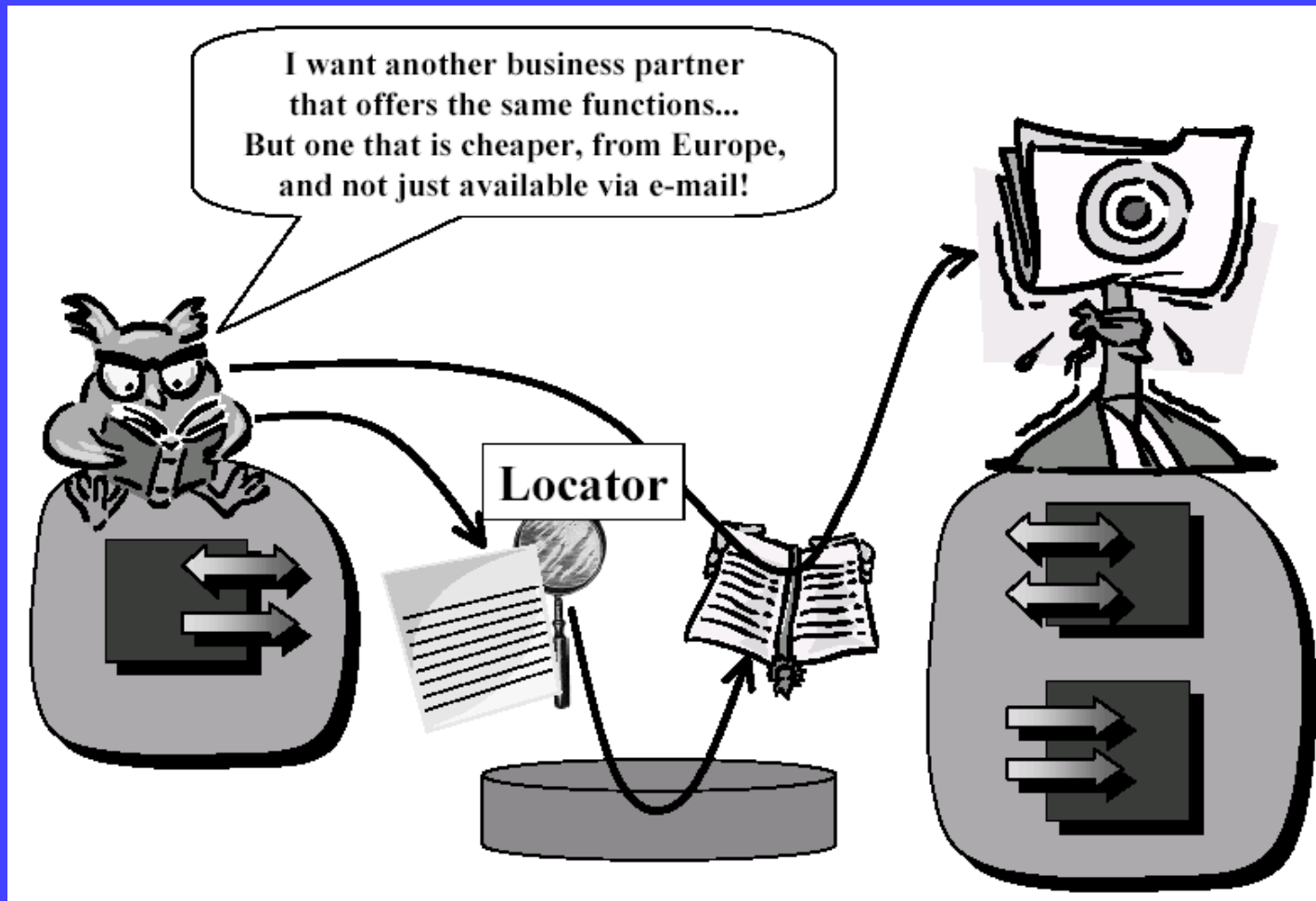
Partnerschaften errichten



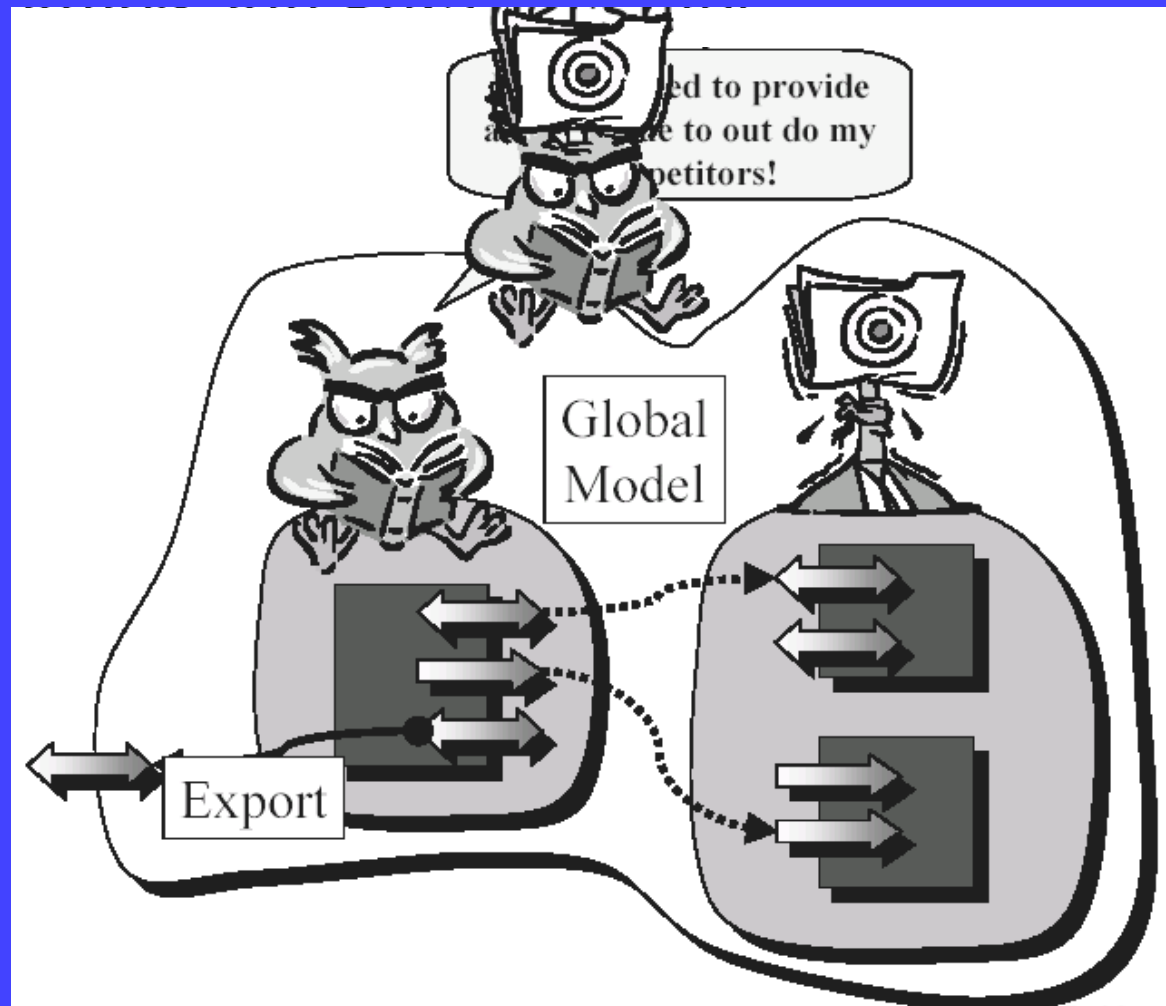
Ordnung zählt!



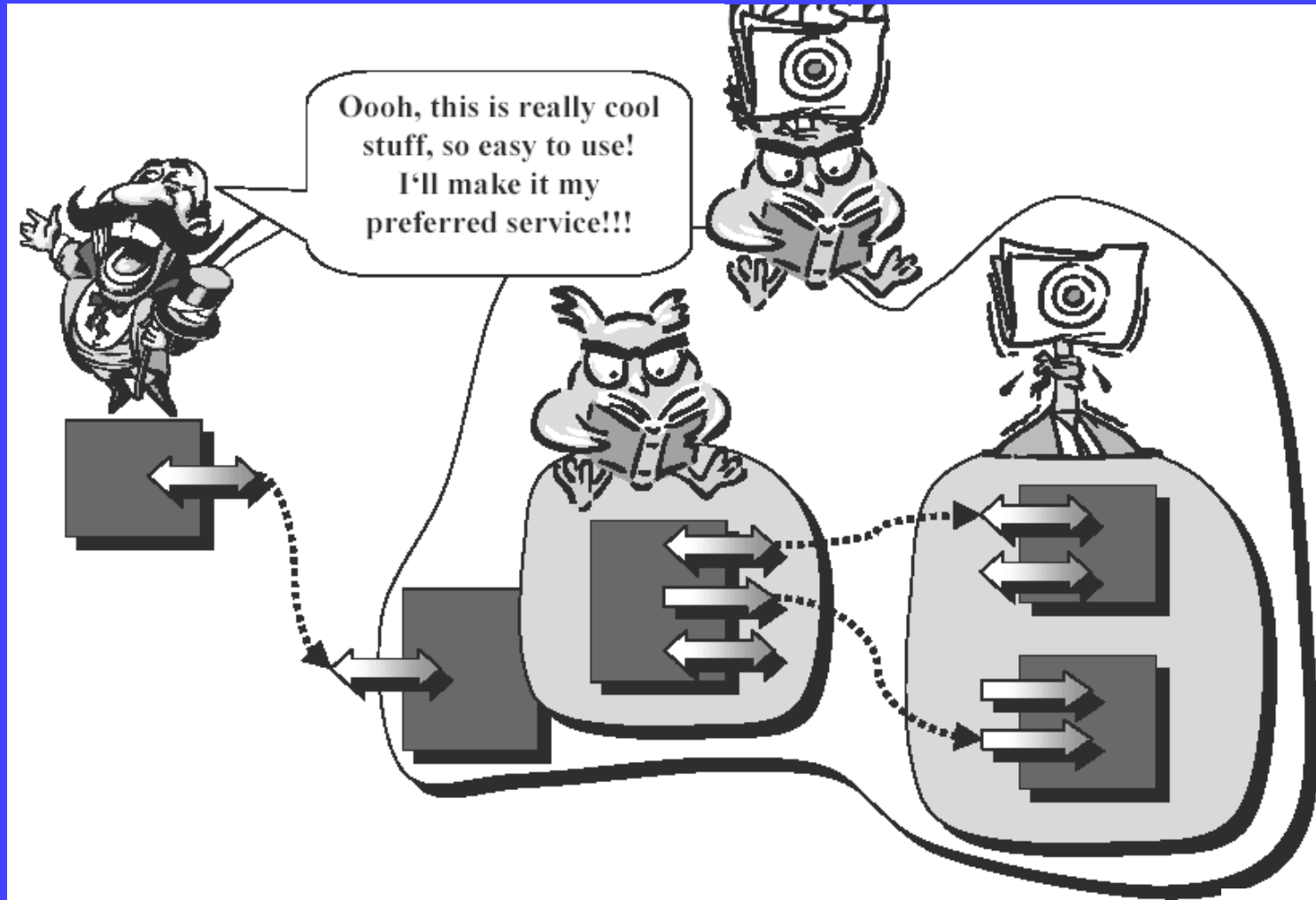
Flexibilität!



Zusatznutzen durch Partnerschaft!



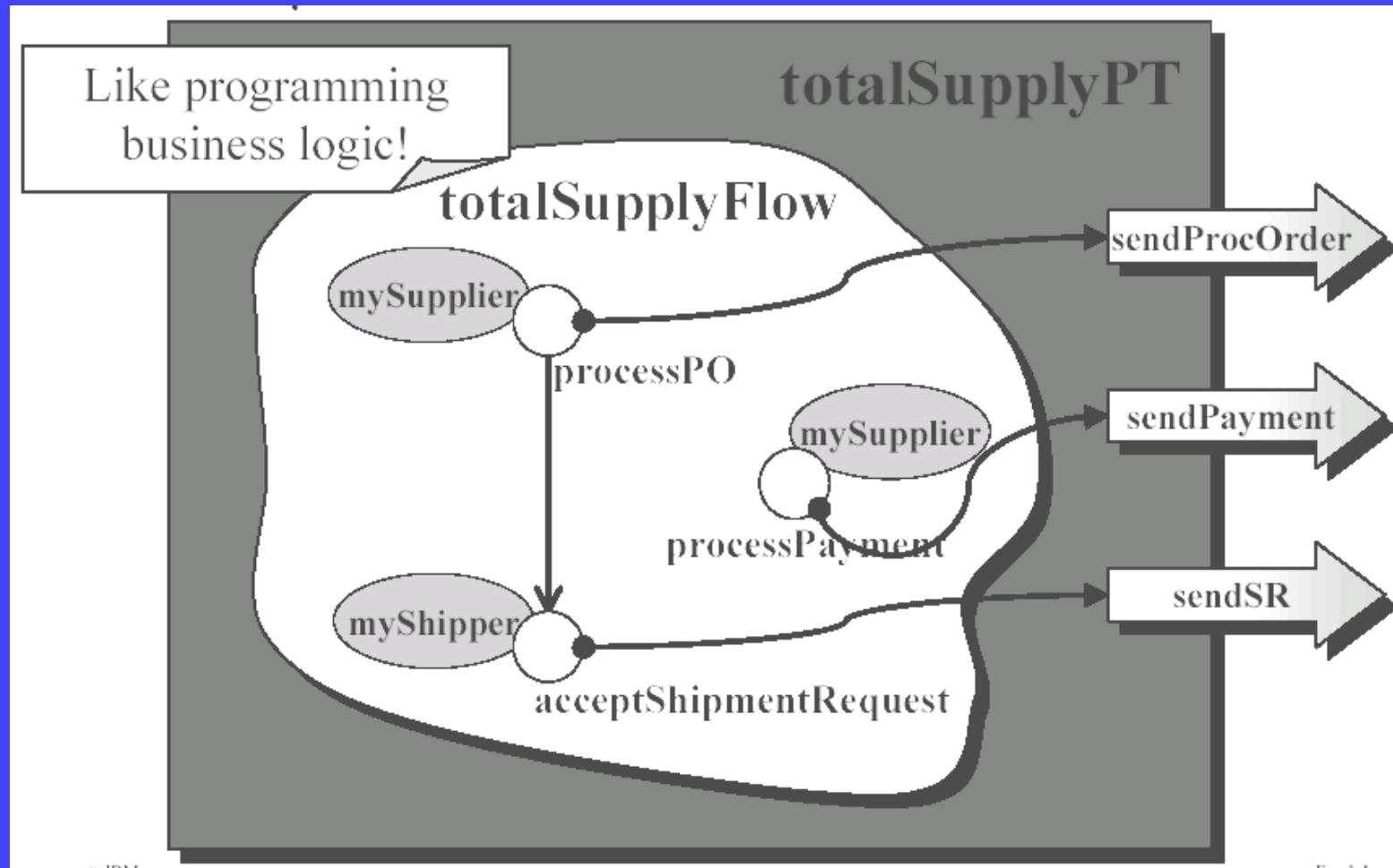
Zusatznutzen durch Partnerschaft (2)!



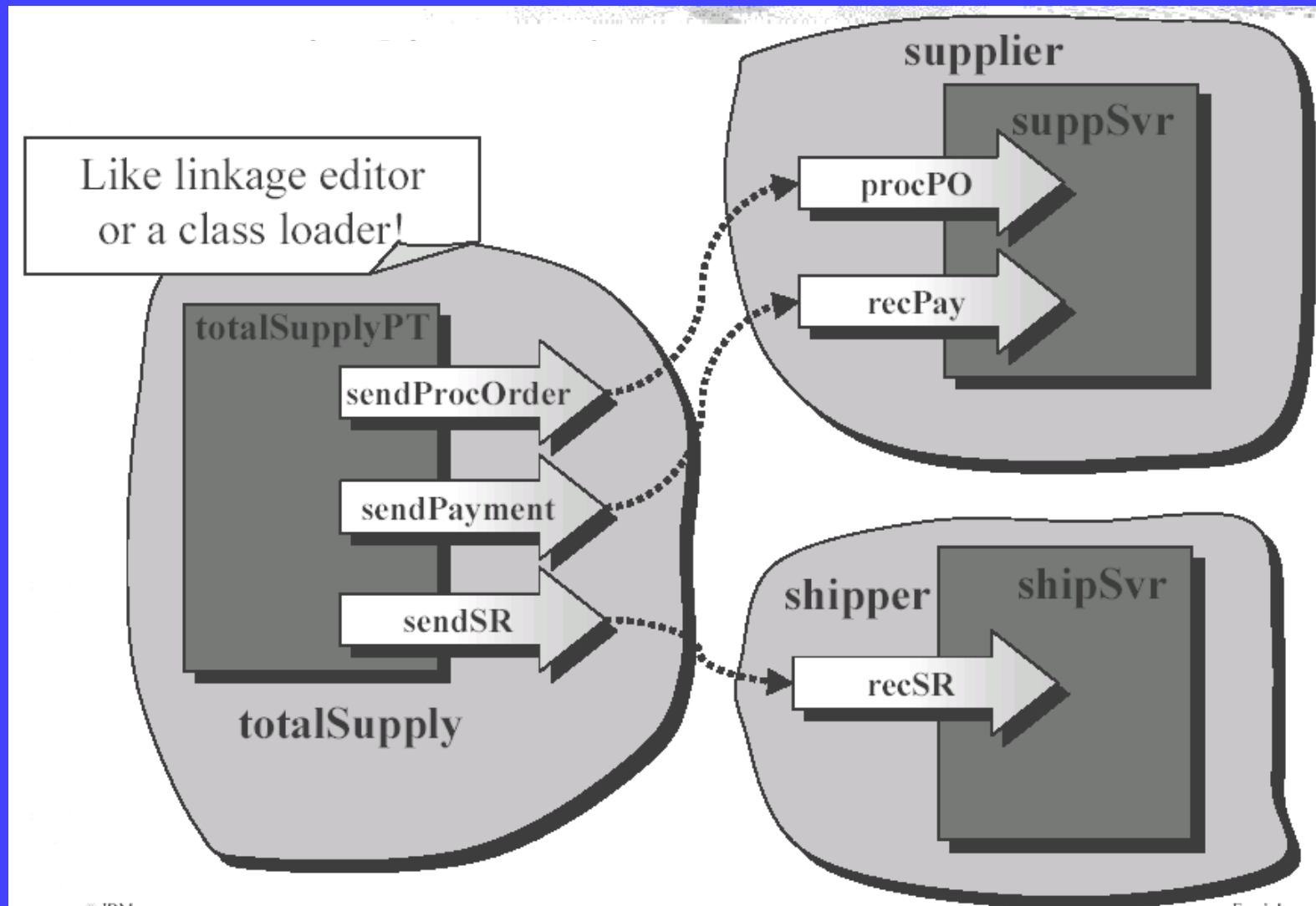
Ein globales Beispiel!



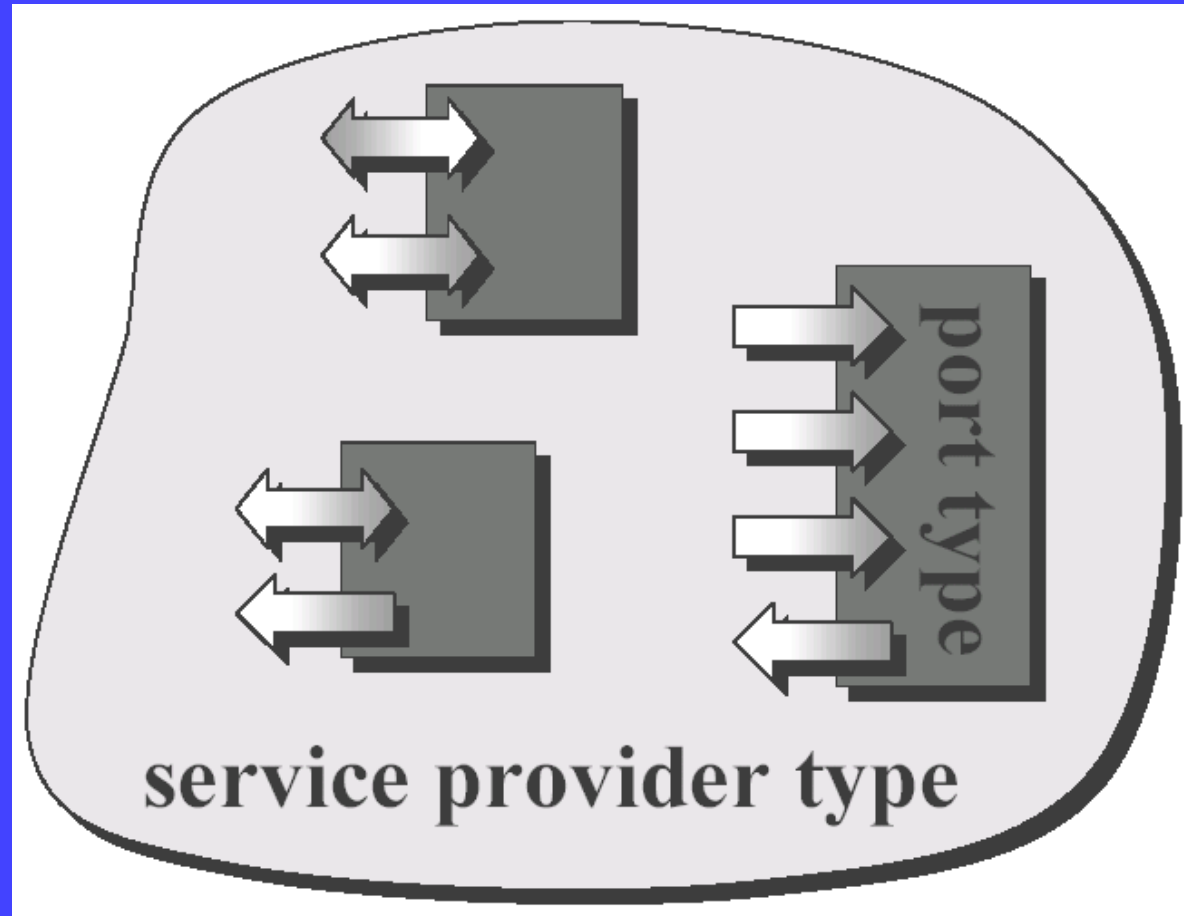
Beispiel-Ablaufmodell



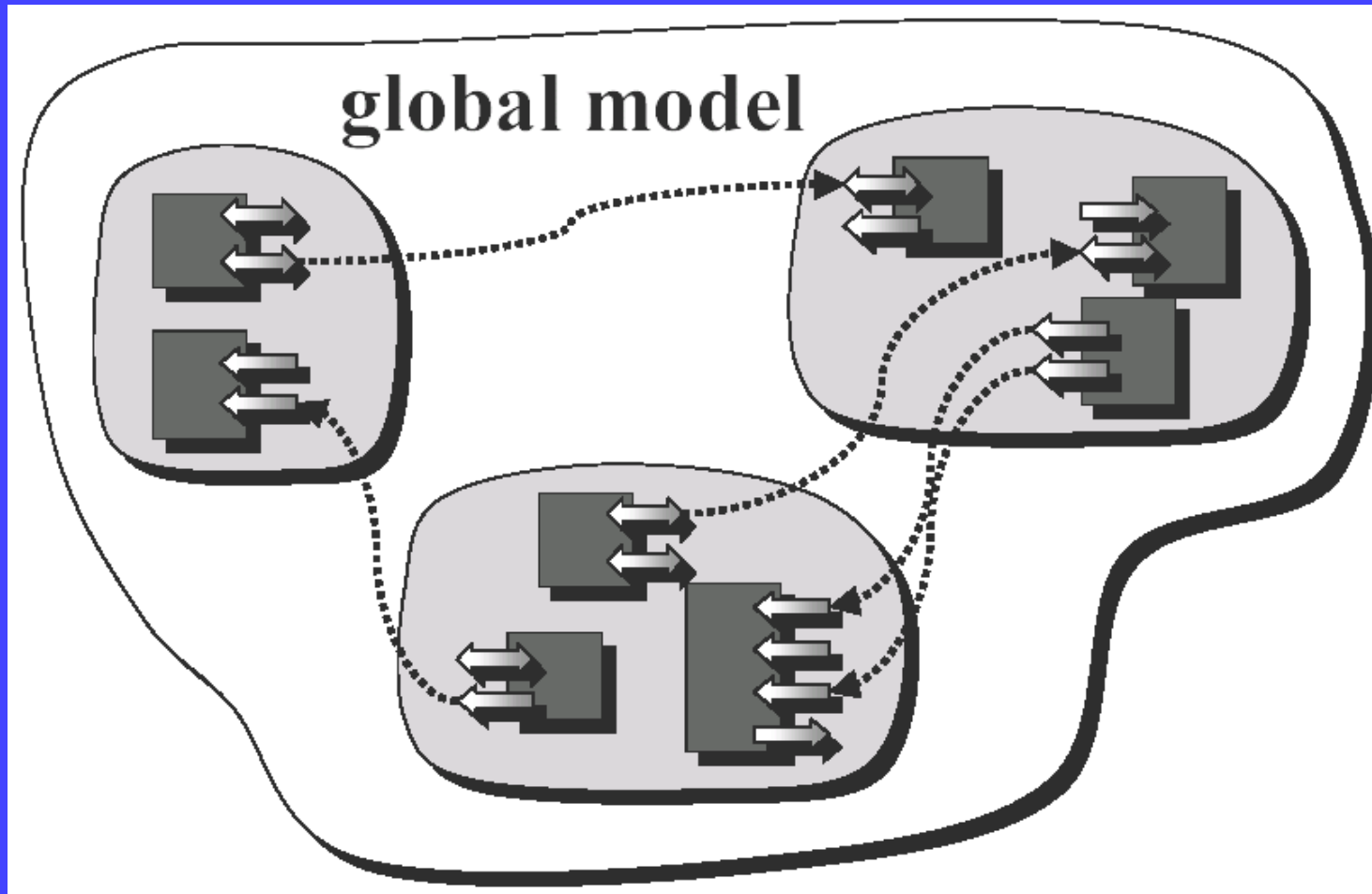
Beispiele für Plug Links



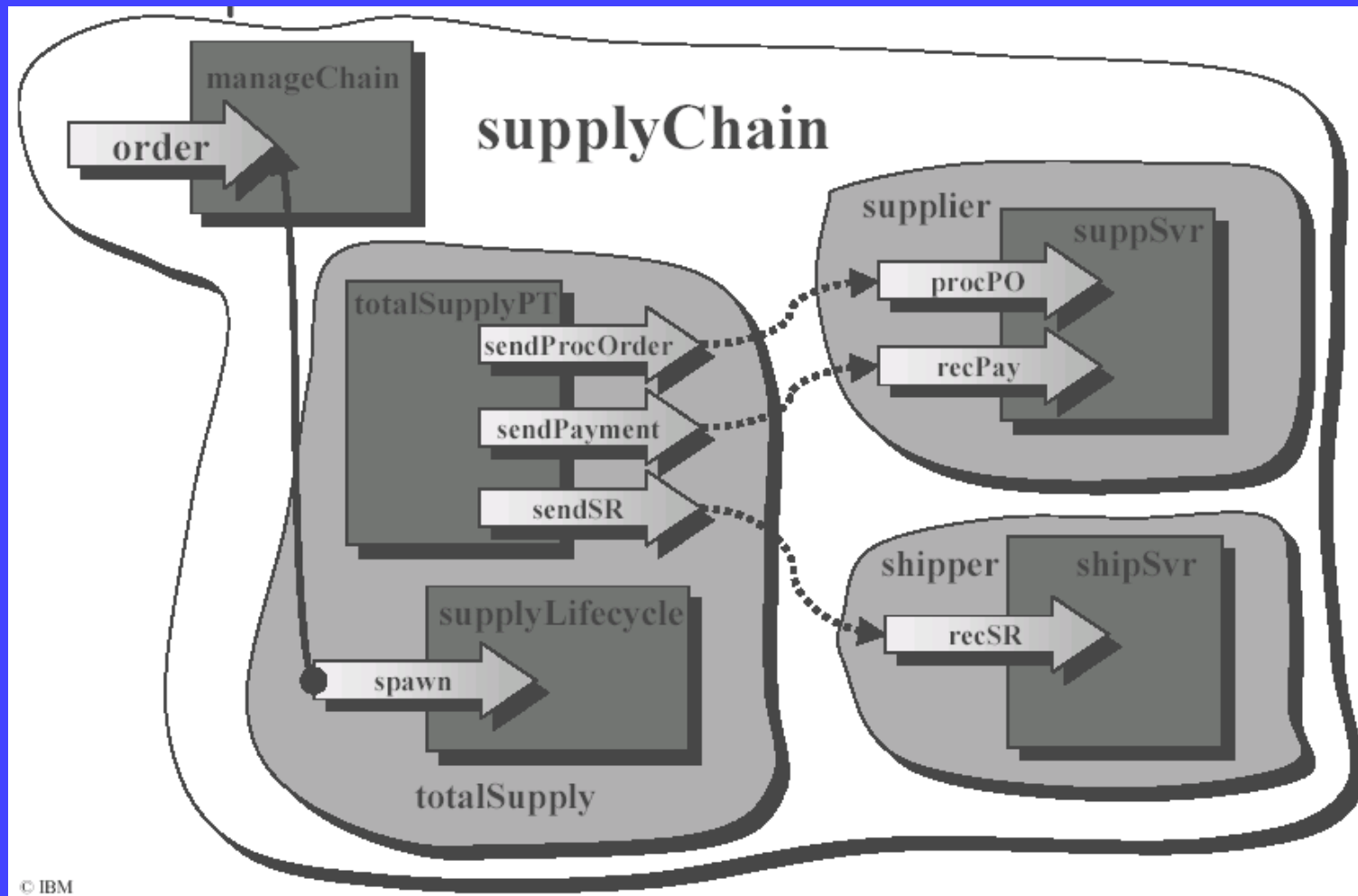
Service Provider Typen



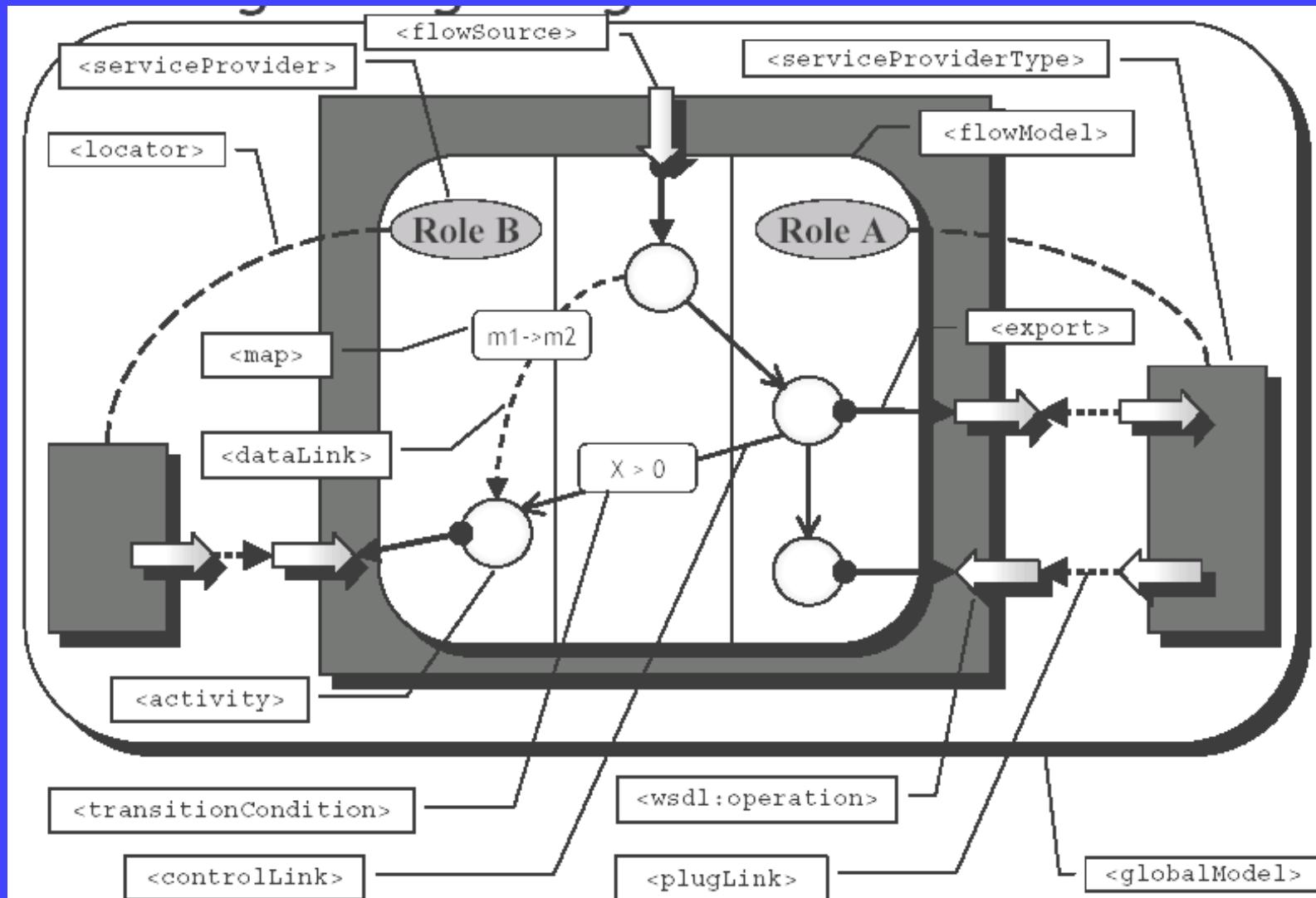
Globales Modell



Beispiel eines globalen Modells



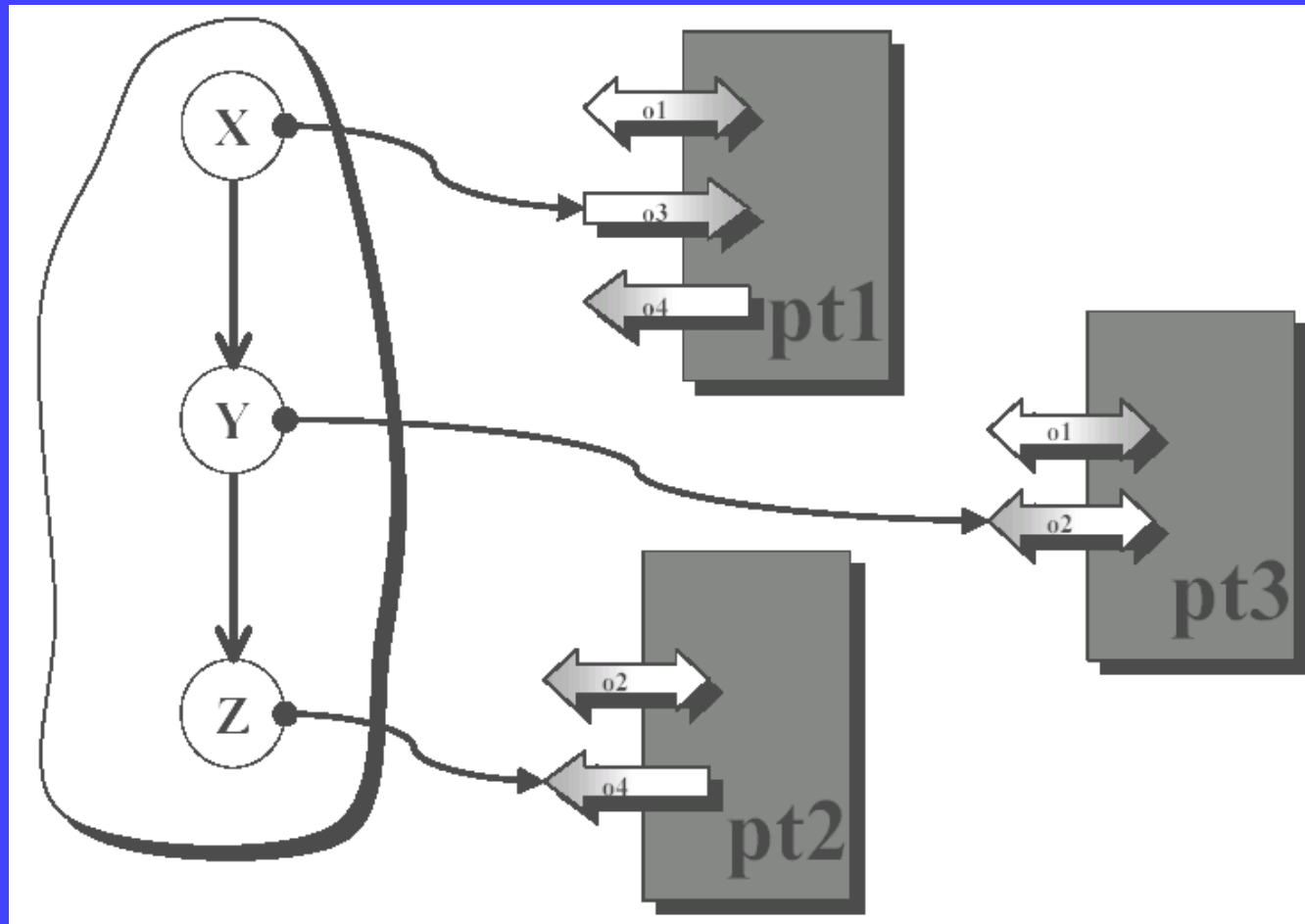
Zusammenbringen dieser Beispiele: WSFL



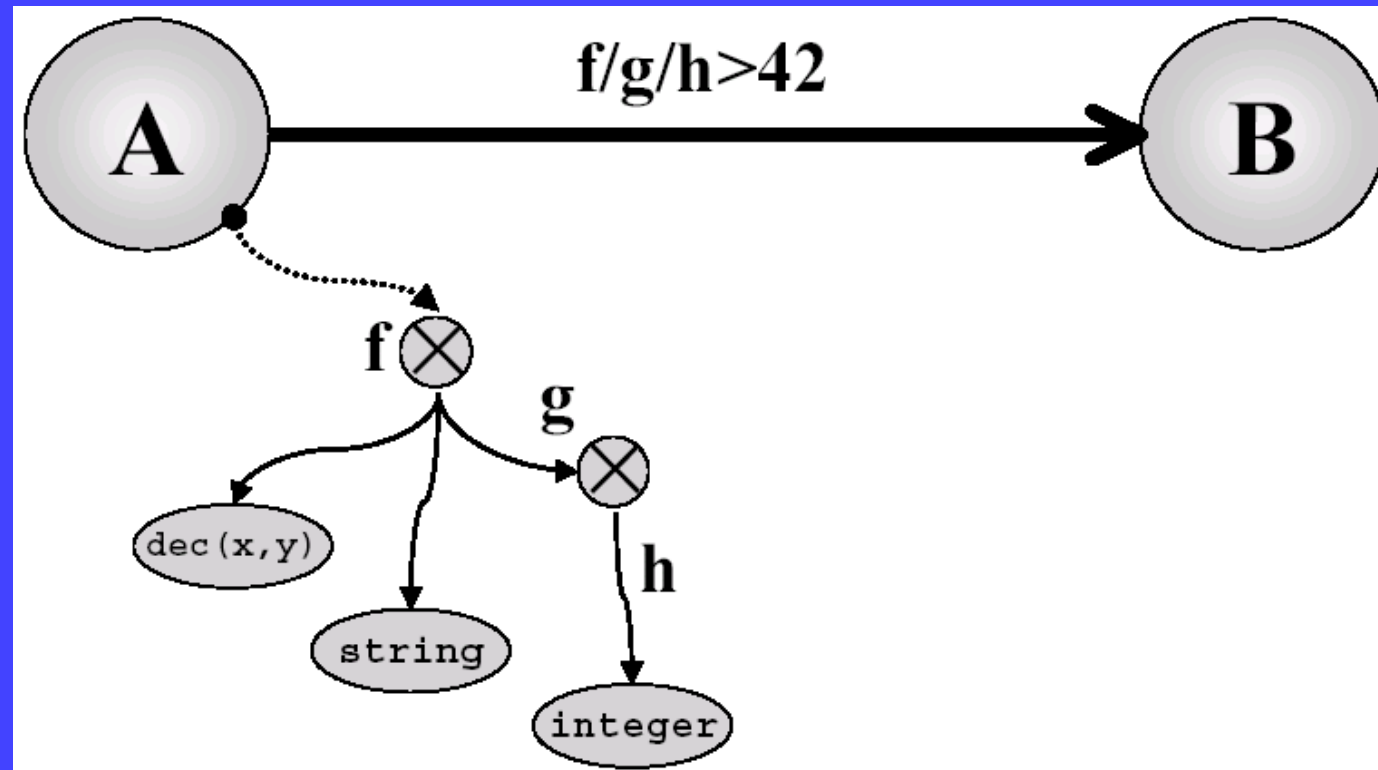
Einige Details über WSFL- Konzepte



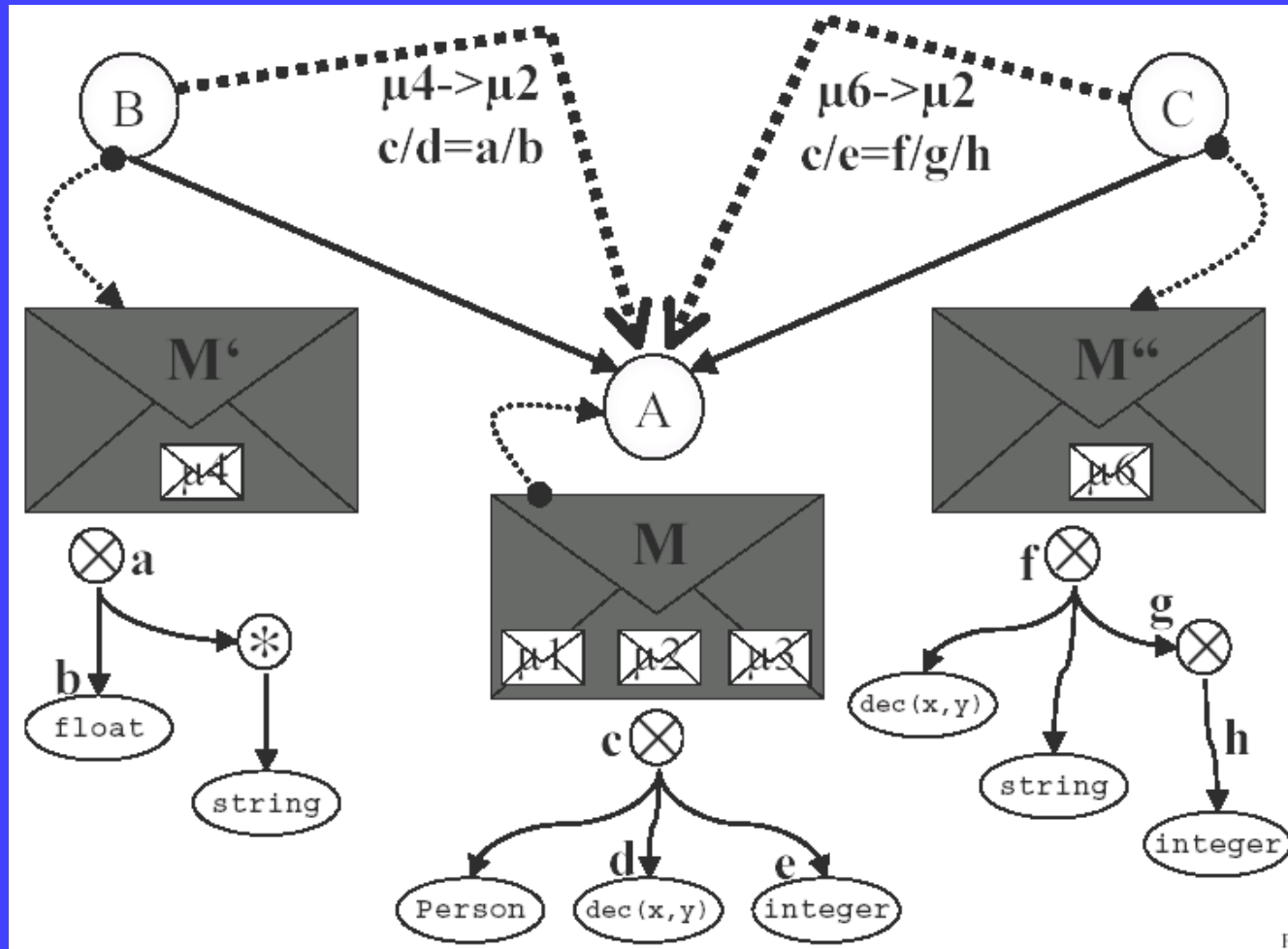
Nutzen von Port-Typen in Geschäftsprozessen



Control Links



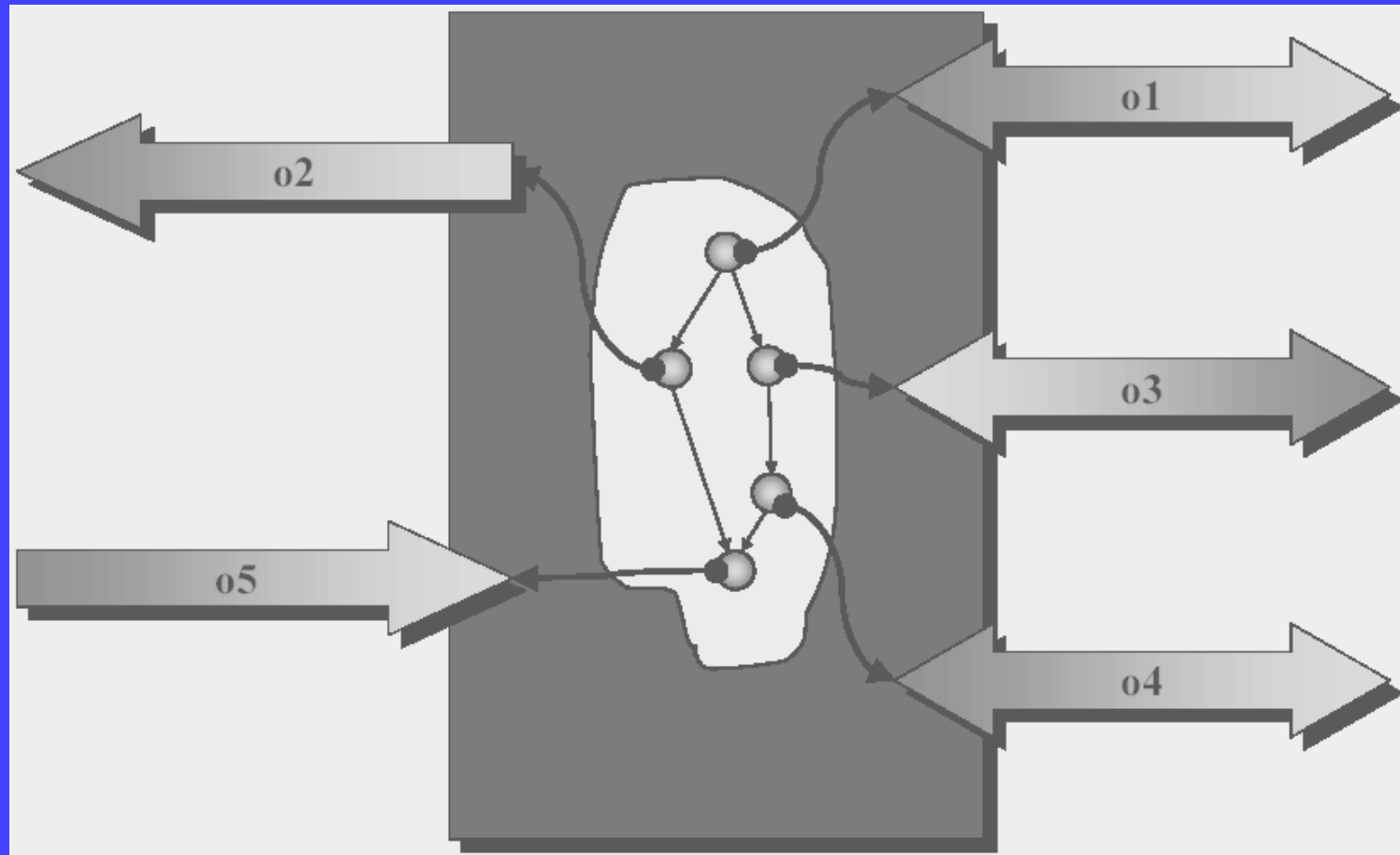
Daten Links



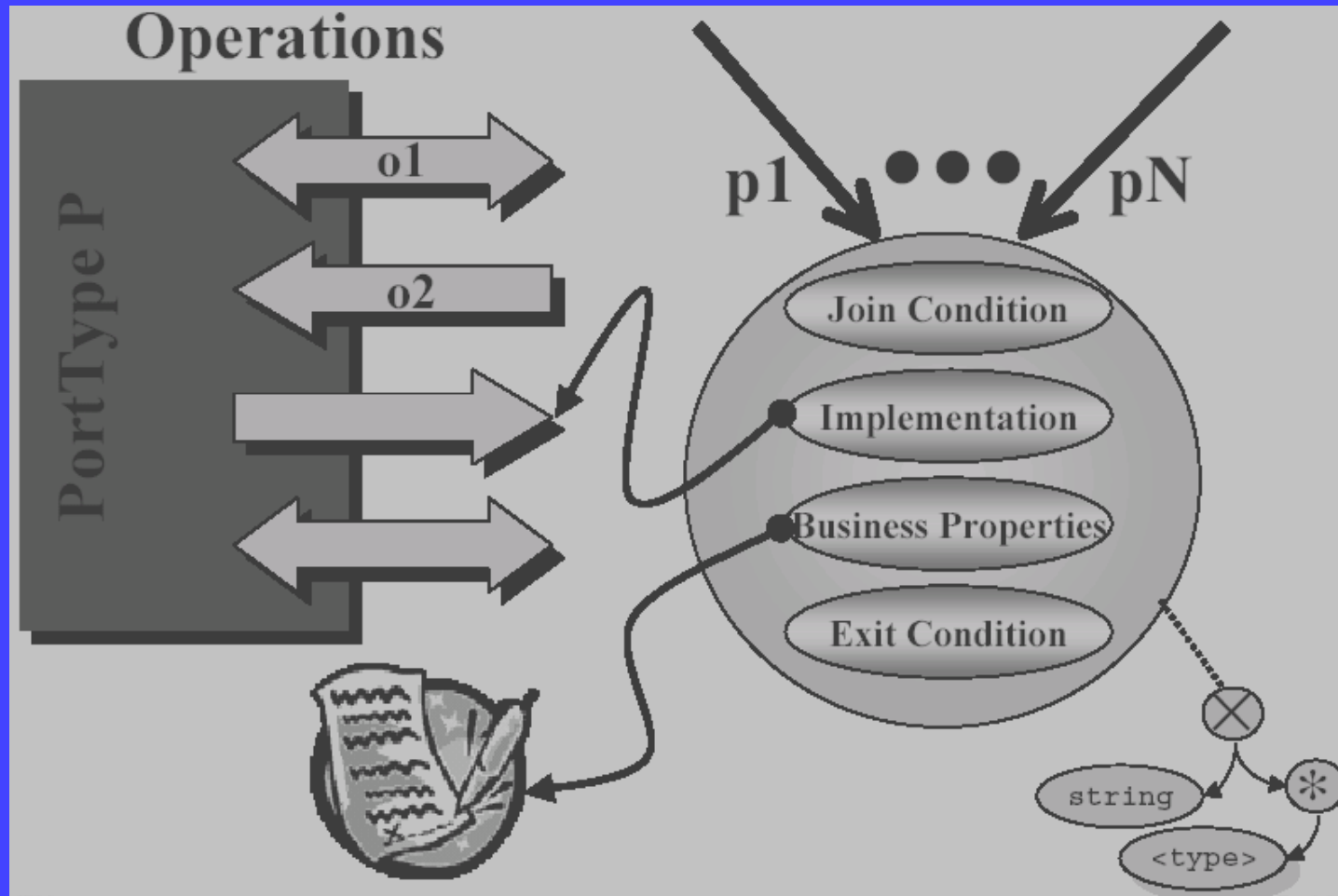
Fr



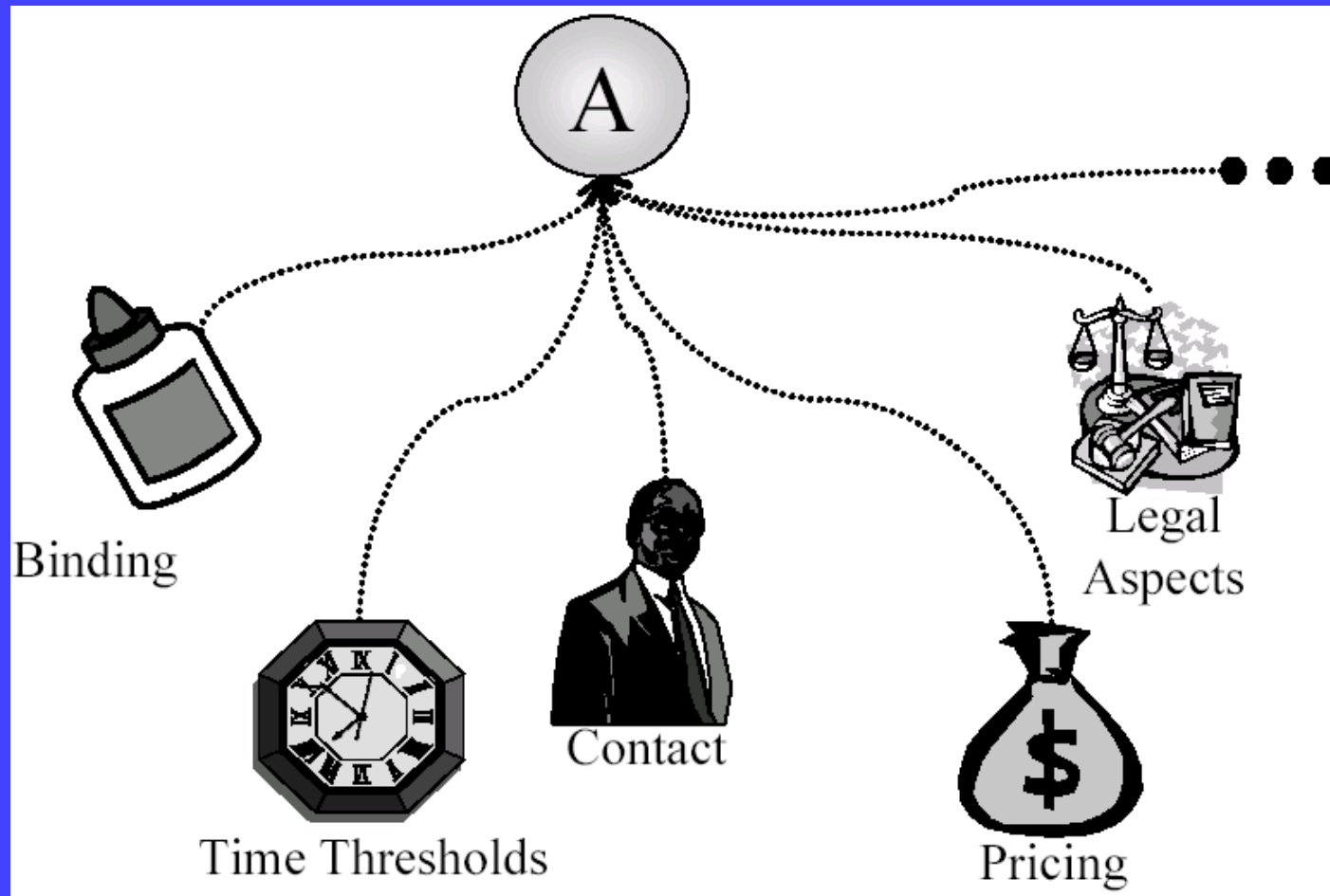
Einschränkung der Aufrufreihenfolge



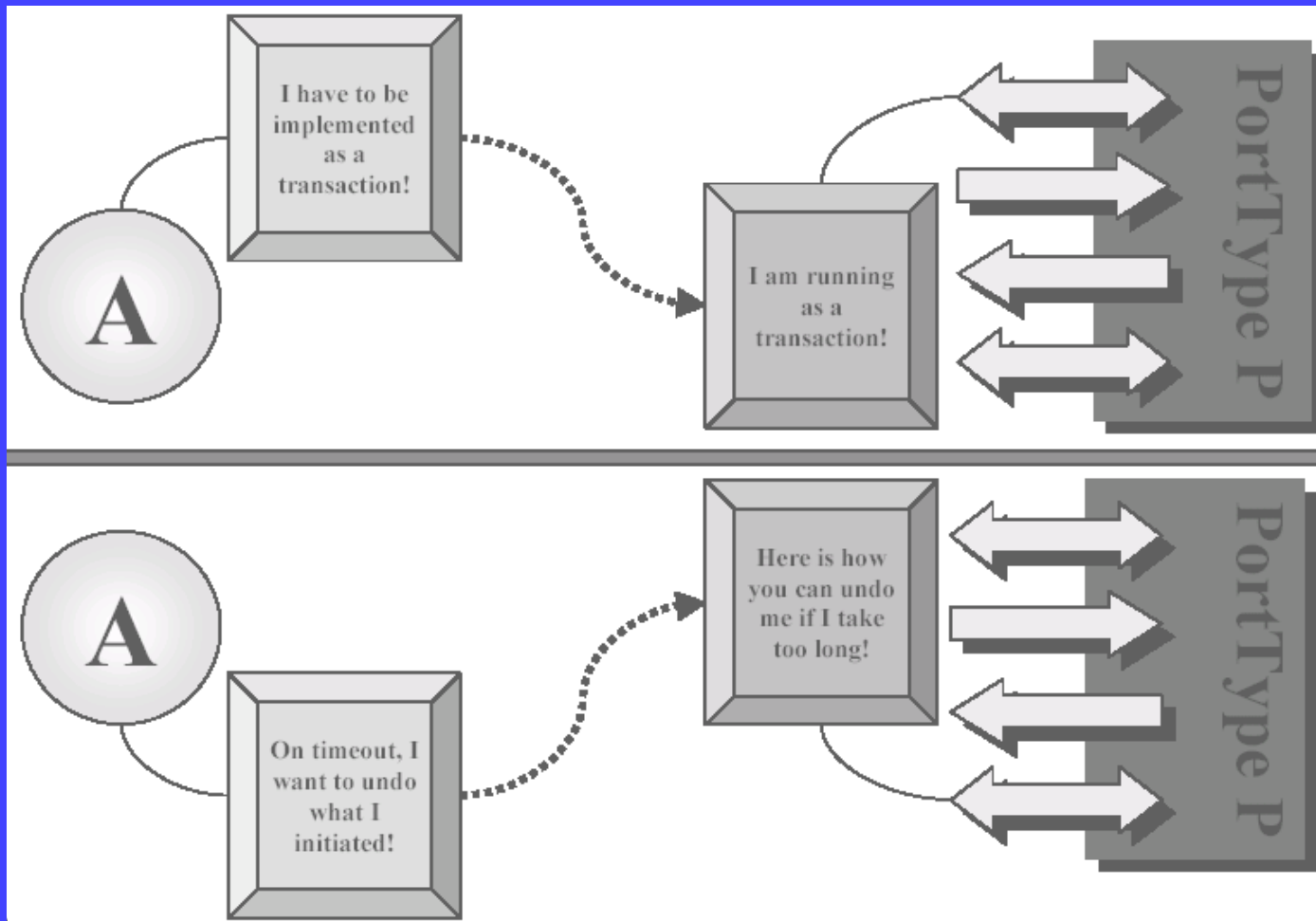
Feine Aktivitätenstruktur



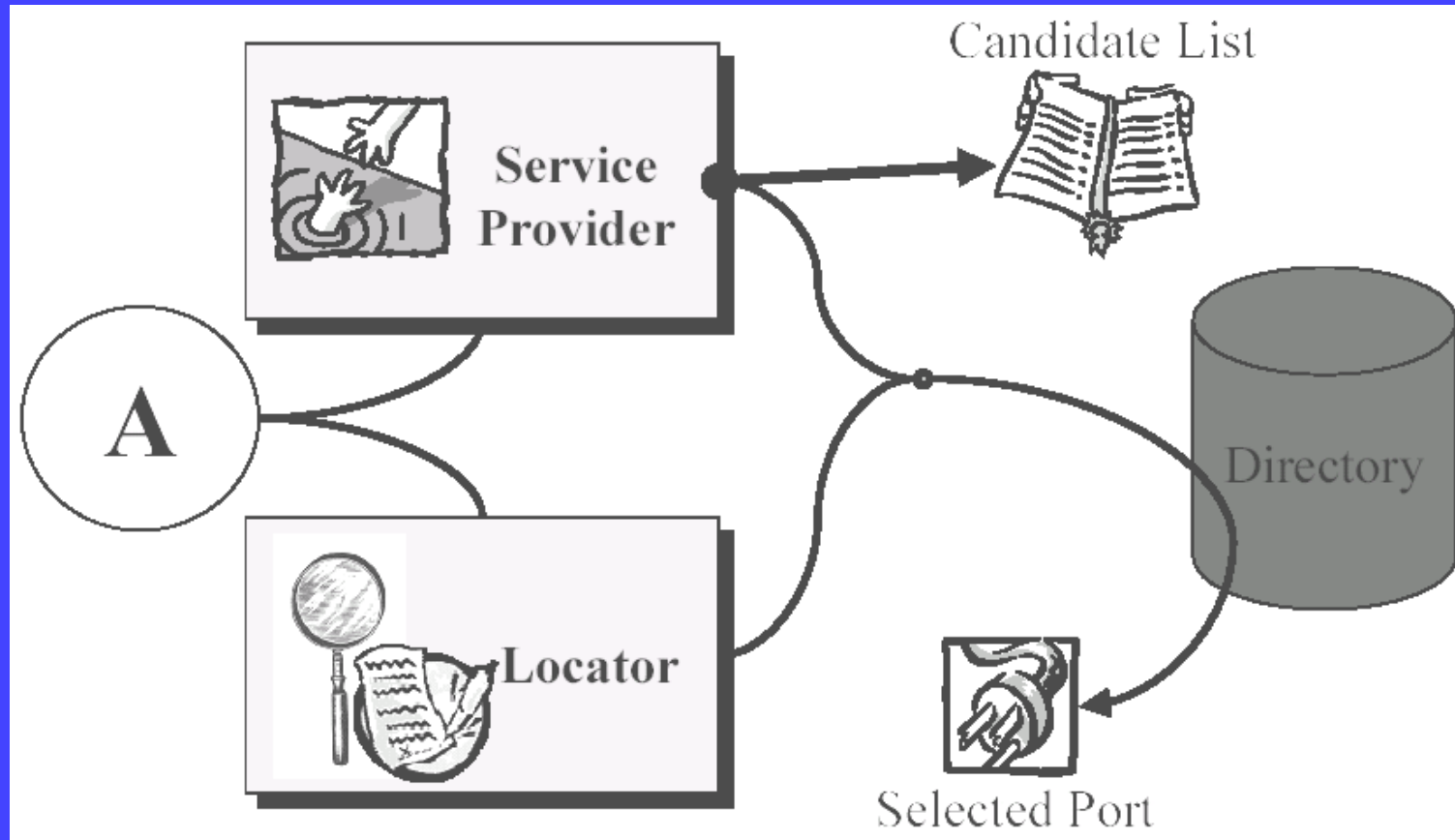
Geschäftseigenschaften einer Aktivität



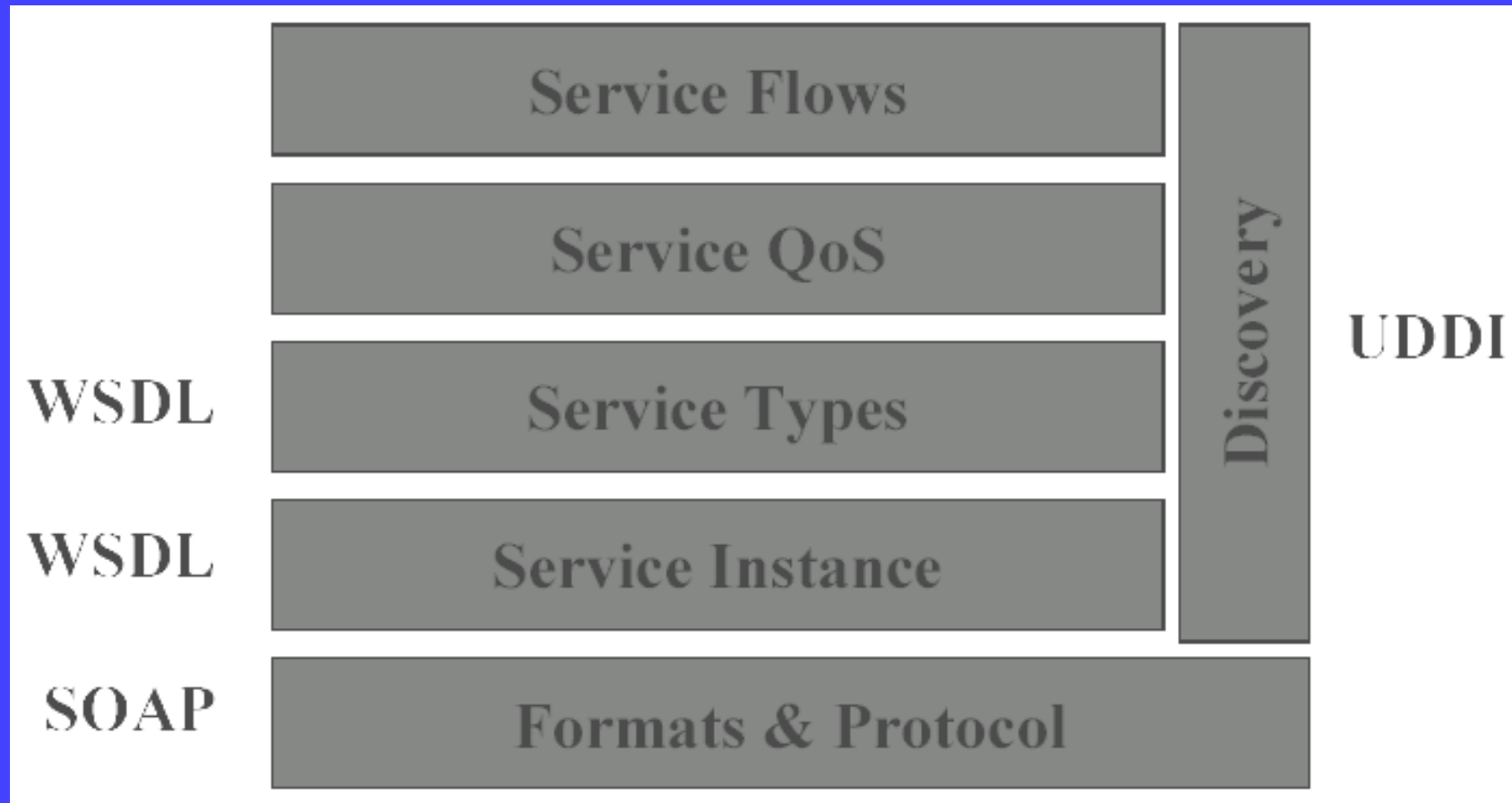
Quality Of Services



passende Endpunkte



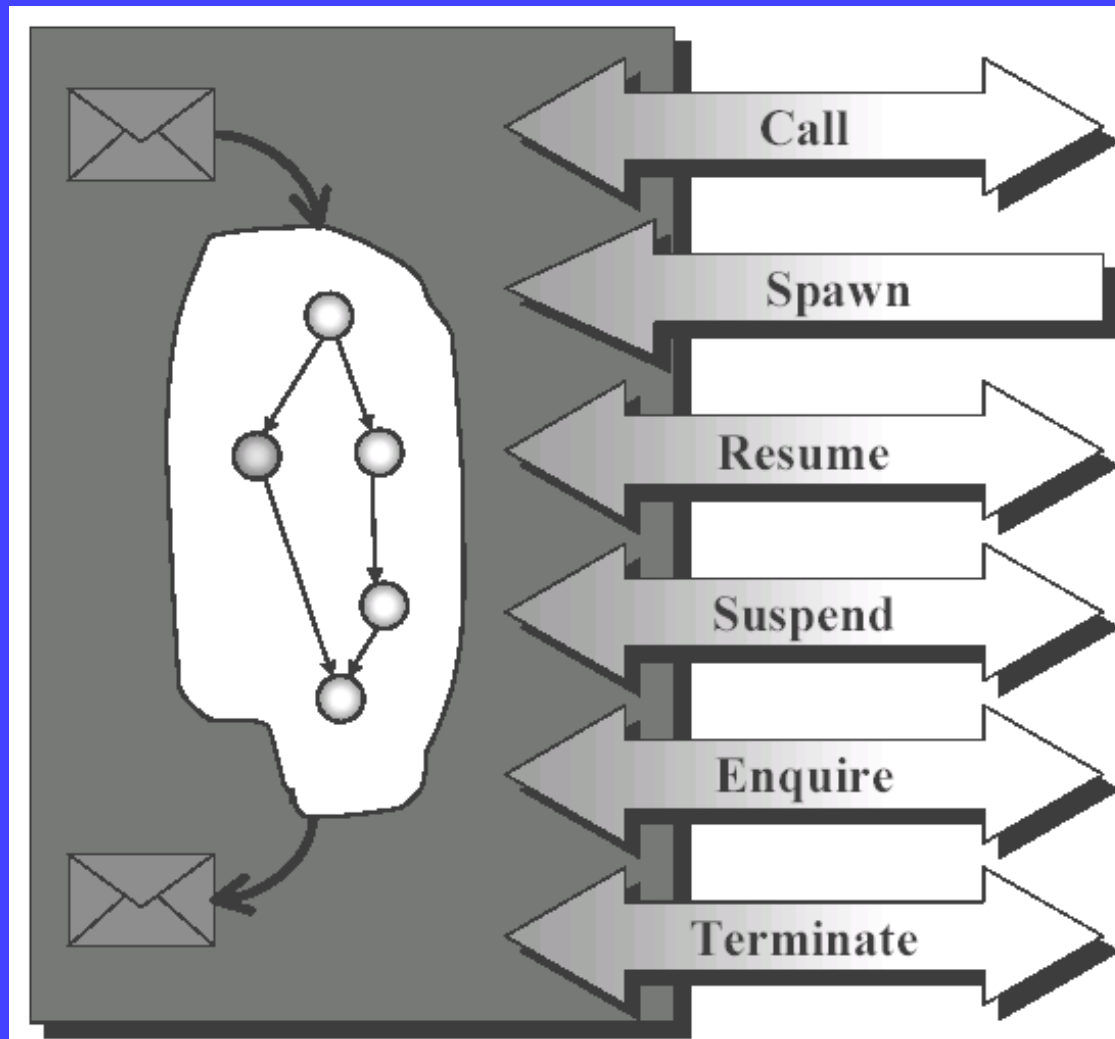
Der Web Services Stack



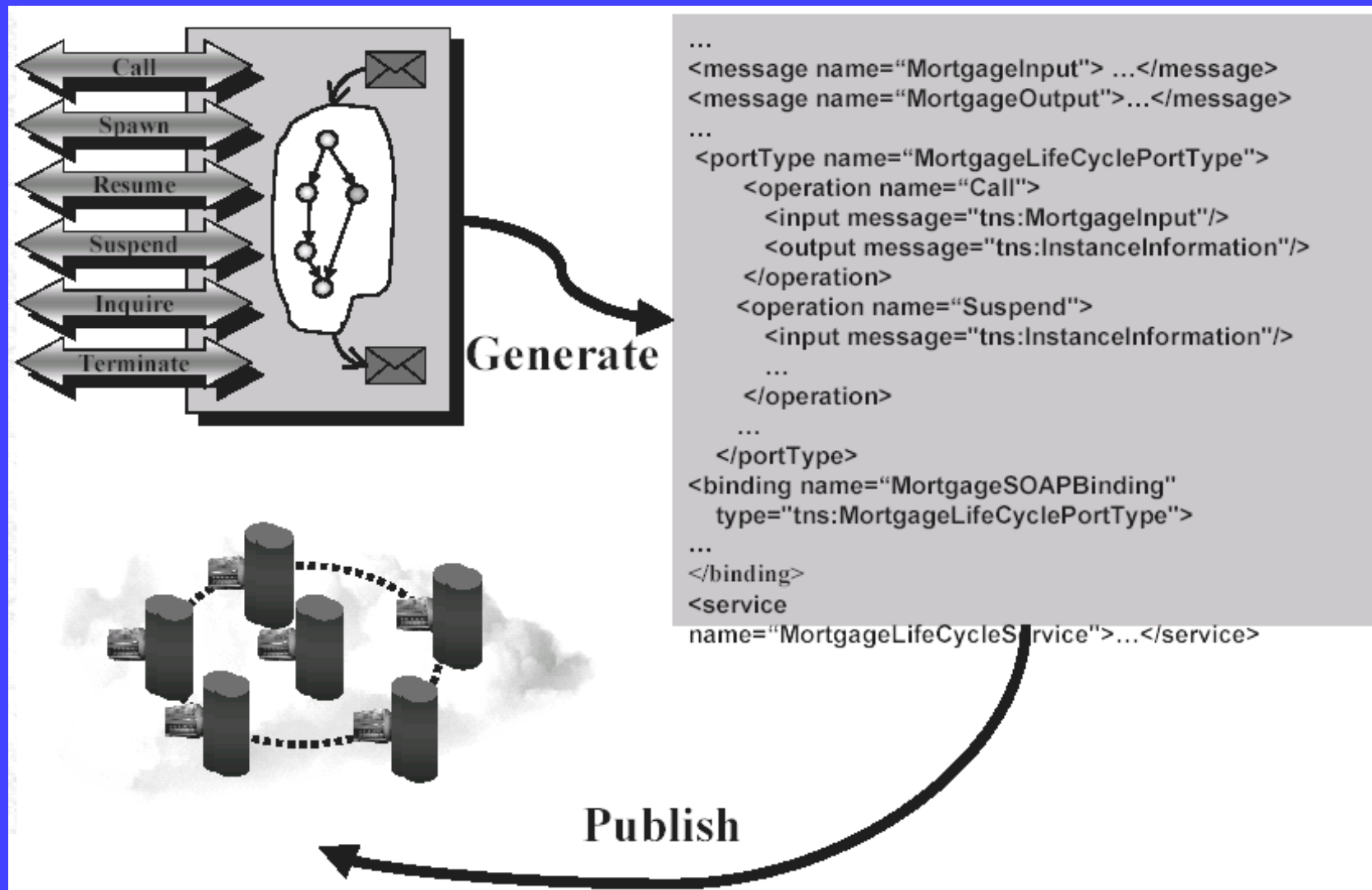
Workflows als Web Services



Workflows als Web Services



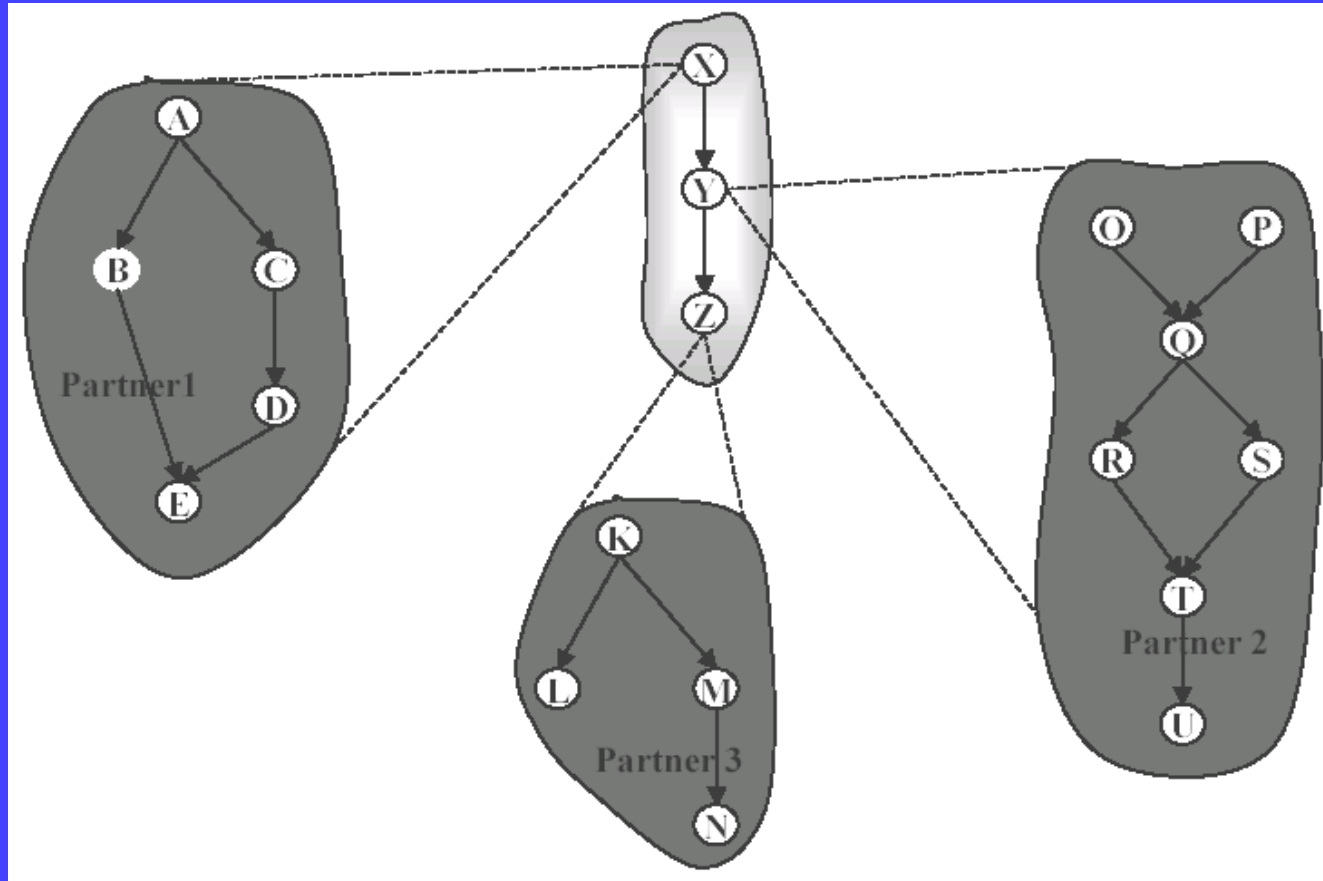
Workflow Lifecycle Services veröffentlichen



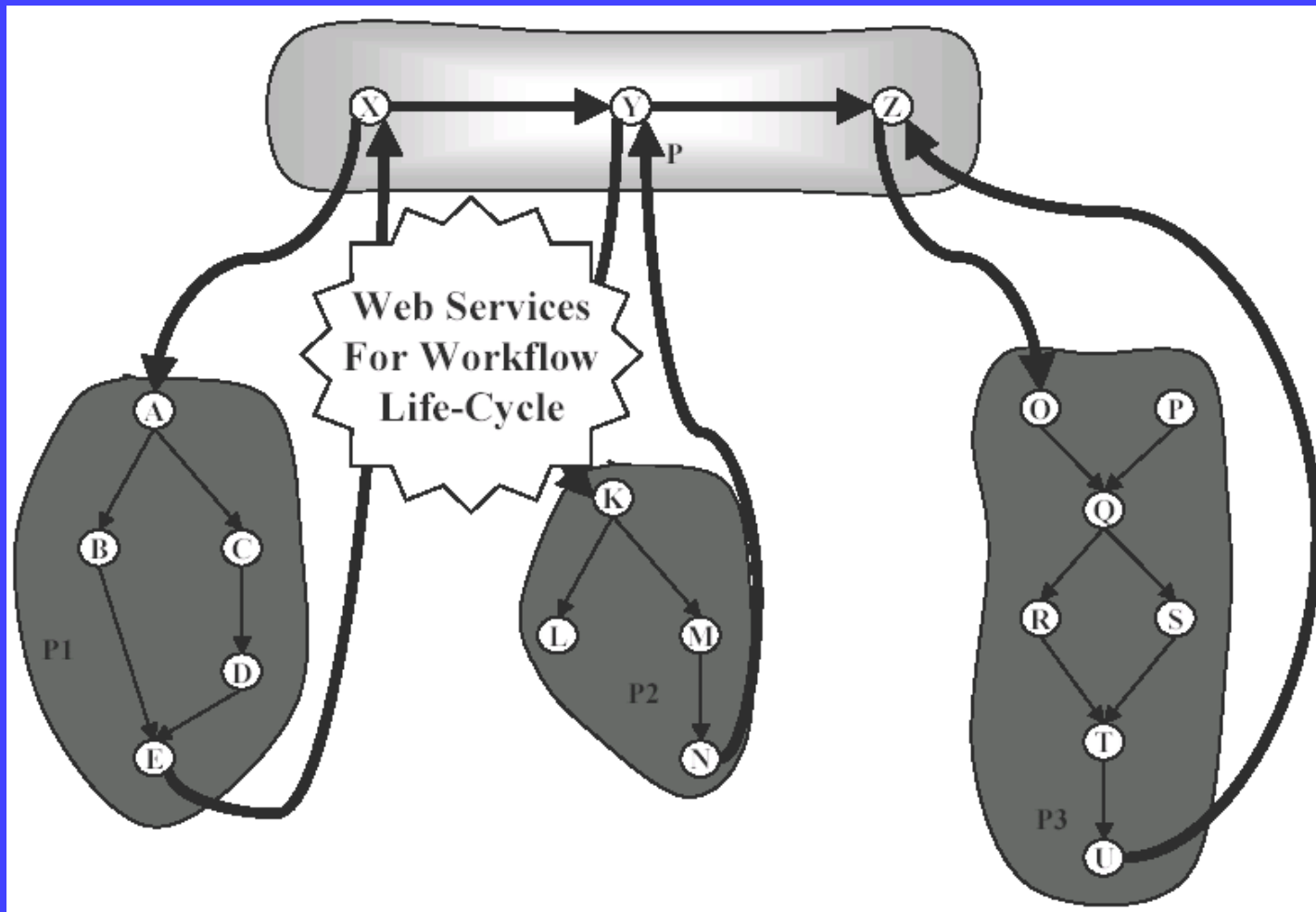
Web Services Choreographie: hierarchische Abläufe



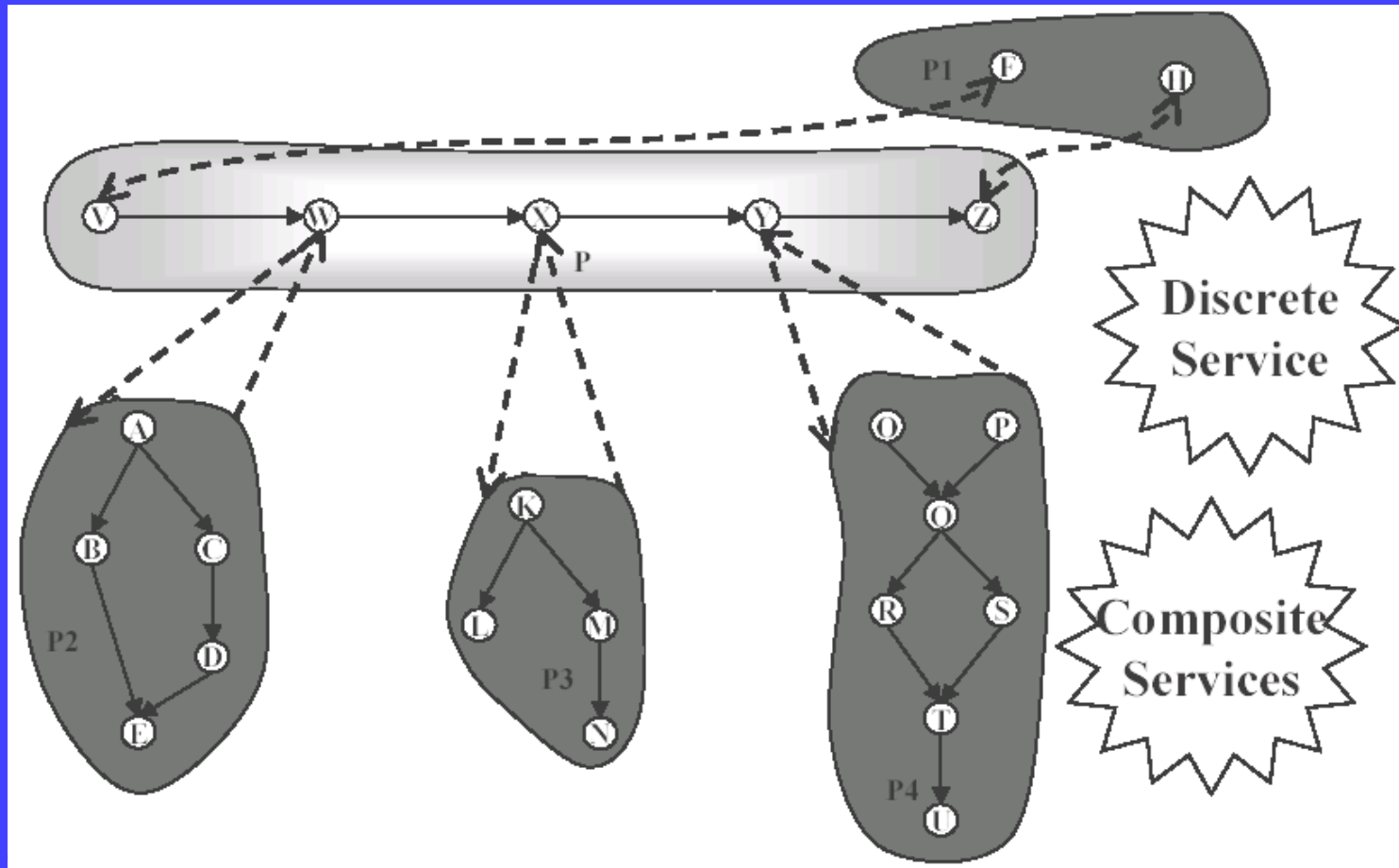
Multi-laterale Vereinbarung



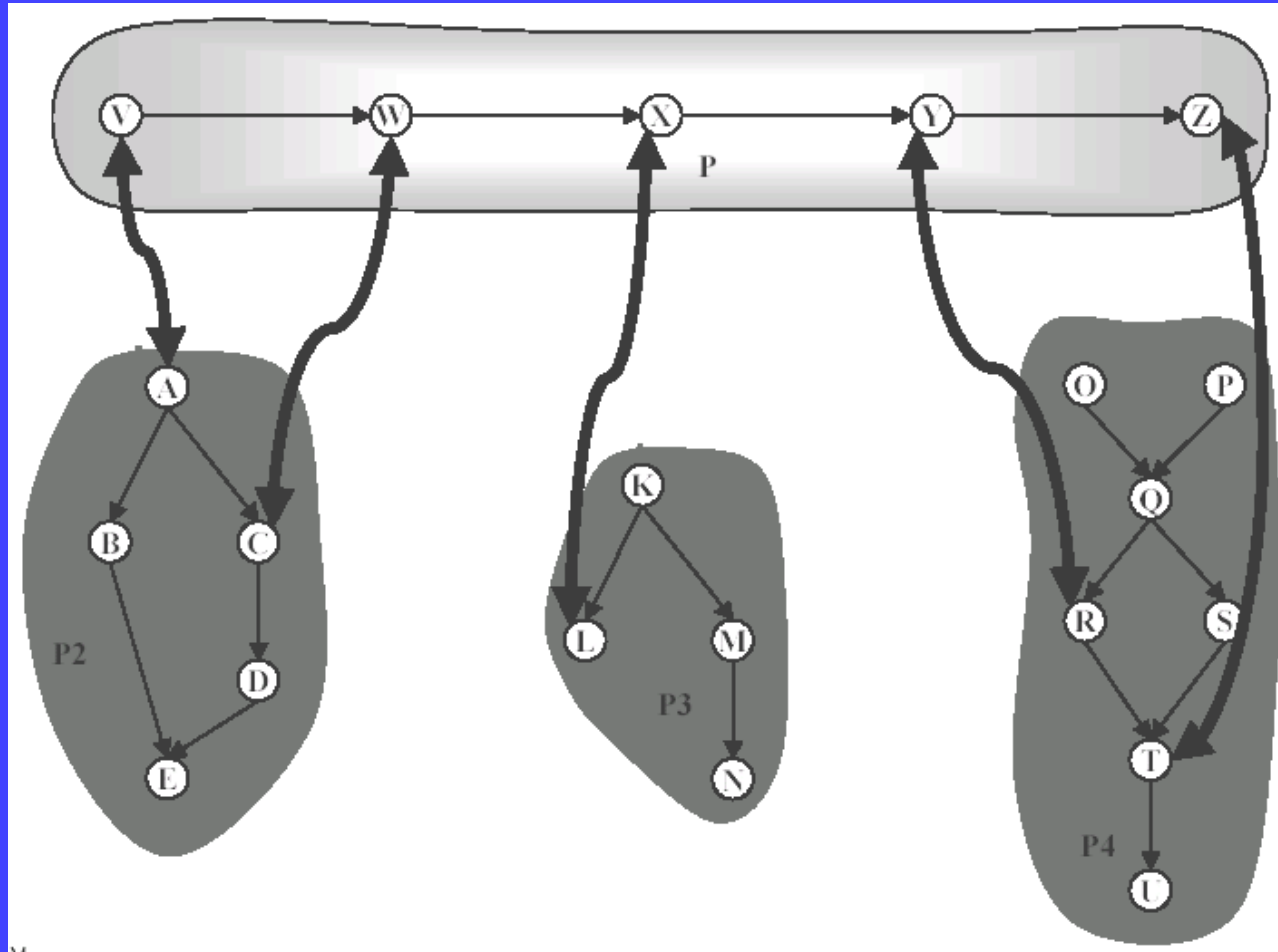
Öffentlicher Workflow



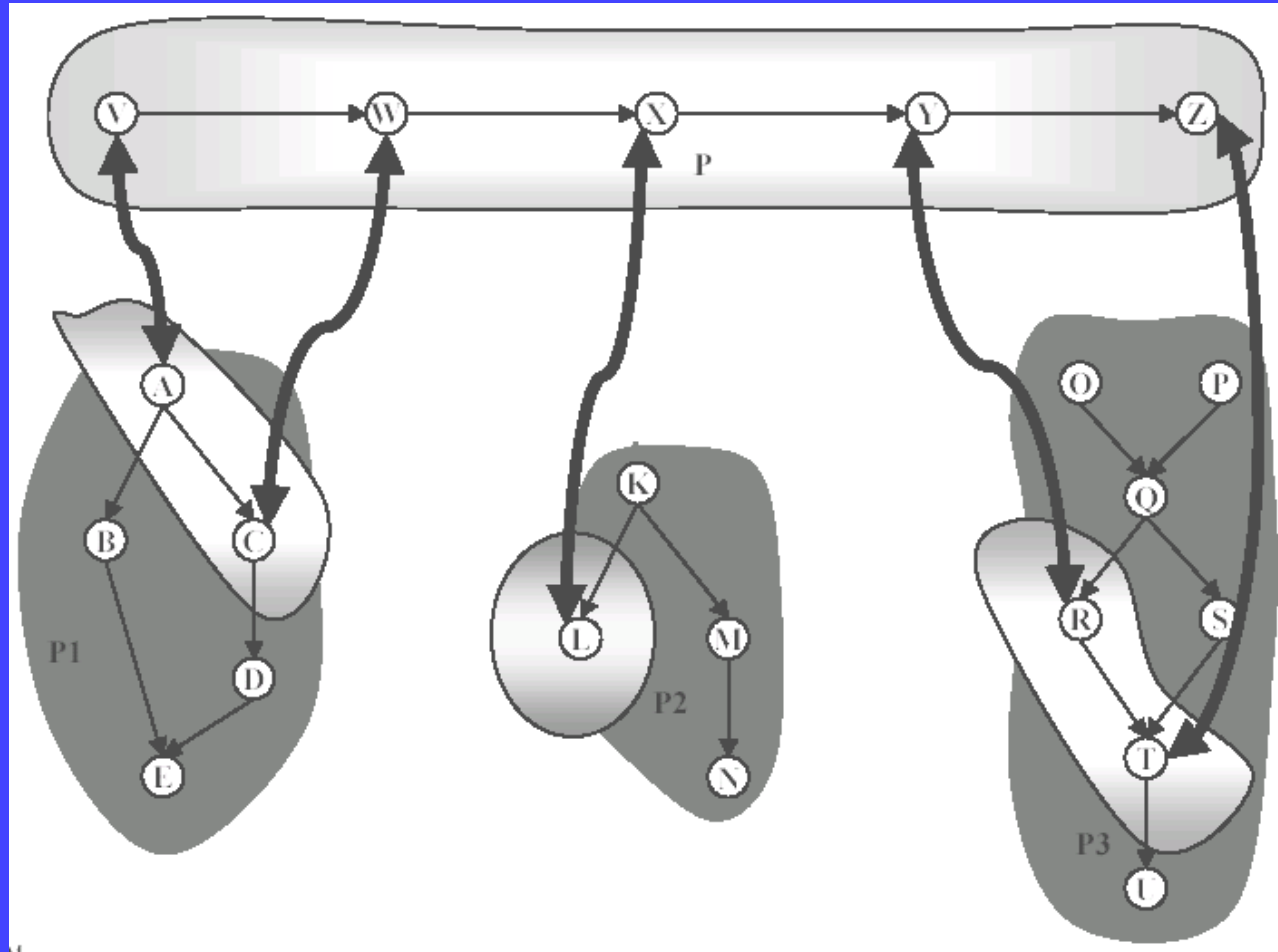
Hierarchisches B2B-Szenario



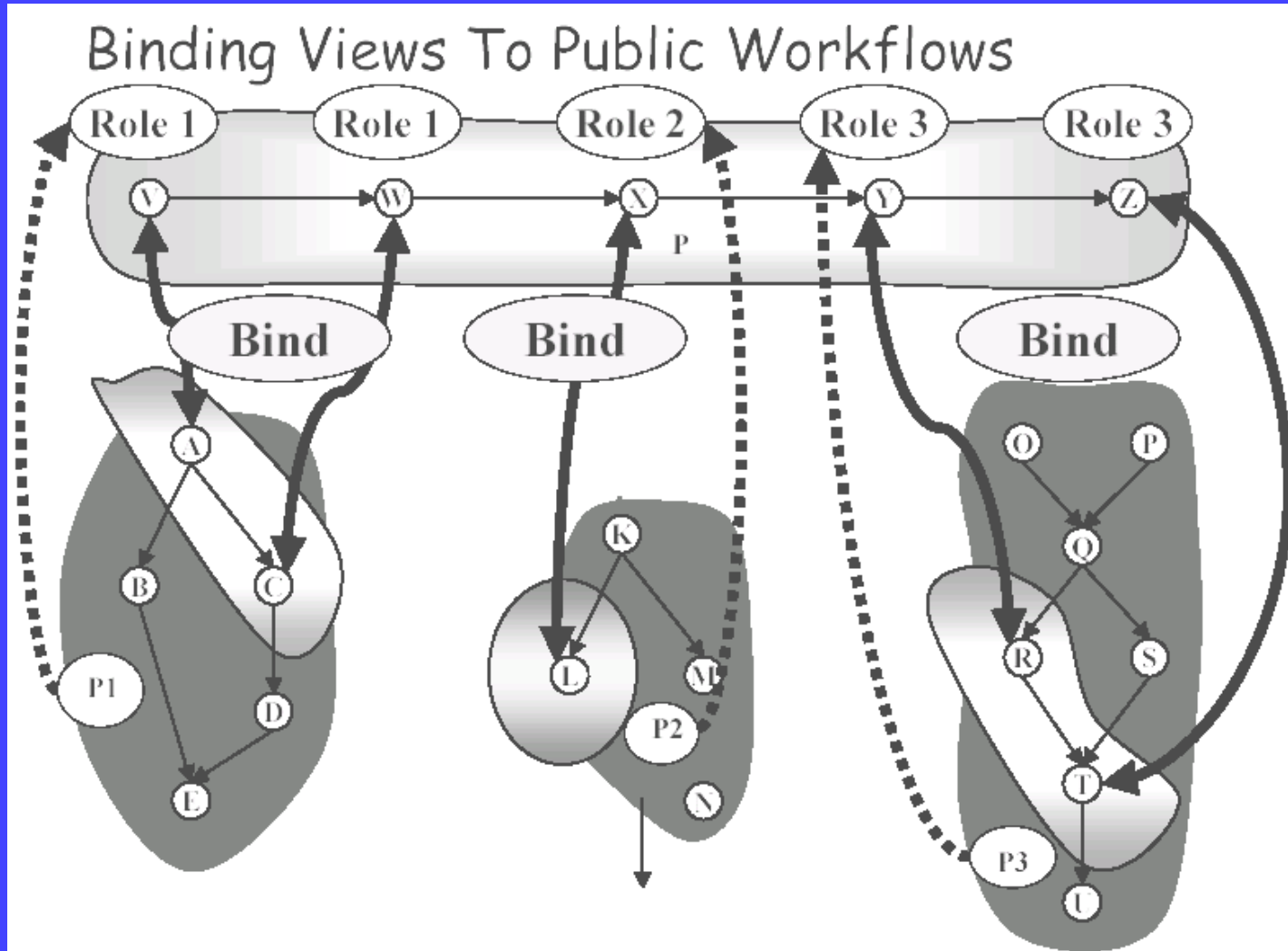
“Einzelne” Teilnahme an öffentlichen Workflows



Öffentliche Sichten auf privaten Workflows



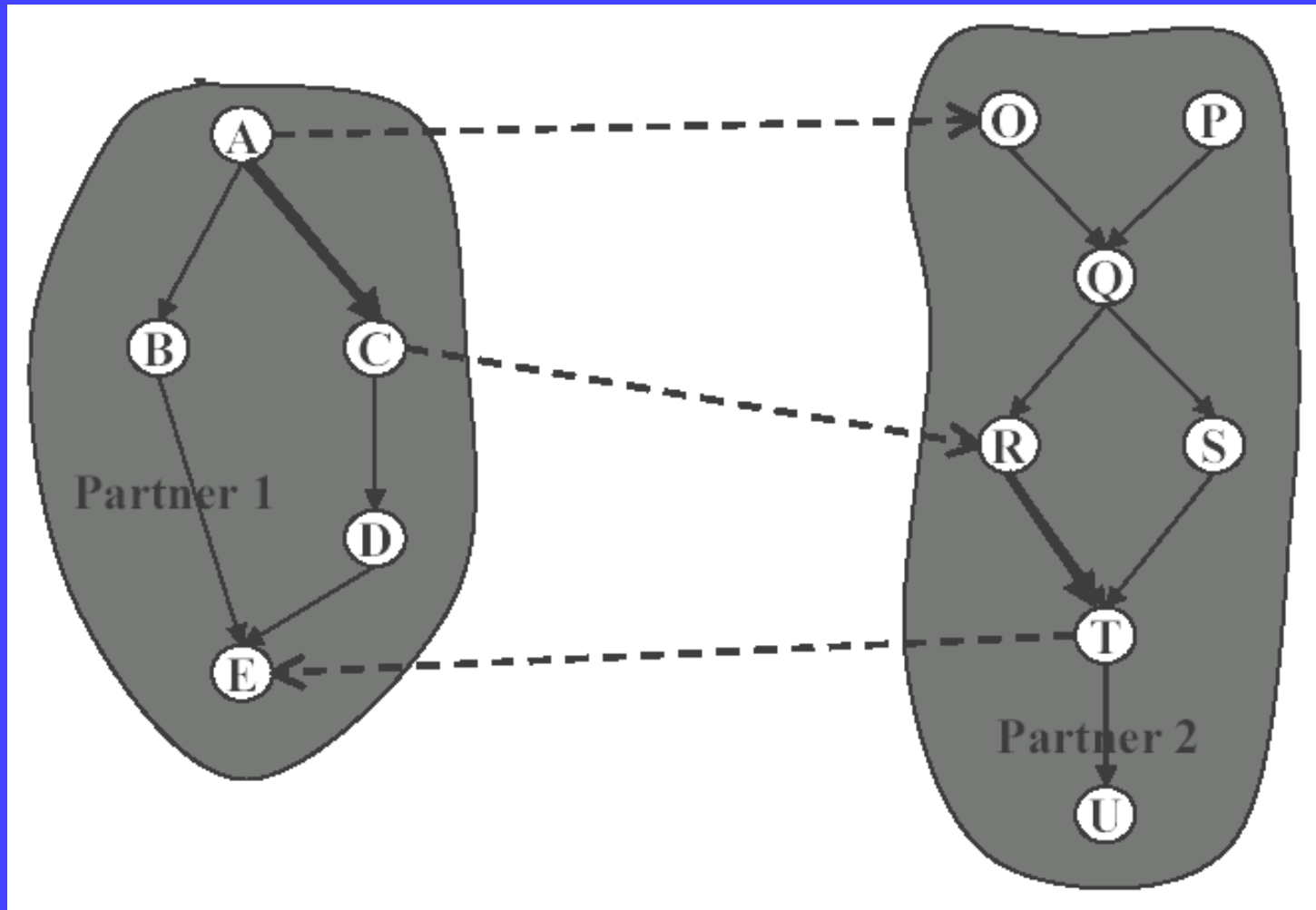
„Binding“ von Sichten an öffentliche Workflows



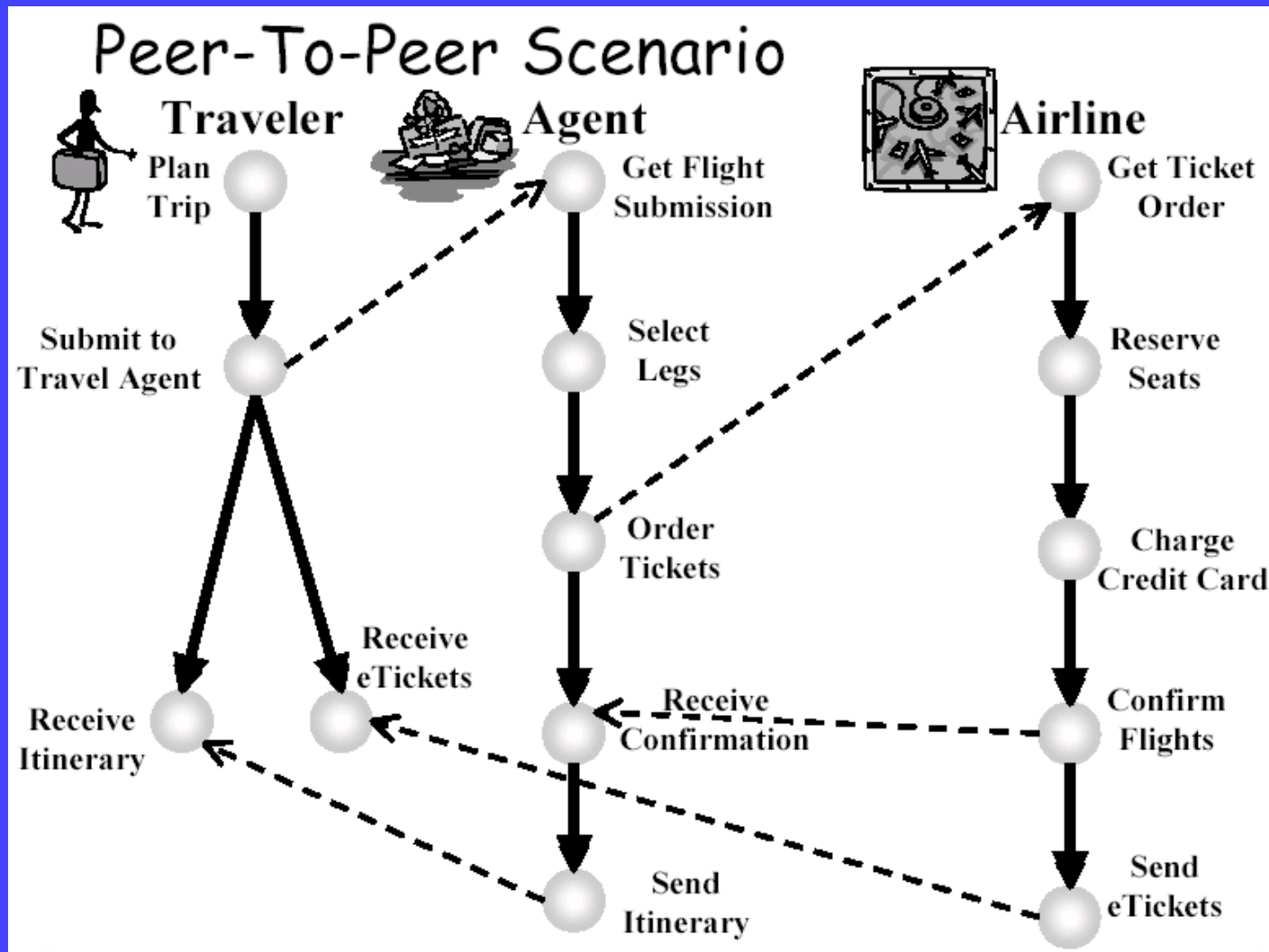
Web Services Choreographie: Peer-To-Peer Abläufe



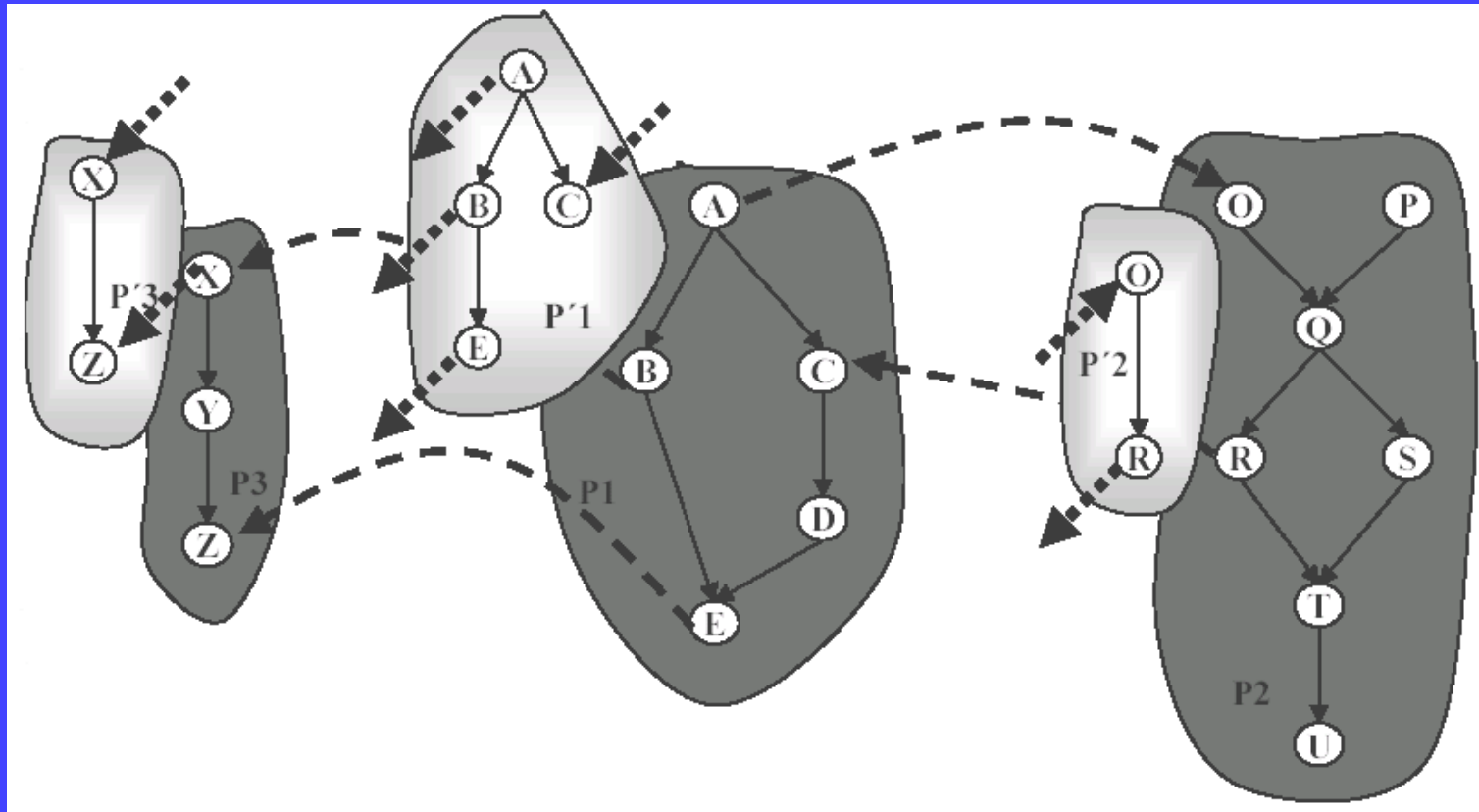
Peer-To-Peer B2B-Szenario



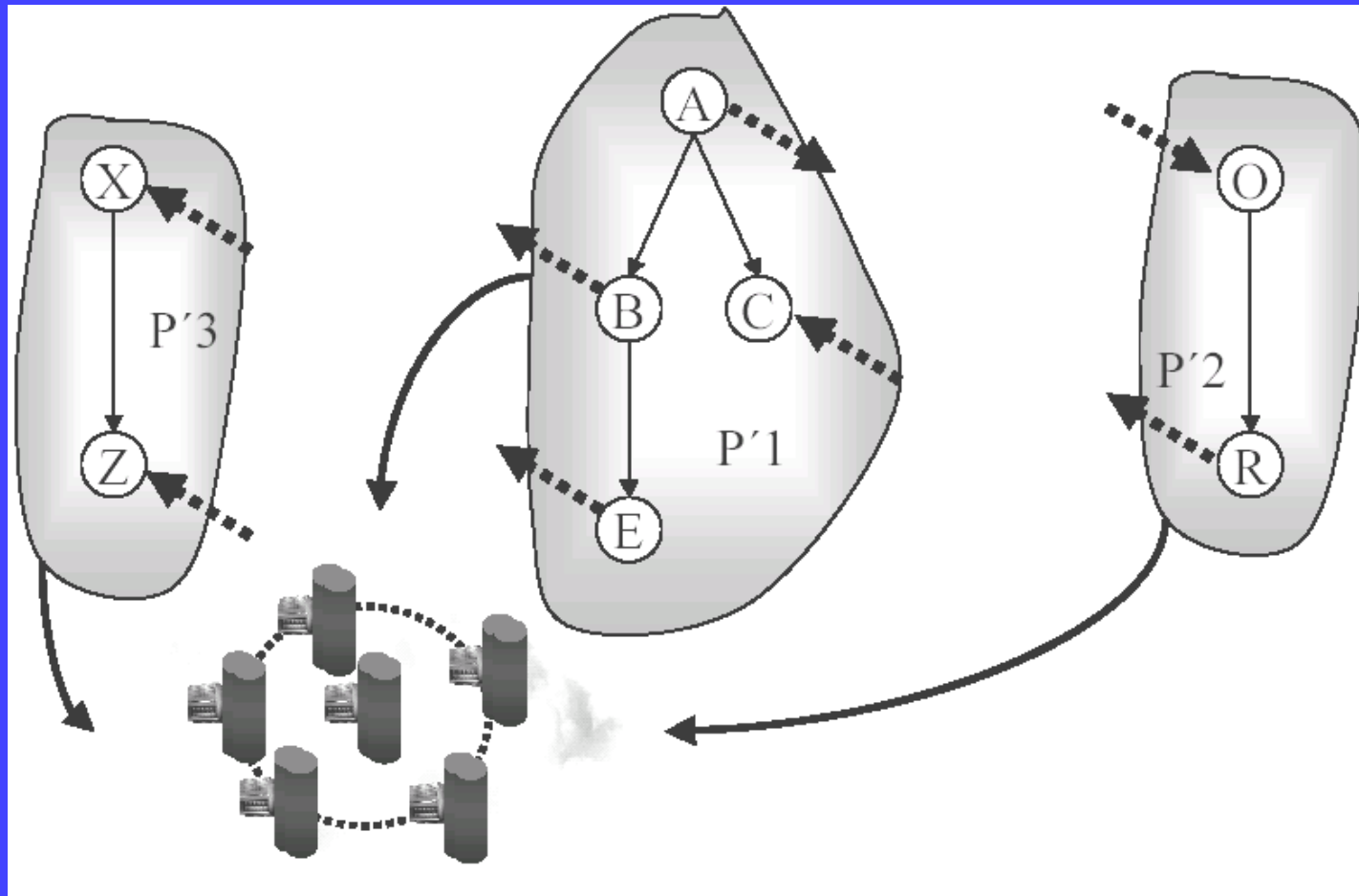
Peer-To-Peer Szenario



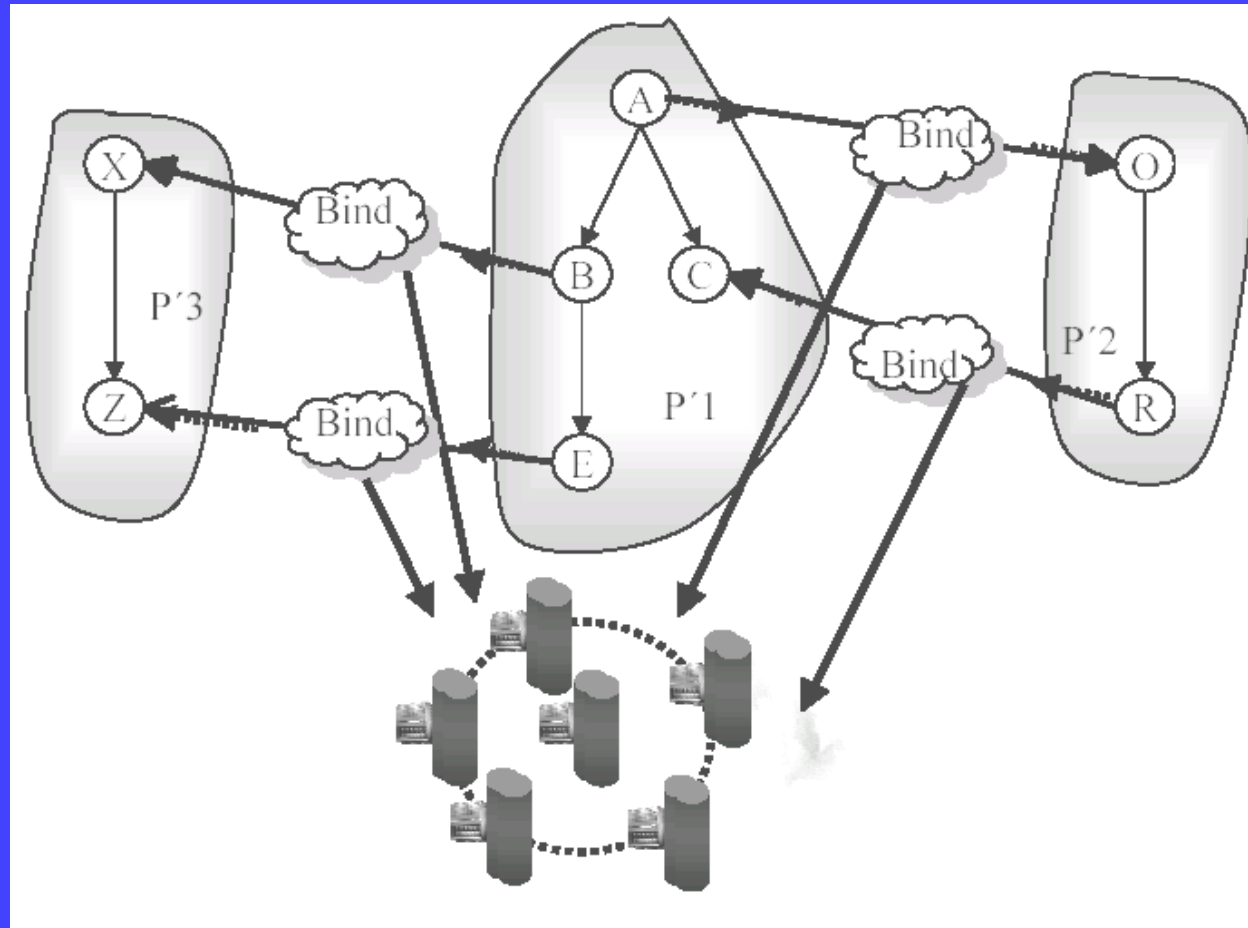
Sichtendefinition auf Workflows...



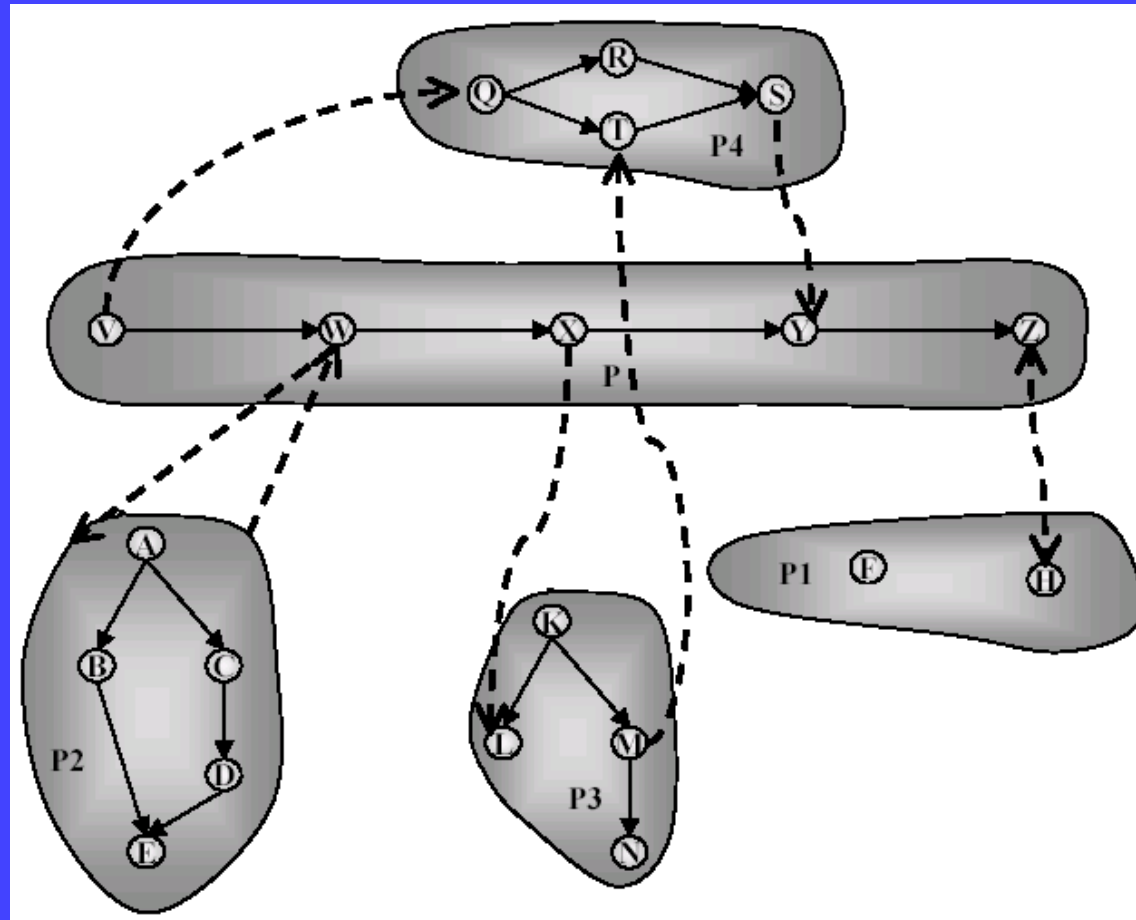
...und veröffentlichen der Sichten als Web Services



Dynamisches „Binding“ zur Laufzeit



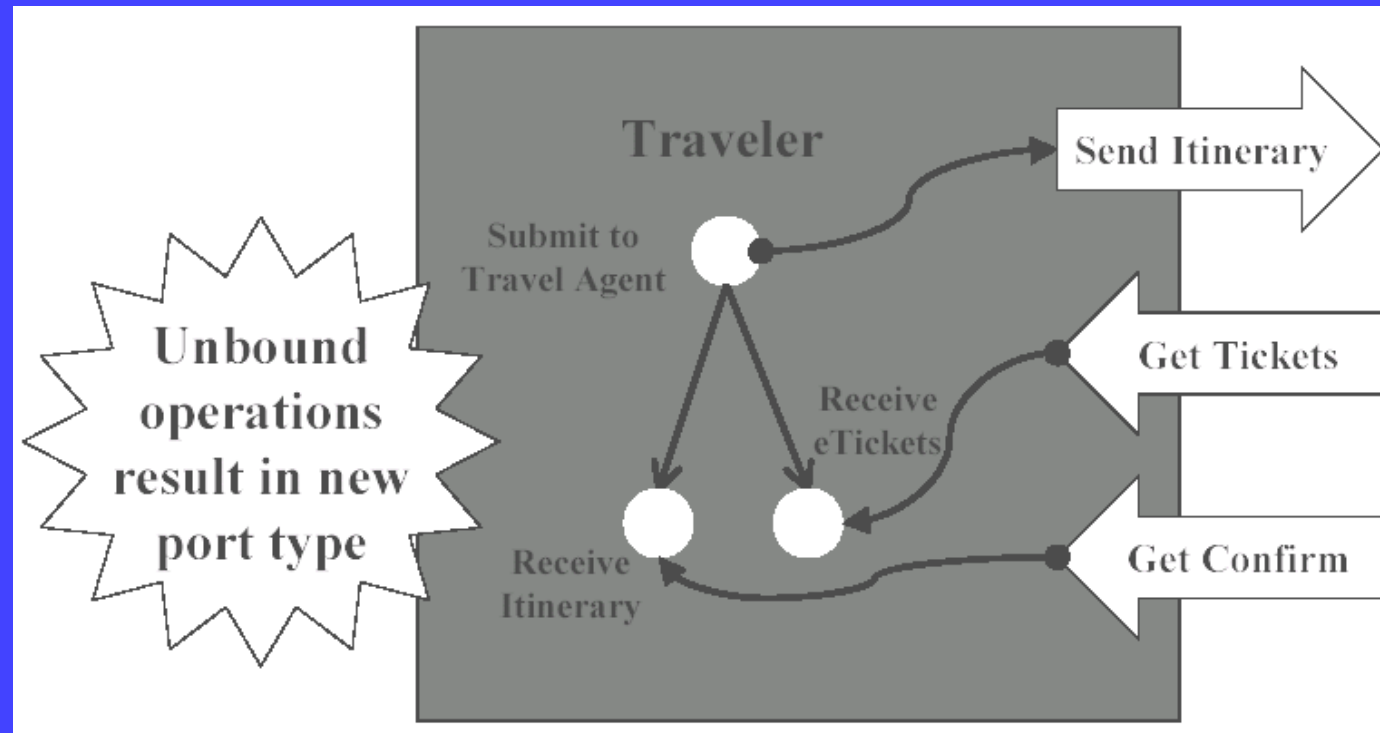
und schließlich: ein gemischtes Szenario



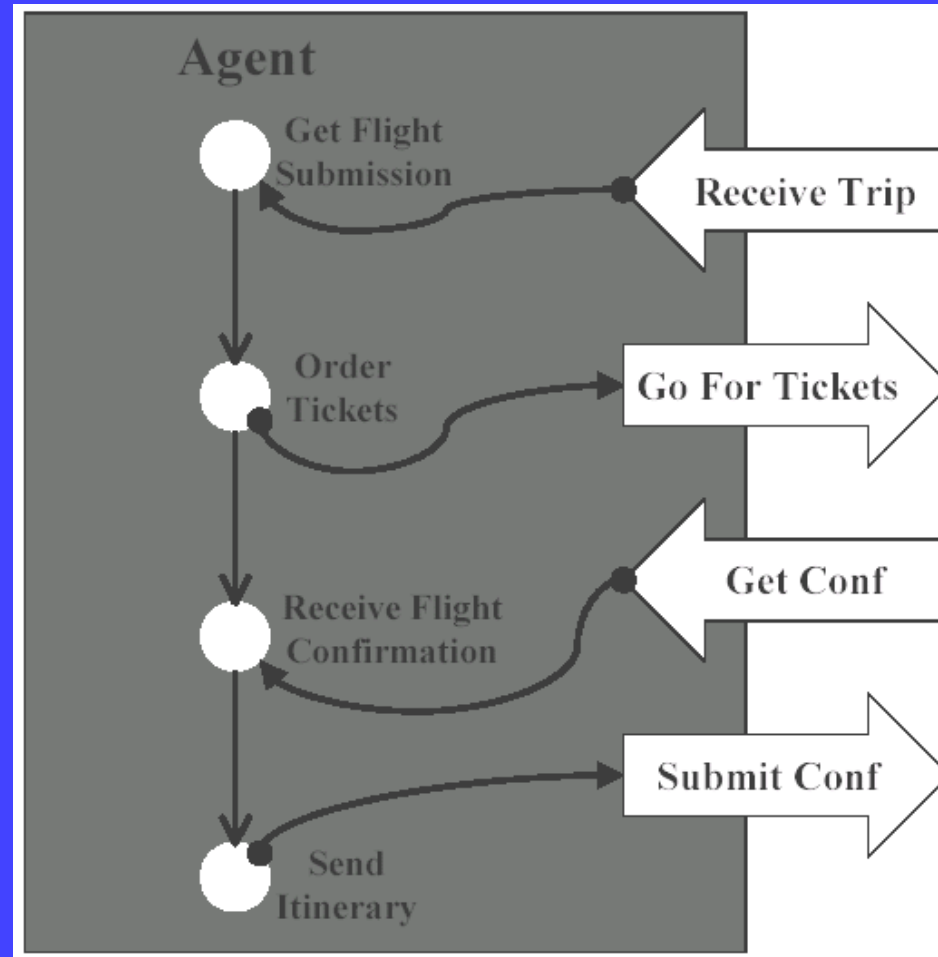
Web Services Choreographie: Aggregierung von Web Services



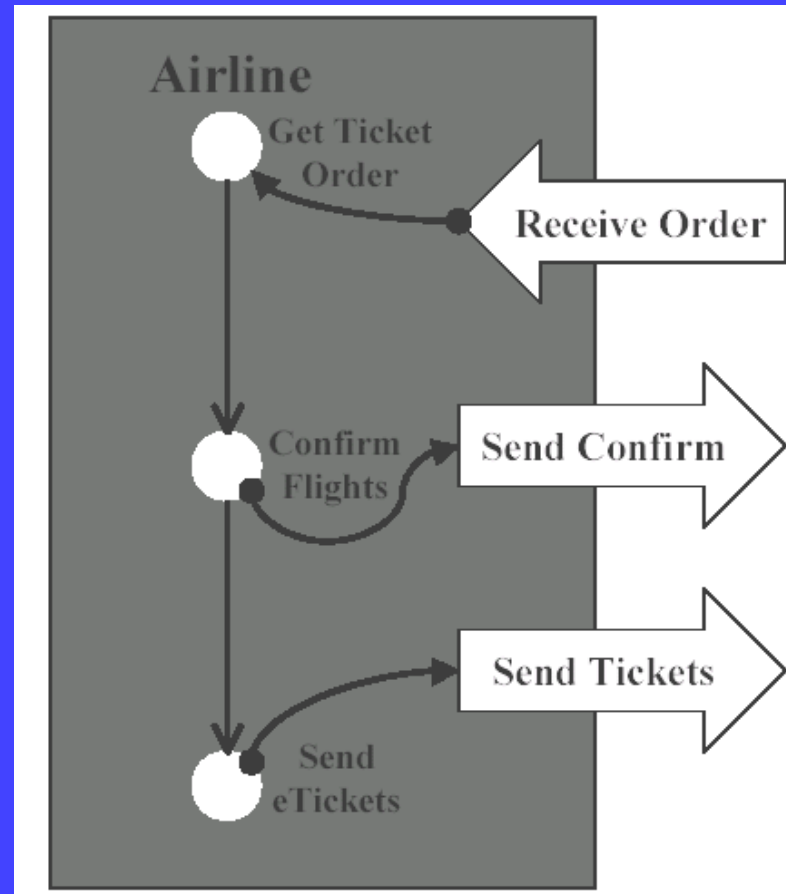
Reiseablauf als zusammengesetzter Service



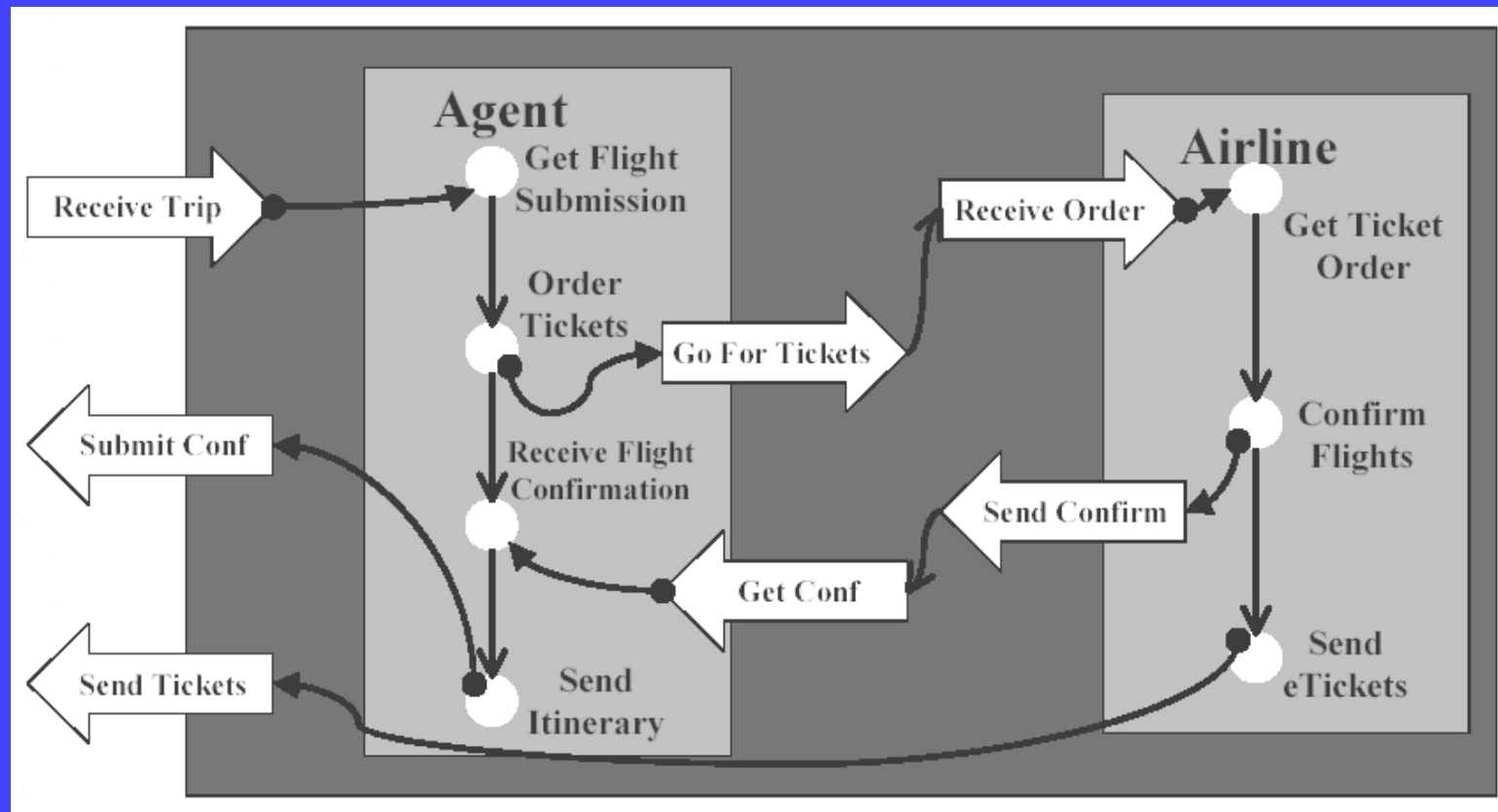
Agent als zusammengesetzter Service



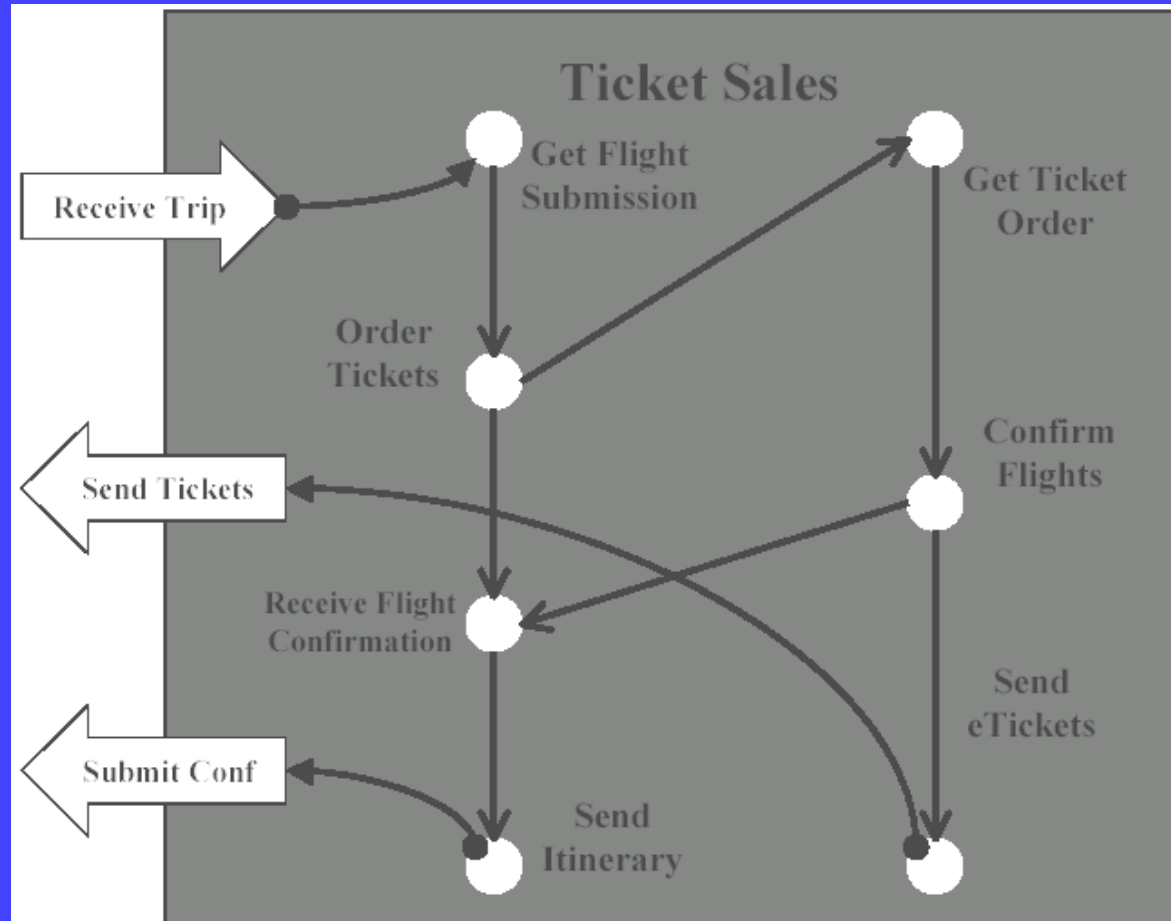
Fluggesellschaft als zusammengesetzter Service

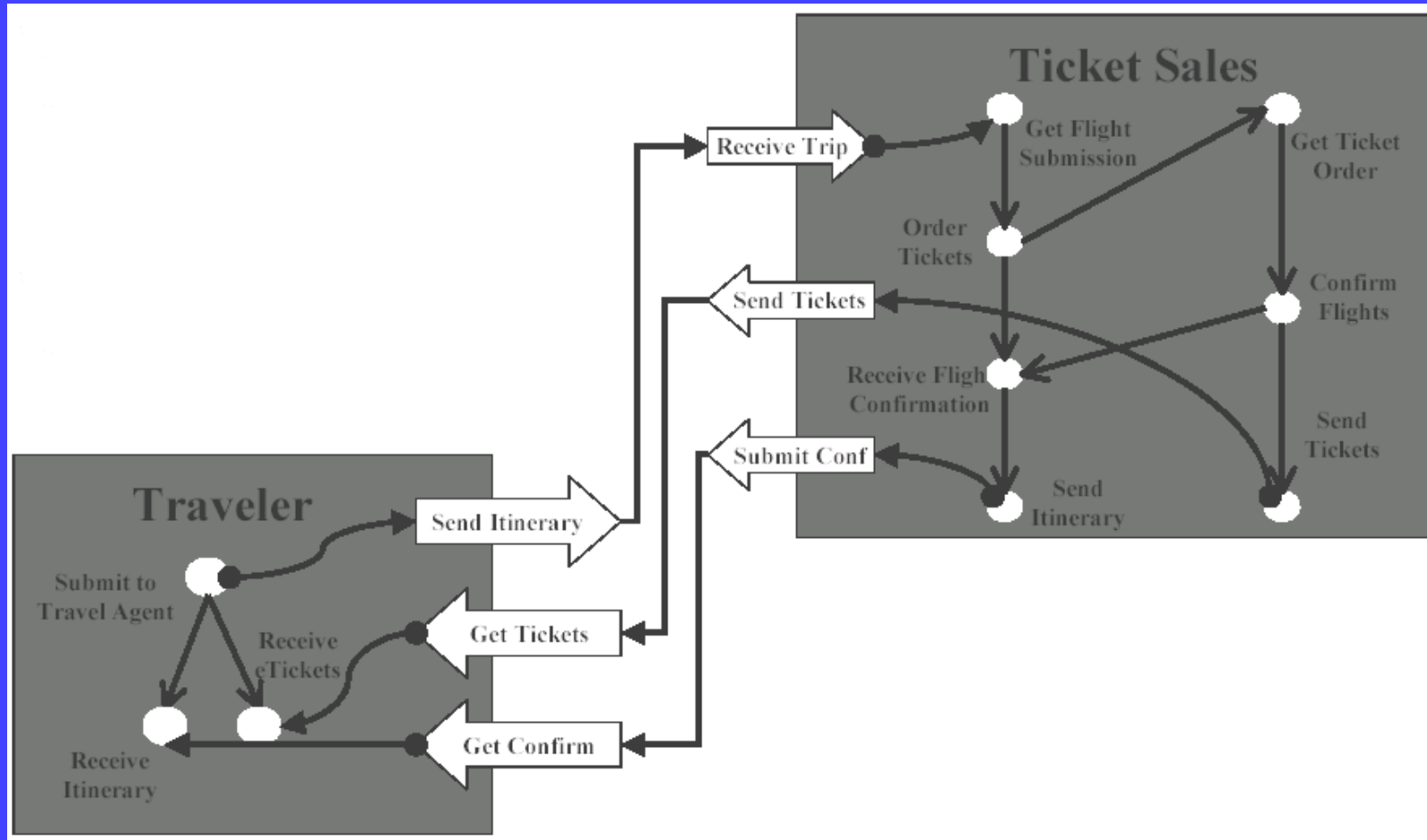


rekursive Komposition...

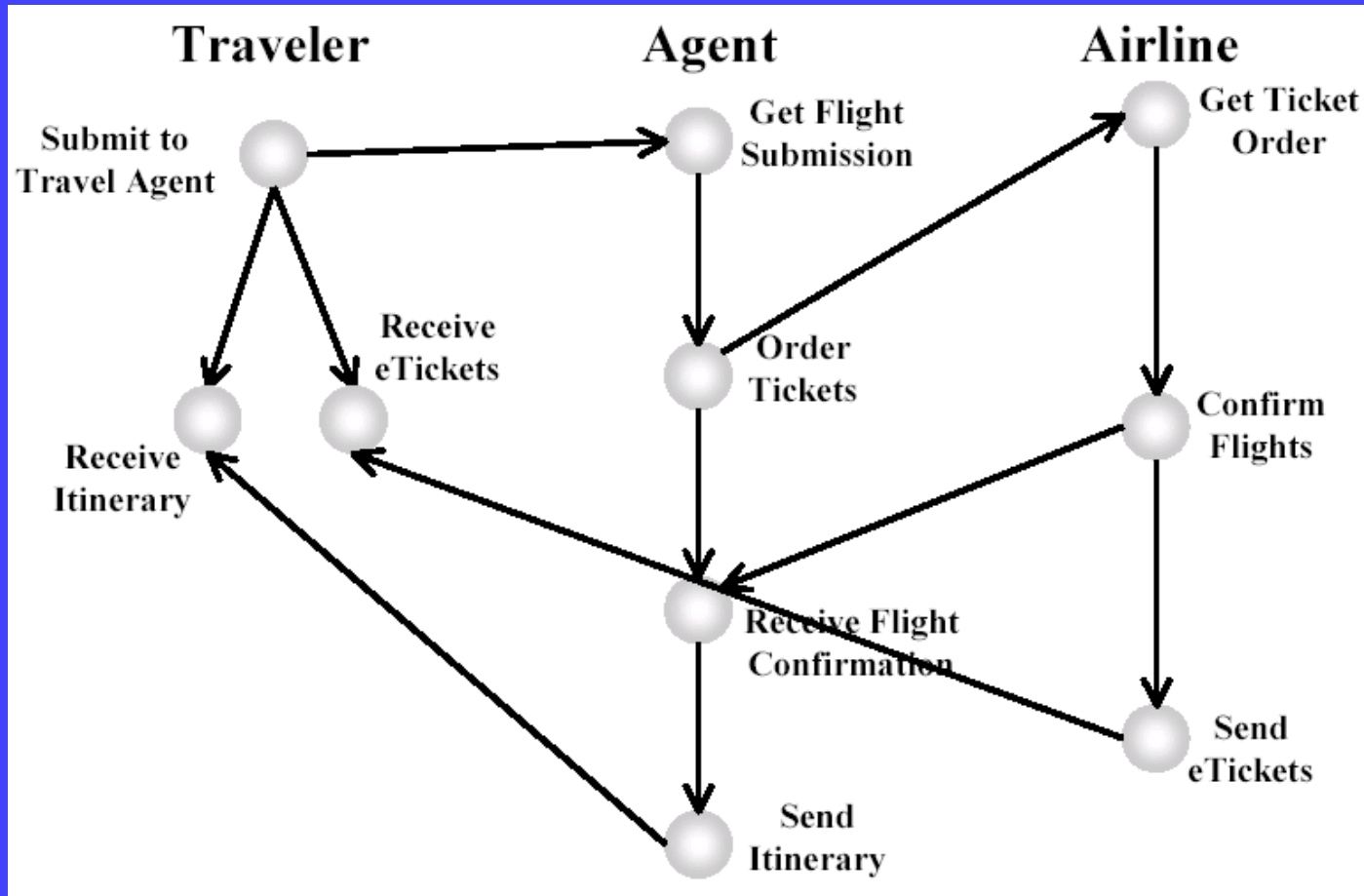


...und ist die Sicht von außerhalb





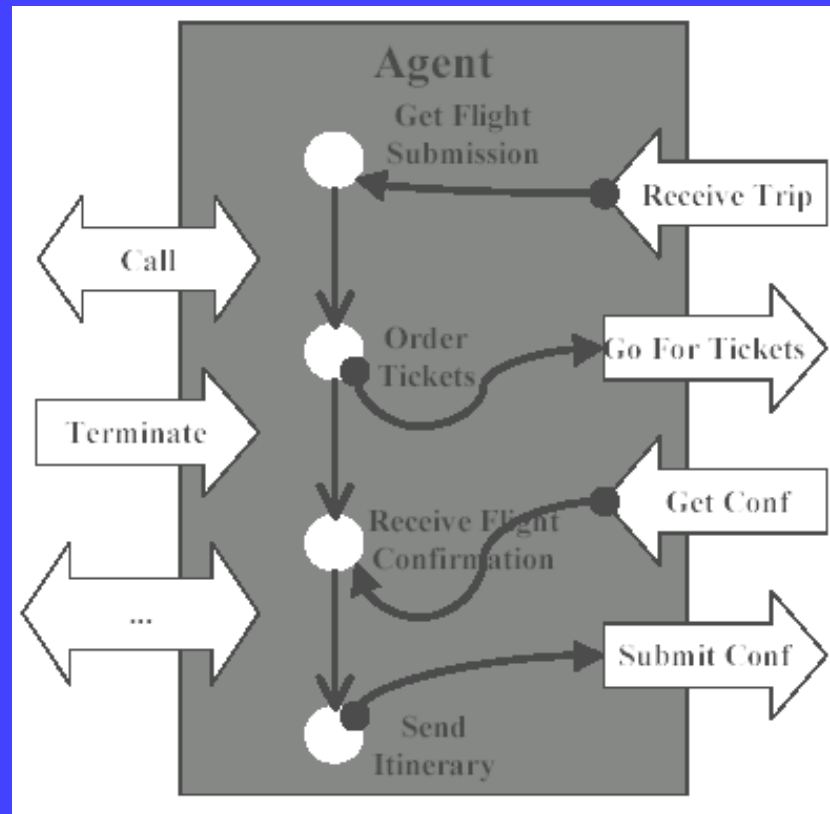
...das kennt man schon!



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)



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‘.

