



# Vorlesung "SCC – Service and Cloud Computing"

## 5. SaaS - Entwicklung und Komposition von Diensten

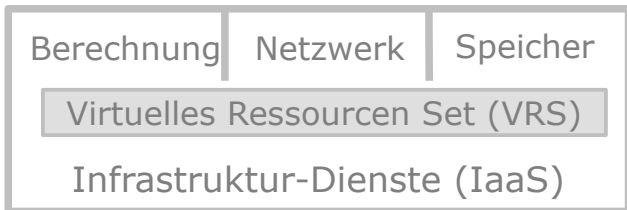
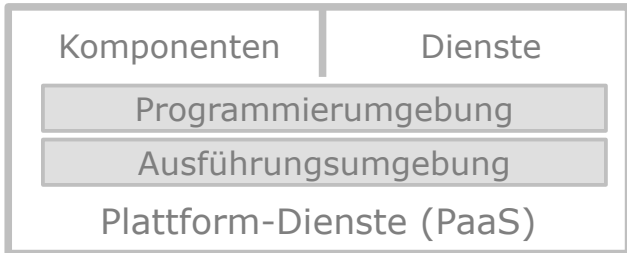
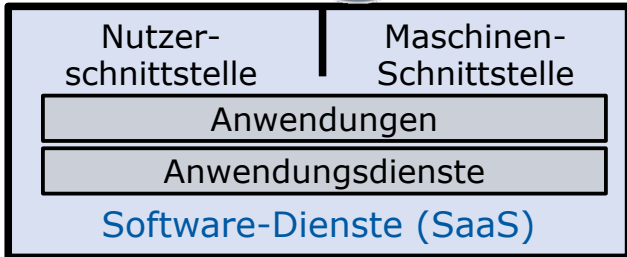
### 5.1 Design und Entwicklung von Diensten

Dr.-Ing. Iris Braun

- Wiederholung Software as a Service (SaaS)
- Konzeption serviceorientierter Anwendungen
  - Abbildung von Geschäftsprozessen
  - Entwicklung von Diensten
  - Integration
- Modellierung von Geschäftsprozessen
- Komposition – Motivation, Begriffsdefinitionen
- Choreographie vs. Orchestration
- Kompositionssprachen
  - BPEL
  - BPML
- Server-Komponenten
- Design-Werkzeuge



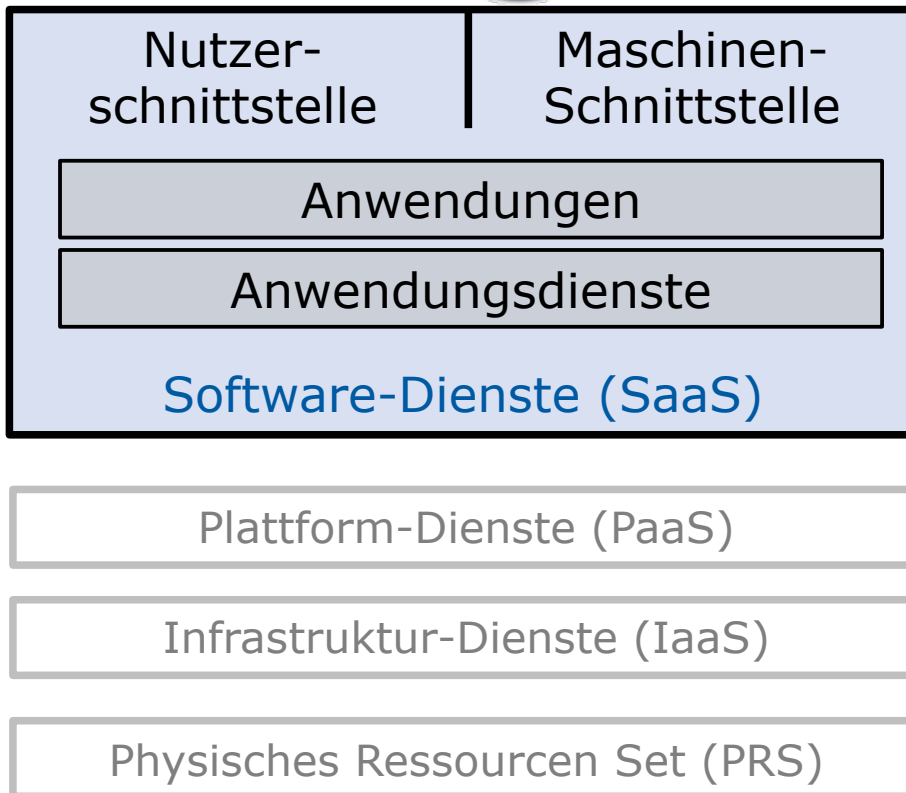
Nutzer/Clients



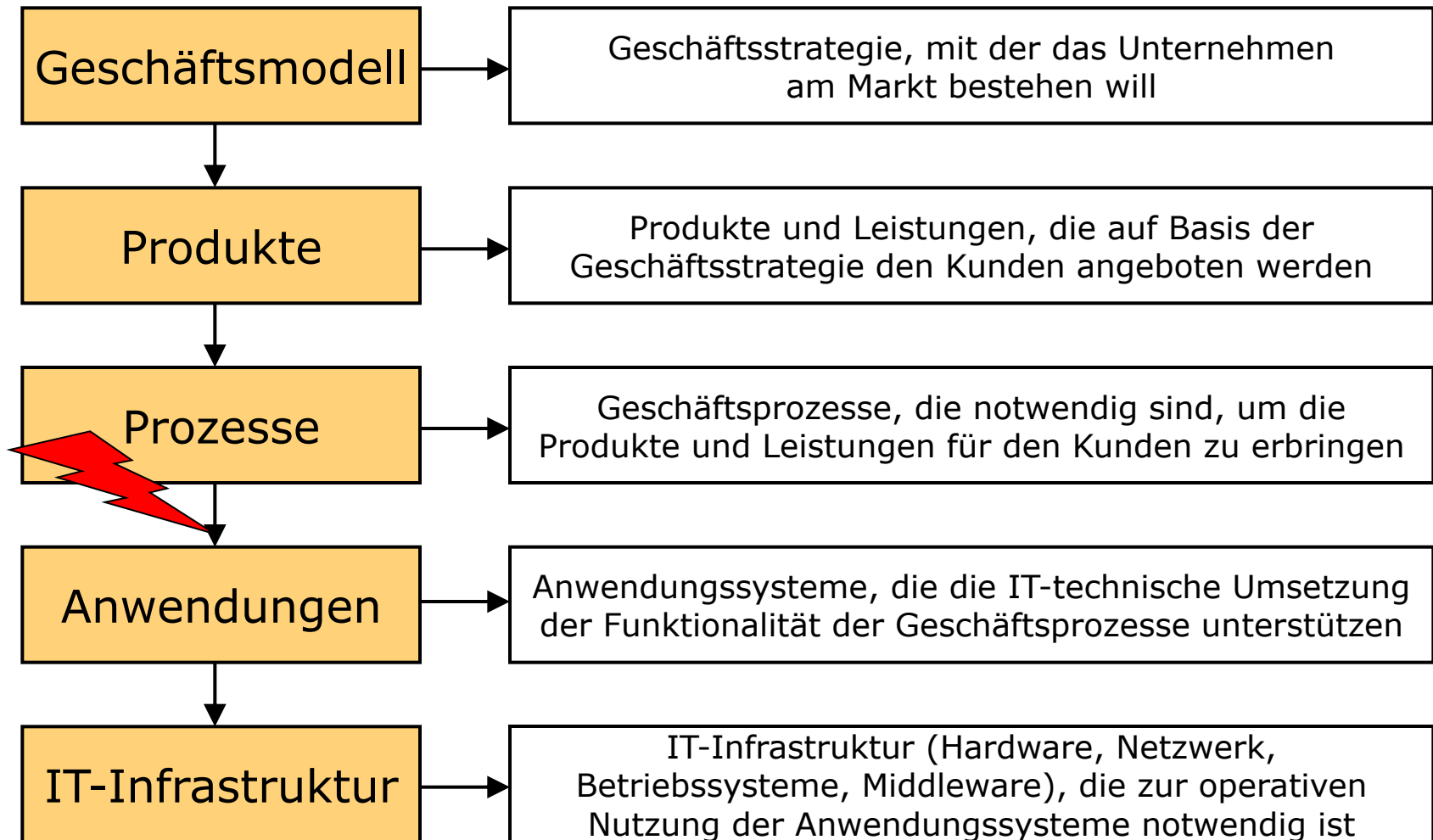
Nutzer



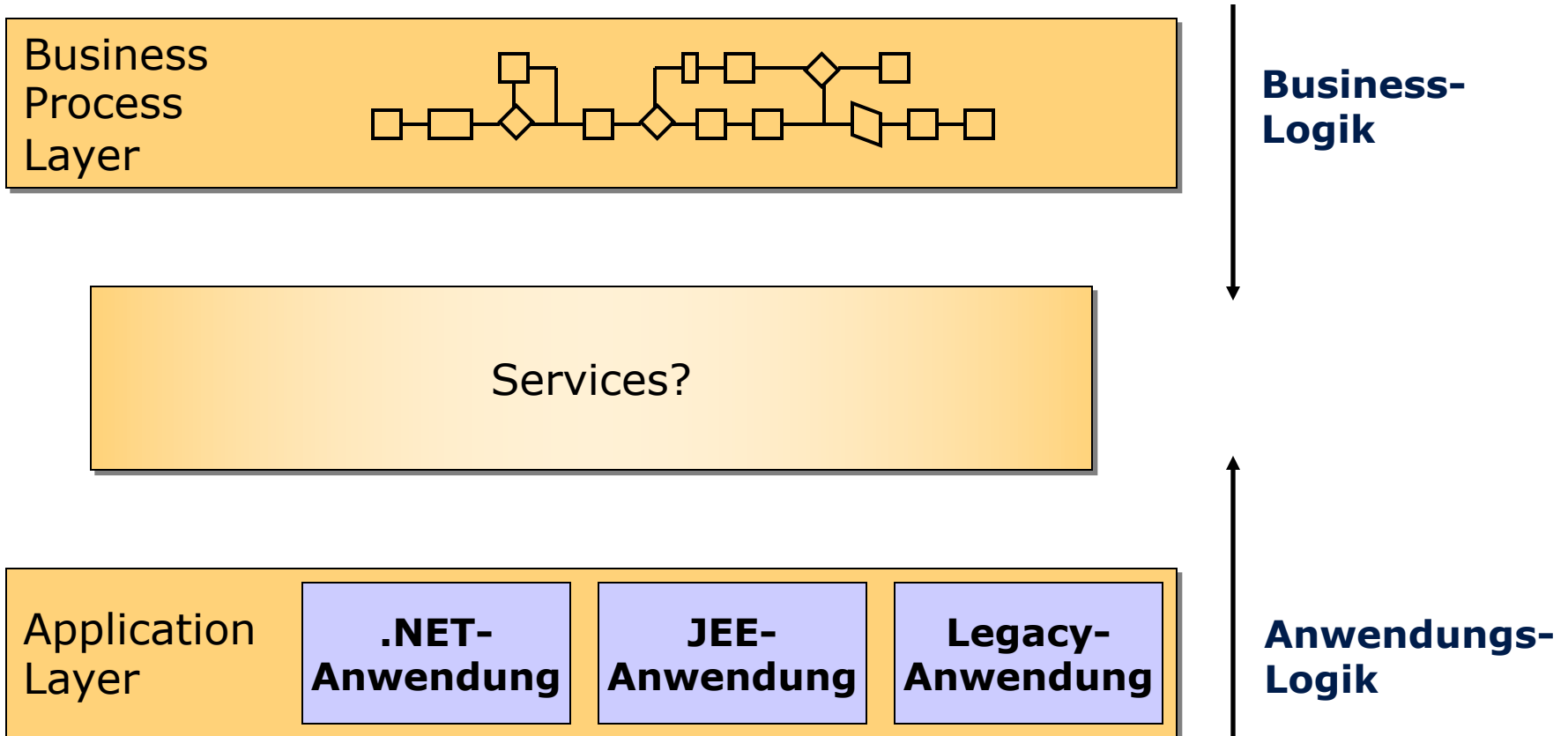
Clients



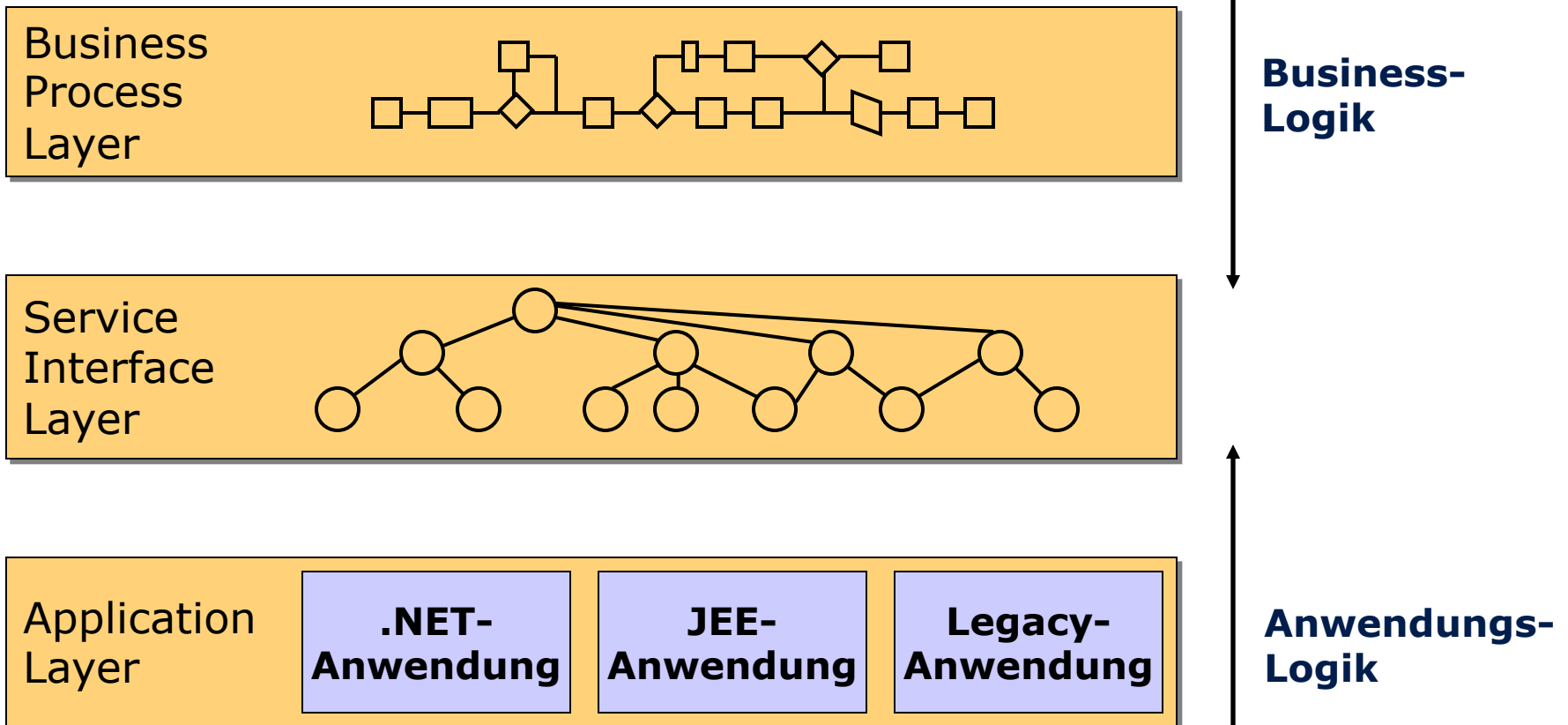
- Nutzerschnittstellen meist als Web-Applikationen oder mobile Apps
- Maschinenschnittstellen meist als SOAP oder REST
- Komplexe Anwendungen aus mehrere Diensten zusammengesetzt
- Ausführungsumgebung und Datenhaltung wird durch Plattform-Dienste bereitgestellt
- Flexible Infrastruktur-Dienste ermöglichen Skalierbarkeit und Ausfallsicherheit der Anwendungen

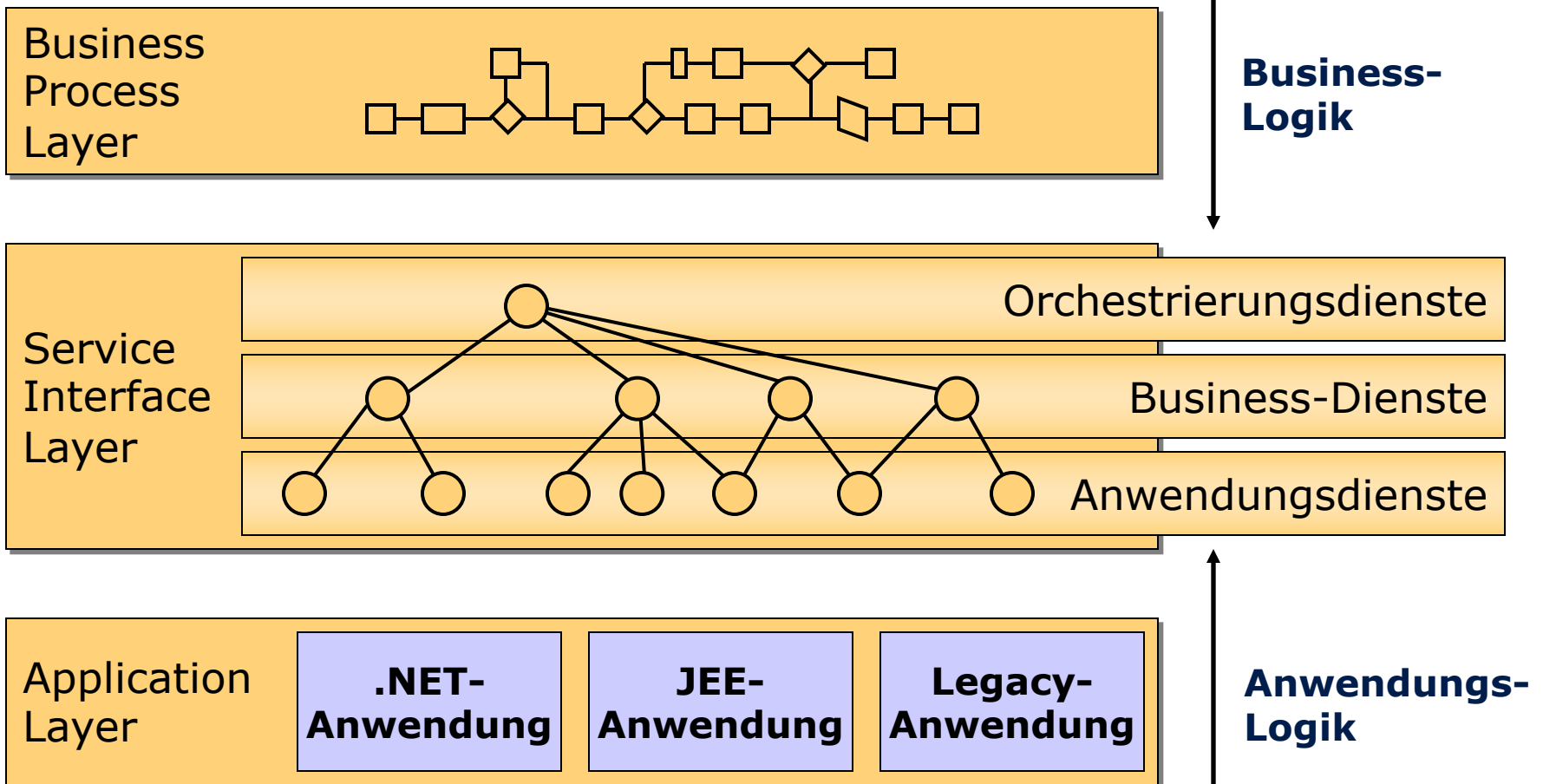


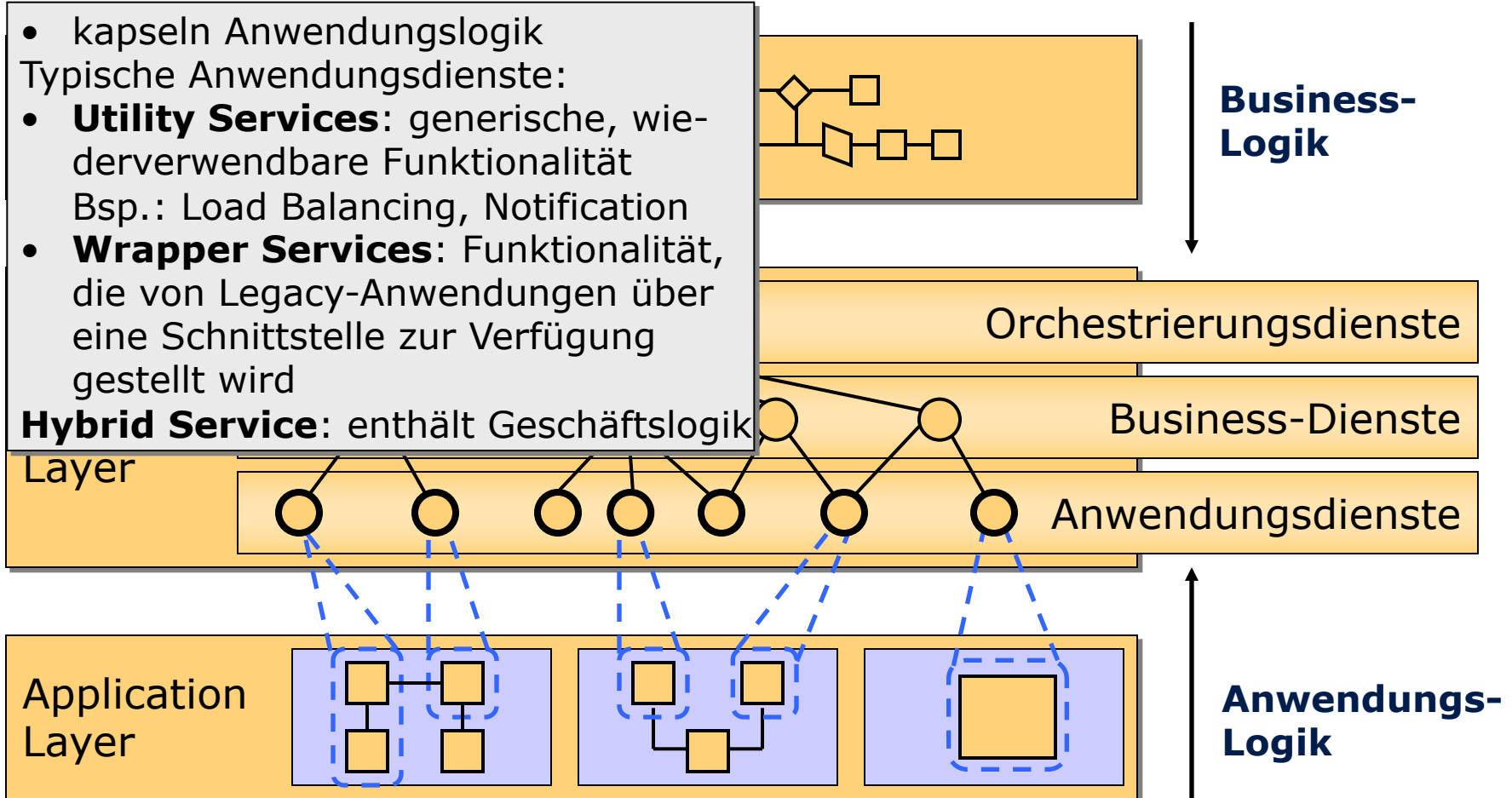
Unternehmen besteht aus Geschäftslogik und Anwendungslogik:

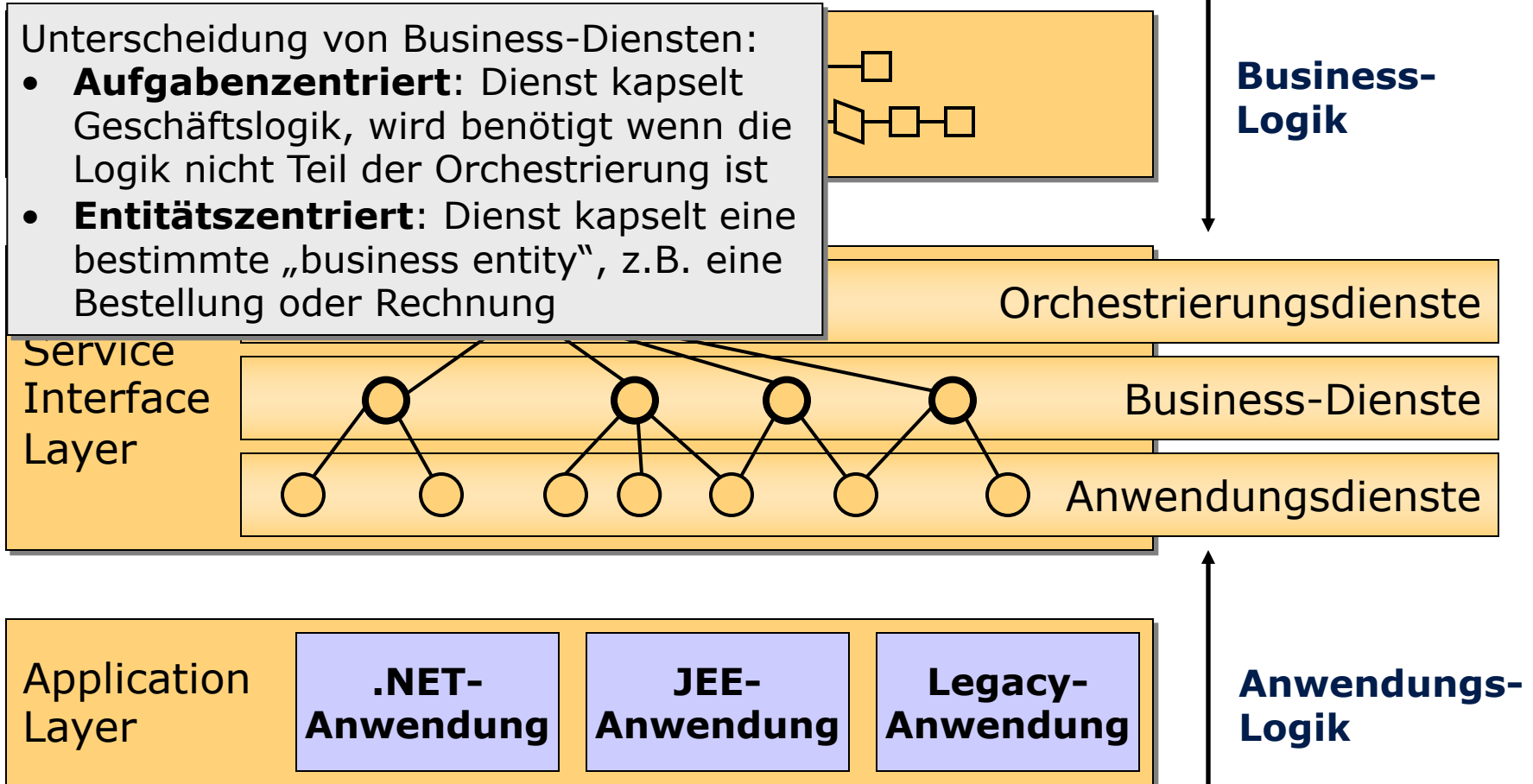


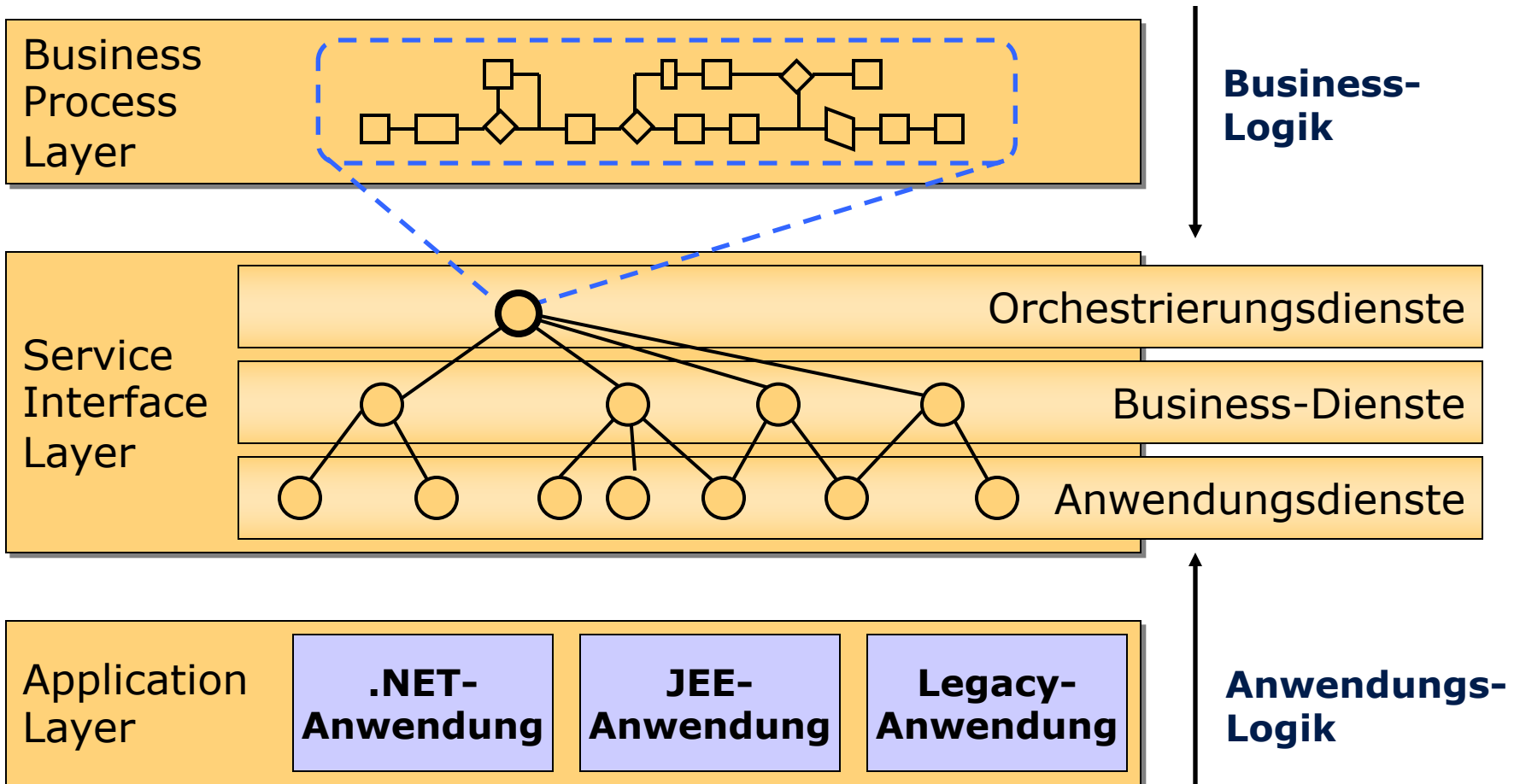
Schichtenmodell, aus Thomas Erl 2005:











- Aufteilung ermöglicht unabhängige Weiterentwicklung von Geschäftsprozess-Logik und technologieabhängiger Anwendungslogik ⇒ lose Kopplung
- Entwicklung der Anwendungslogik in Hinblick auf Service-Orientierung
- Geschäftsprozess-Logik lässt sich getrennt betrachtet leichter auf das Geschäftsmodell des Unternehmens abstimmen
- Agilität wird verbessert, indem Änderungen im Geschäftsprozess möglichst unabhängig von Implementierungen der Geschäfts- und Anwendungslogik realisierbar sind (Ziel der Orchestrierung)



## Grundlegende Schritte der serviceorientierten Analyse:

### 1. Umfang der Analyse festlegen

- Nur einzelne wiederverwendbare Dienste entwickeln oder vollständige Geschäftsprozesse auf Basis von Diensten?

### 2. Existierende Systeme identifizieren

- Welche davon können von Veränderungen betroffen sein?

### 3. Dienstkandidaten (service candidates) modellieren

- Geschäftsprozess zerlegen (z.B. mittels Flussdiagramm oder BPMN)
- Einordnen der einzelnen Operationen in logische Kontexte
- Wiederverwendbarkeit im Auge behalten
- Business-Analysten einbeziehen (für Business-Dienste)

Empfehlenswerte Vorgehensweise:

- **Entitätszentrierte** Businessdienste entwerfen
  - Entities relativ unabhängig von anderen Diensten, bilden gute Basis für ressourcen-orientierten Entwurf (REST)
- **Anwendungsdienste** entwerfen
  - Abstraktion der technischen Umgebung, wichtigste Eigenschaft: Wiederverwendbarkeit
- **Aufgabenzentrierte** Businessdienste entwerfen
  - Teilaktivitäten der Geschäftsprozesse, die einer bestimmten Aufgabe zuzuordnen sind (operationale Dienste SOAP)
- **Geschäftsprozessdienste** entwerfen
  - Akkurate Abbildung der Geschäftsprozesslogik, Verwendung von Aktivitäts-/Sequenzdiagrammen üblich (BPMN, WS-BPEL)



# Vorlesung "SCC – Service and Cloud Computing"

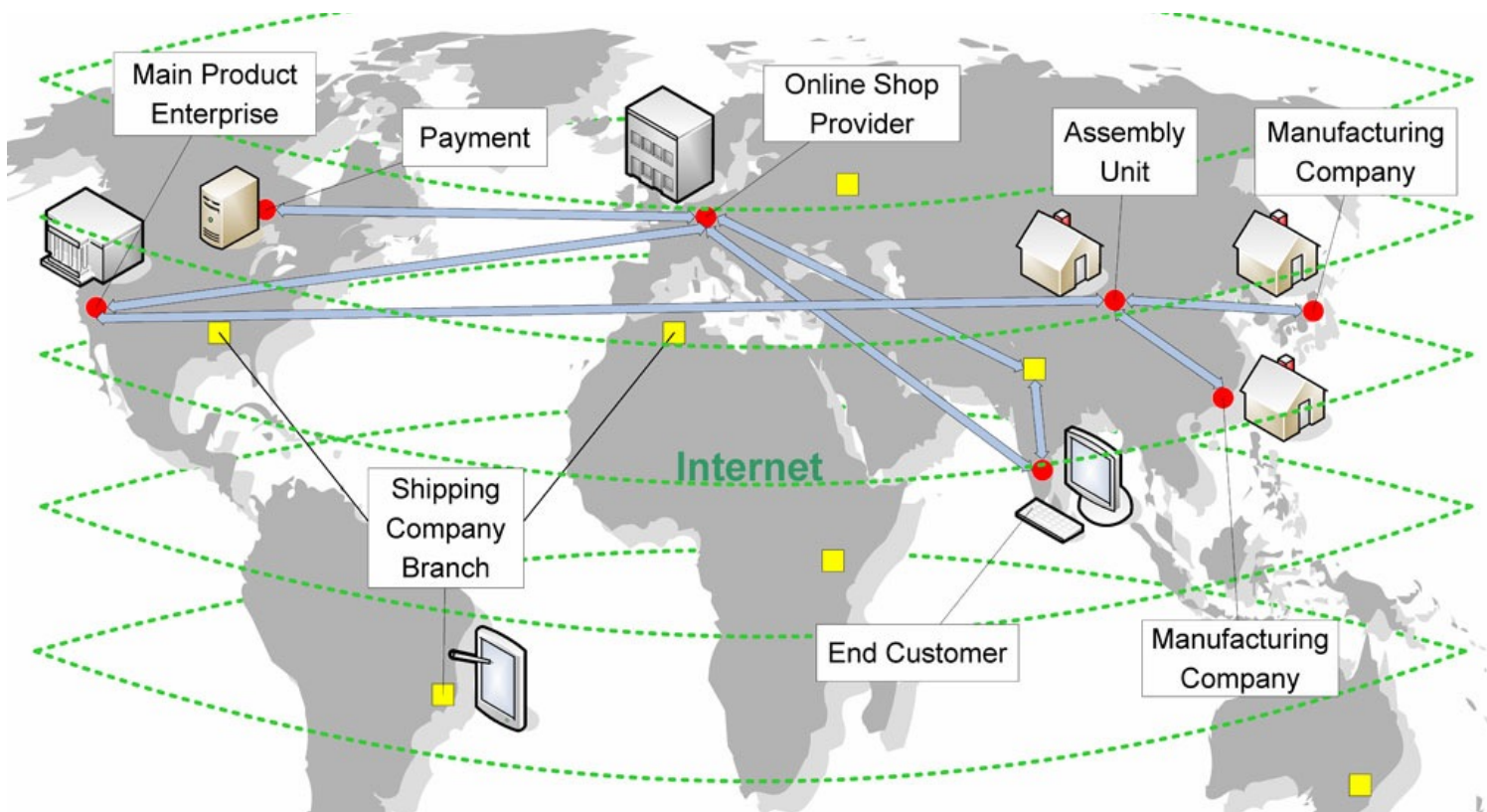
## 5. SaaS - Entwicklung und Komposition von Diensten

### 5.2. Komposition von Diensten

Dr.-Ing. Iris Braun

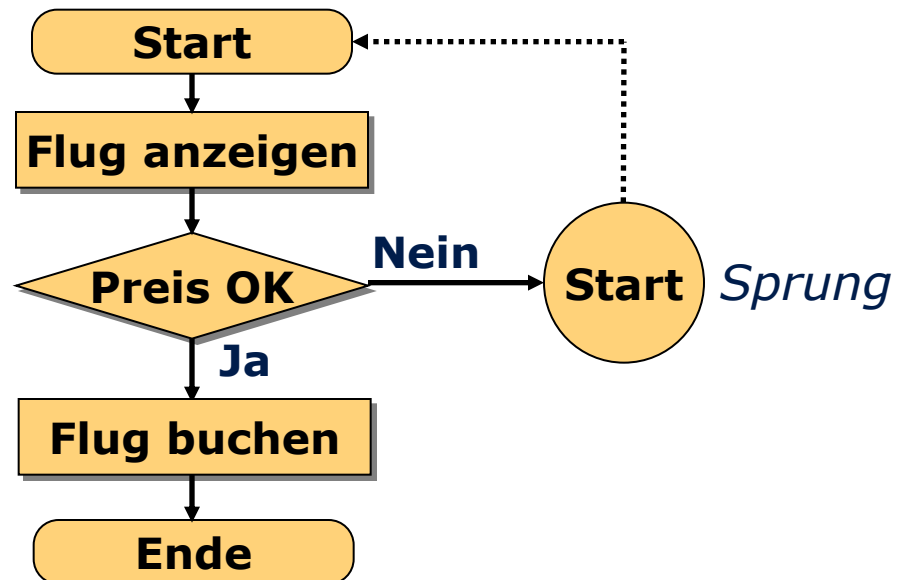
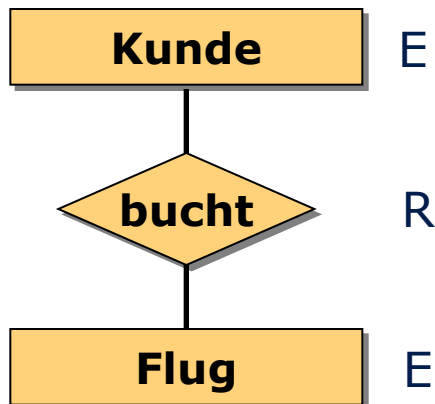
- Wiederholung Software as a Service (SaaS)
- Konzeption serviceorientierter Anwendungen
  - Abbildung von Geschäftsprozessen
  - Entwicklung von Diensten
  - Integration
- Modellierung von Geschäftsprozessen
- Komposition – Motivation, Begriffsdefinitionen
- Choreographie vs. Orchestration
- Kompositionssprachen
  - BPEL
  - BPML
- Server-Komponenten
- Design-Werkzeuge

**Geschäftsprozess:** Anzahl von Aktivitäten, die einen Mehrwert für den Kunden generieren



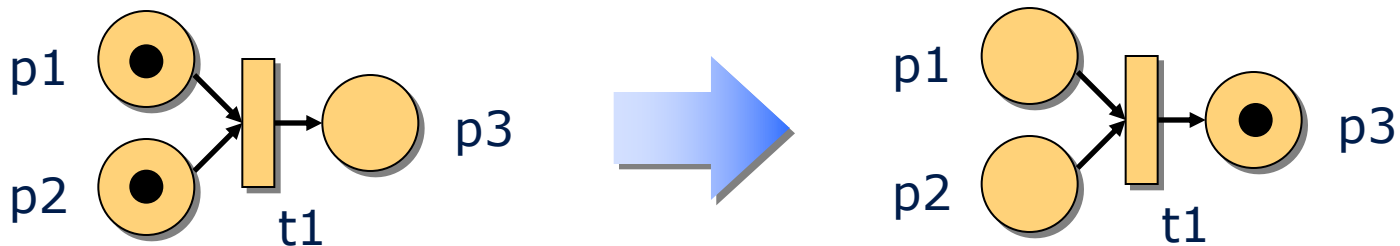
## Definierte Geschäftsprozess-Modellierung:

- Notwendig für die Zusammenarbeit von Business-Analysten und Softwareentwicklern
- Frühe Form: ER-(Entity-Relationship-)Diagramme
  - Datenmodellierung, nicht für komplexe Prozesse geeignet
- Flussdiagramme: Darstellung als Abfolge von Aktivitäten



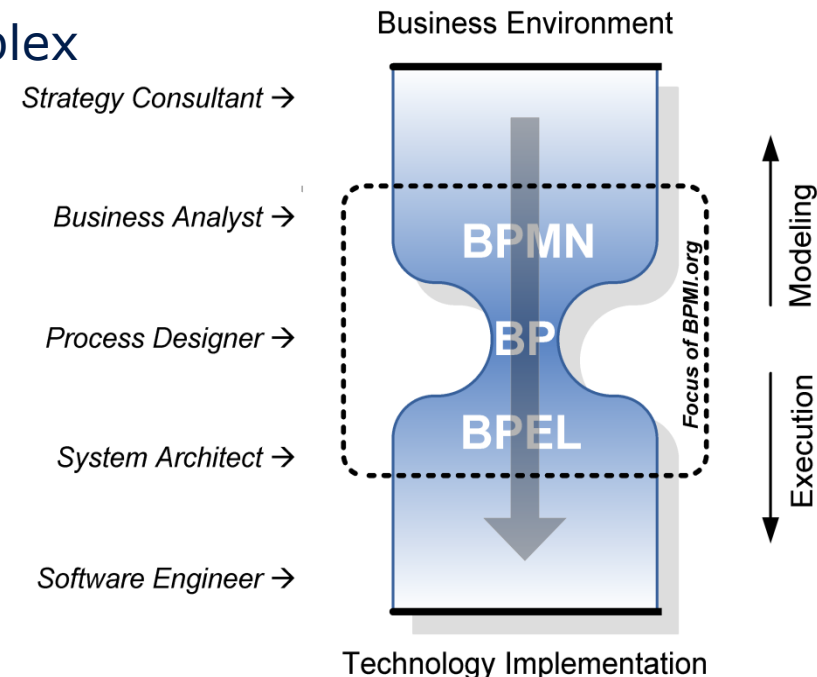
## Präzisere Notationen: basierend auf Petri-Netzen

- 1962 vom Mathematiker Carl Adam Petri vorgestellt
- Formales mathematisches Modell: gerichteter Graph
- Stellen (Places) und Übergänge (Transitions), die durch gerichtete Kanten verbunden sind
- Stellen können mit Marken belegt sein und eine Kapazität haben
- Mehrere Erweiterungen (verschiedene Marken, Zeit)  $\Rightarrow$  komplex



Darauf basierende Notationen zur Modellierung:

- Aktivitäts-Diagramme in UML (Unified Modeling Language ) der Object Management Group (OMG)
  - Zu eingeschränkt für die Modellierung komplexer Prozesse
- Yet Another Workflow Language (YAWL)
  - Erweitert Petrinetze, relativ komplex
- Business Process Modeling Notation (BPMN) von BPMI.org
  - Basiert auf Flussdiagrammen
  - Mapping von BPMN auf BPEL
  - Breite Unterstützung durch die Industrie



1. Kunde fragt im Reisebüro nach Hotelzimmer
2. Reisebüro liefert Angebot
3. Kunde bucht das Hotelzimmer laut Angebot
4. Bank bucht Geld von Konto des Kunden ab und benachrichtigt Reisebüro über erfolgreiche Abbuchung
5. Reisebüro übergibt Buchungsbestätigung an Kunde



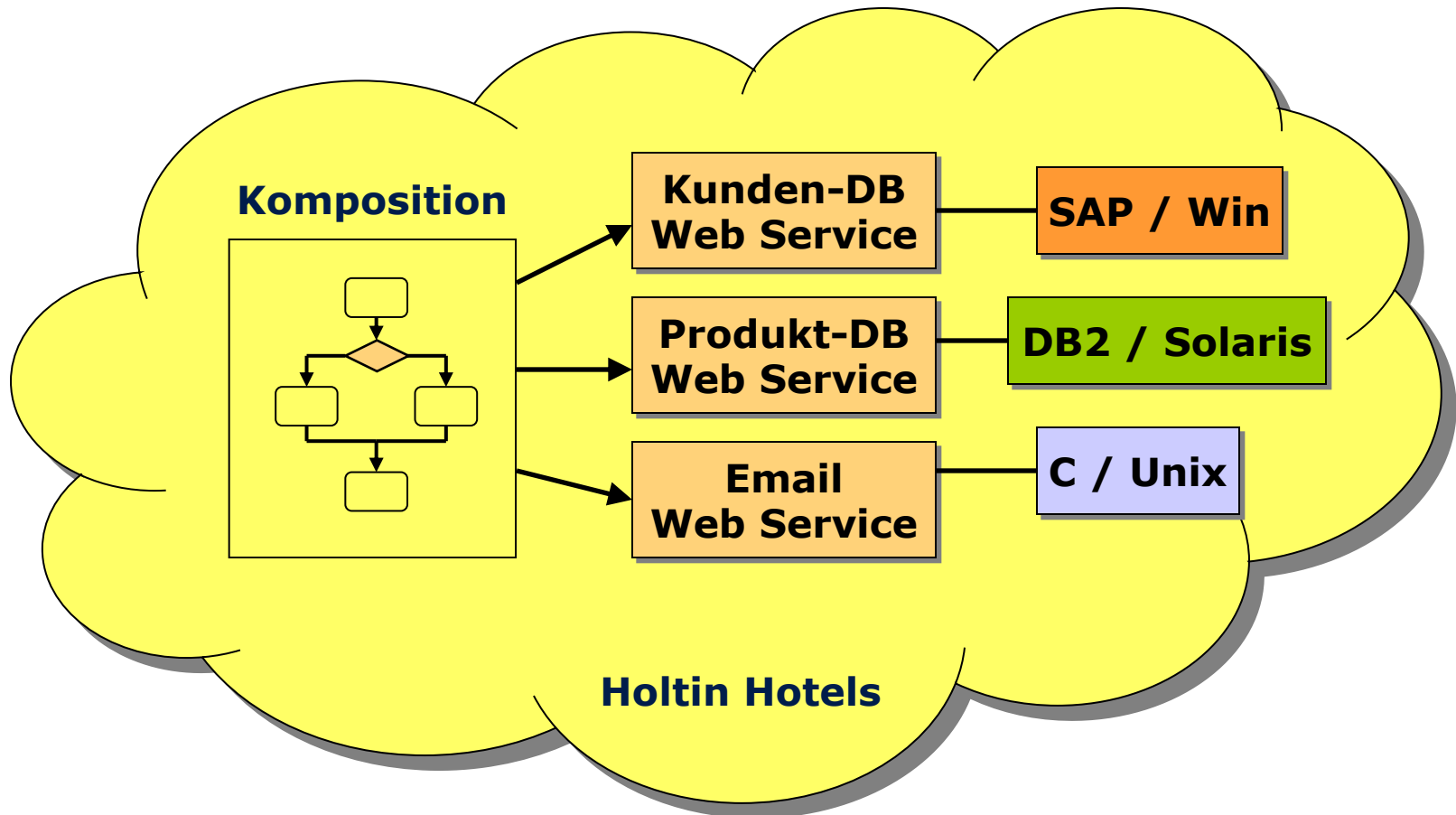
- Mit SOAP, WSDL, UDDI möglich?

## Was ist mit SOAP, WSDL, UDDI möglich?

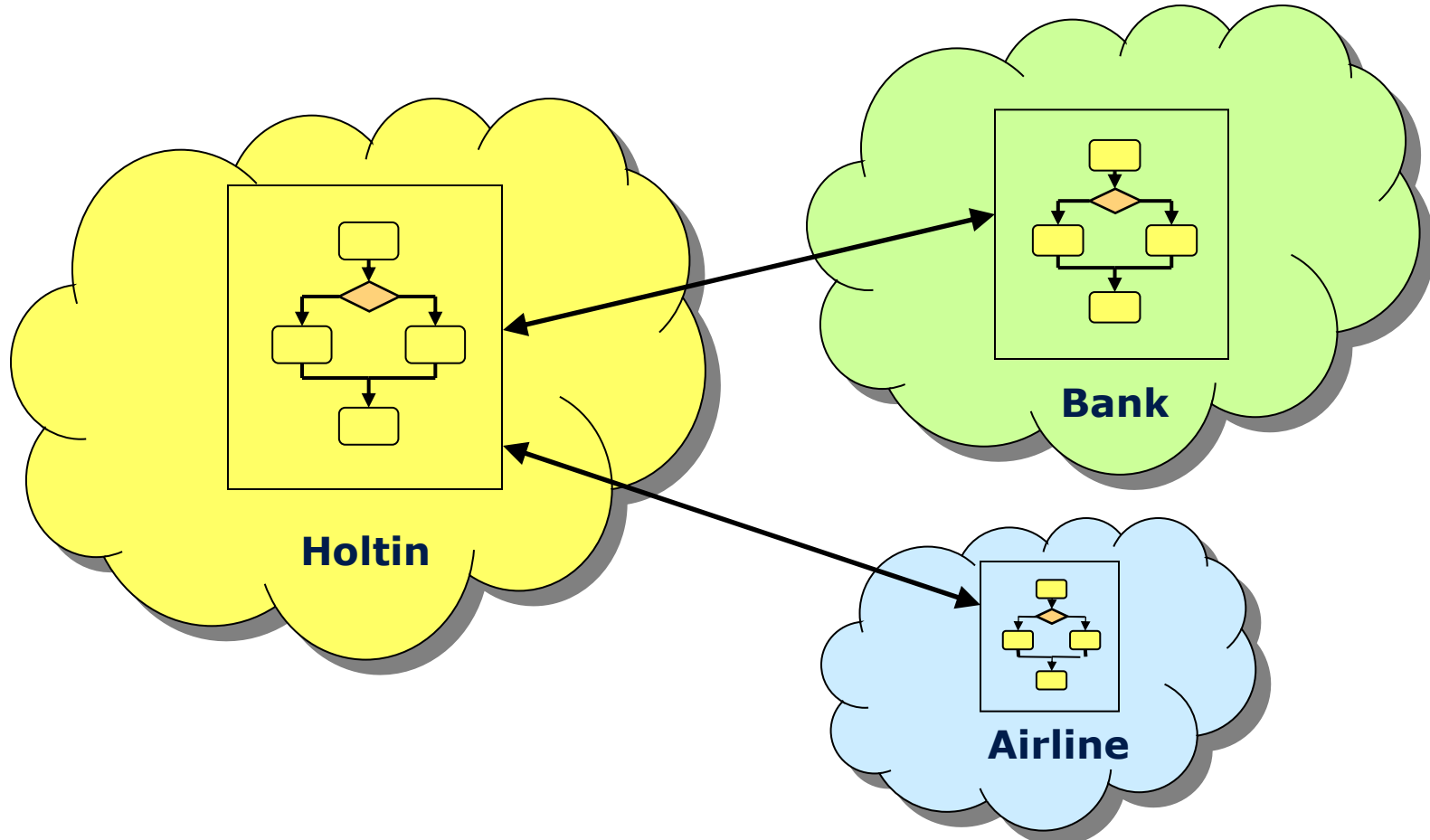
- Beschreibung von Diensten
    - Operationen
    - Nachrichten
    - Datentypen der Nachrichten-Inhalte
  - Übertragung von Nachrichten
    - XML-Basiertes Nachrichtenformat (SOAP)
    - über Standard-Internetprotokolle (HTTP, SMTP, ...)
  - Standardisierung der Dienstveröffentlichung / -suche
- 
- Beschreibung von einzelnen Operationen: möglich
  - Beschreibung von Operations- und Nachrichtenabfolgen: **nicht** möglich

WSDLSOAPUDDI

- **EAI** (Enterprise Application Integration)



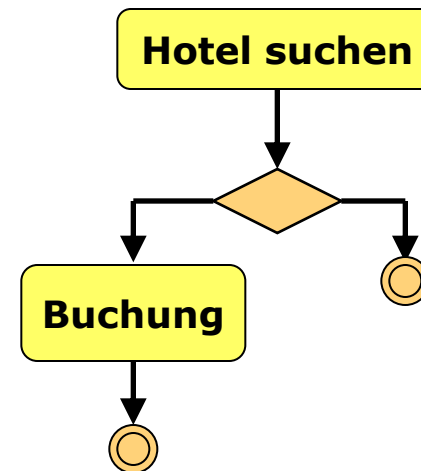
- **B2B-Integration** (Business-to-Business-Integration)



- Kodierung des Ablaufs mit einer Programmiersprache
- Nachteil: kompliziert, nur auf Systemen mit Unterstützung der Sprache ausführbar, Änderungen schwierig

```
import com.holtin.HotelSearch.*;
public class workflow { ...
    public work() {
        HSS hss = HSS();
        Properties prop = Properties();
        ...
        Result r = hss.search(prop);
        if (r!=null) r.book(1,from,to);
    }
}
```

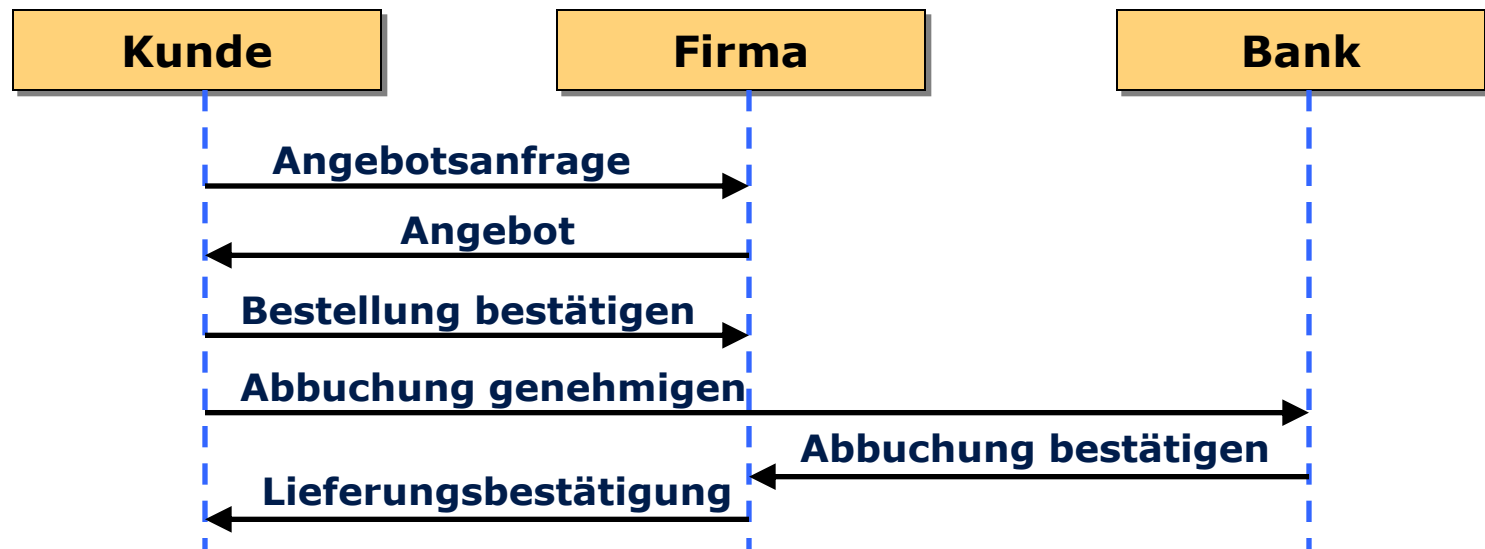
- Deklarative Beschreibung des Ablaufs, Ausführung mittels einer Engine
- Standardisierung: z.B. 1993 durch Workflow Management Coalition



- Mögliche Kontroll- und Datenstrukturen für Prozessmodellierung
- Möglichkeiten zur Fehlerbehandlung
- Transaktionsunterstützung
- Analyse und Monitoring
  
- Verfahren zur Suche und Auswahl geeigneter Web Services
- Beschreibung der Web Services
  - Semantische Beschreibung möglich?
  - Qualitätsaspekte berücksichtigt?
  
- Vor Erläuterung bestimmter Kompositions-Techniken: Definition und Erklärung von Begriffen

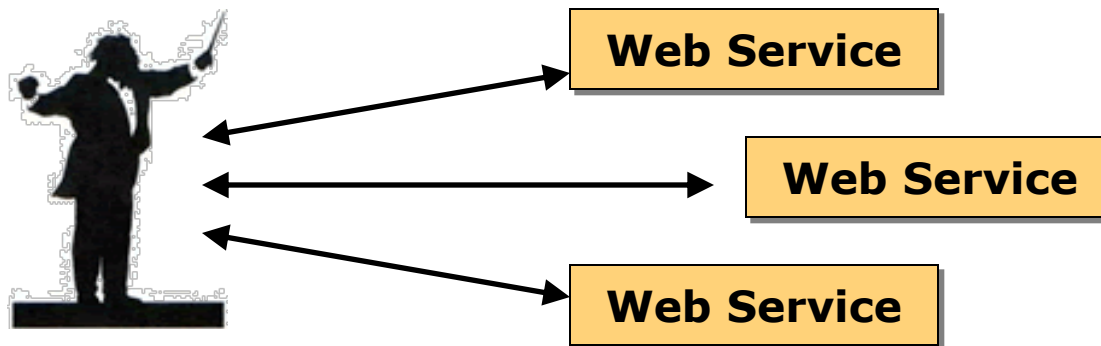
## Konversation (Conversation):

- Abfolge von Operationen zwischen Web Services
- ⇒ Konkrete Kommunikation von zwei oder mehreren Web Services
- Normalerweise Kontext nötig, der den Zusammenhang der einzelnen Operationen definiert (z.B. Vorgangs-/Kundennummer)
- Darstellung: z.B. Aktivitäts-, Zustands- oder Sequenzdiagramm



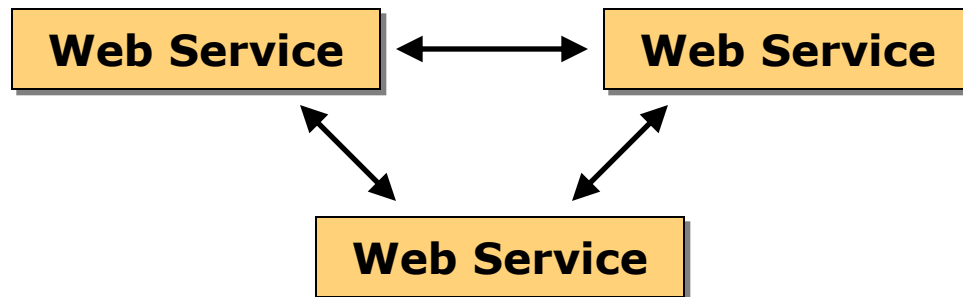
## Orchestrierung (Web Services Orchestration):

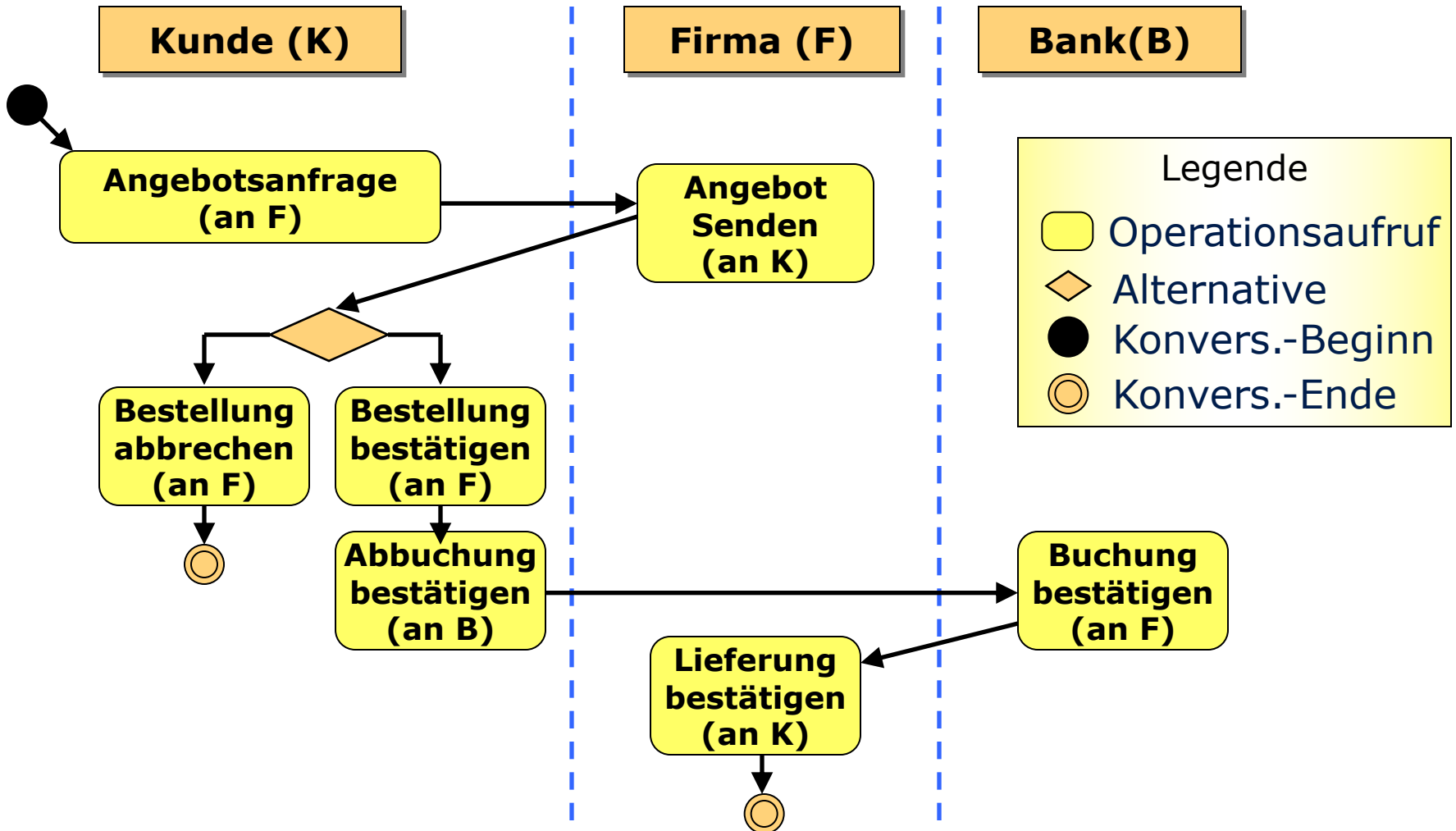
- Komposition von Web Services für Geschäftsprozesse
- Definition von Prozessen, die existierende Web Services wiederverwenden
- Ziel: deklarative (nicht-programmatische) Beschreibung eines zusammengesetzten Dienstes
- Definierte Elemente:
  - Auszuführende Dienste
  - Reihenfolge der Ausführung, Parallelität und bedingte Ausf.

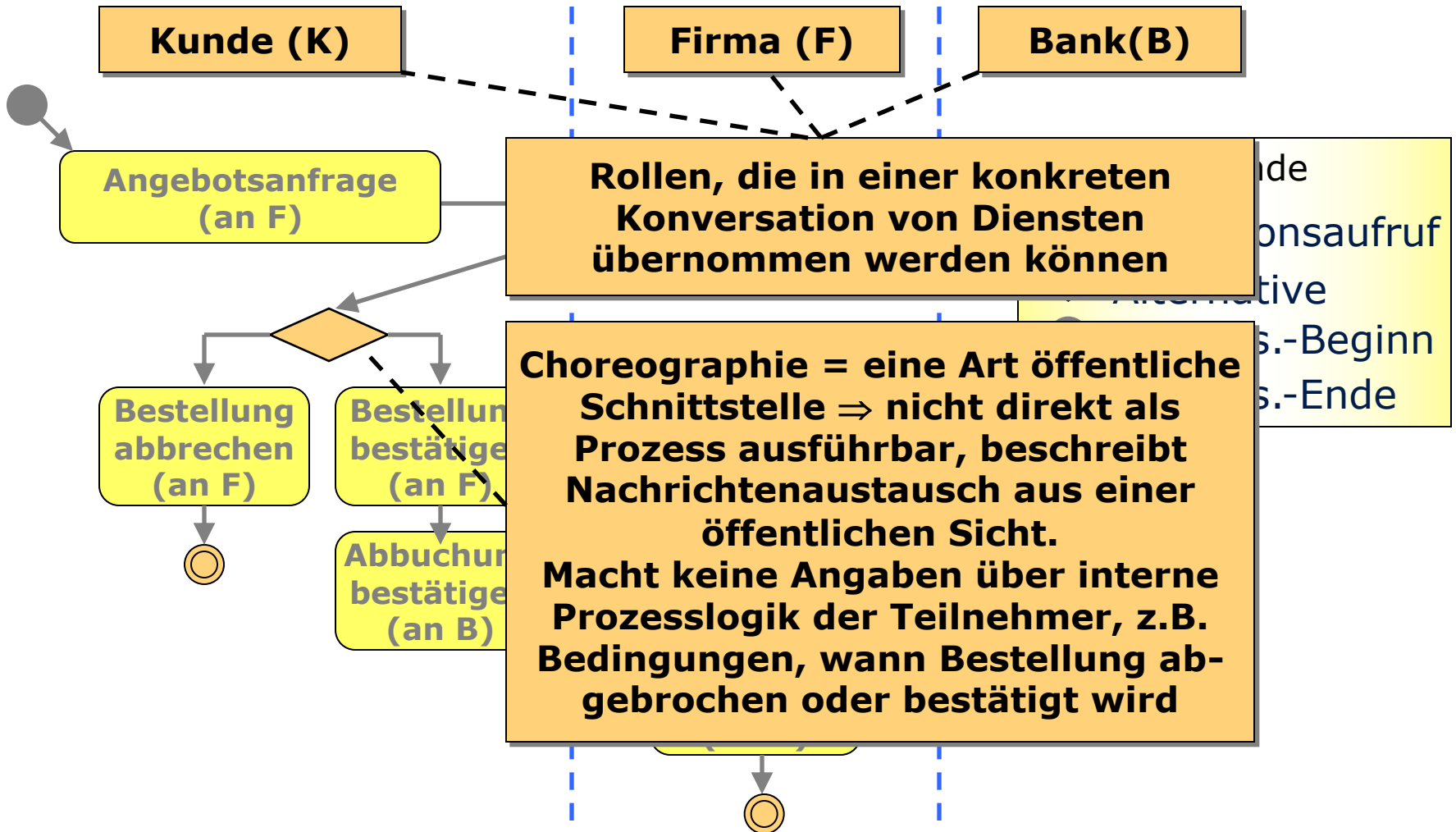


## **Choreographie (Web Services Choreography):**

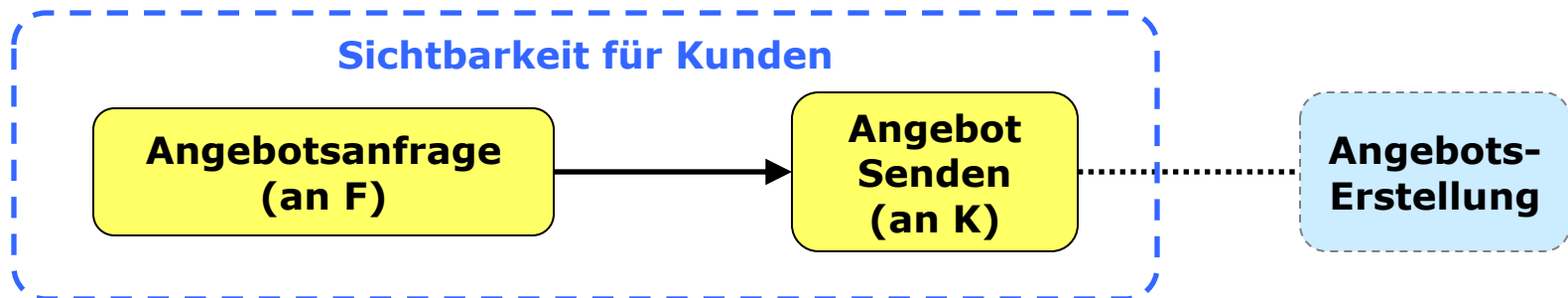
- Komposition von Web Services für geschäftliche Zusammenarbeit
- Definition der Peer-to-Peer-Kollaboration mehrerer Teilnehmer
- Gegensatz zum von einem Teilnehmer ausgeführten Geschäftsprozess (bei WSO): Teilnehmer können ihren Teil der Interaktion (Nachrichtenaustausch zwischen Teilnehmern) beschreiben
- Erforderlich: formale Beschreibung des Nachrichtenaustauschs, die das Protokoll zum Peer-to-Peer-Austausch modelliert



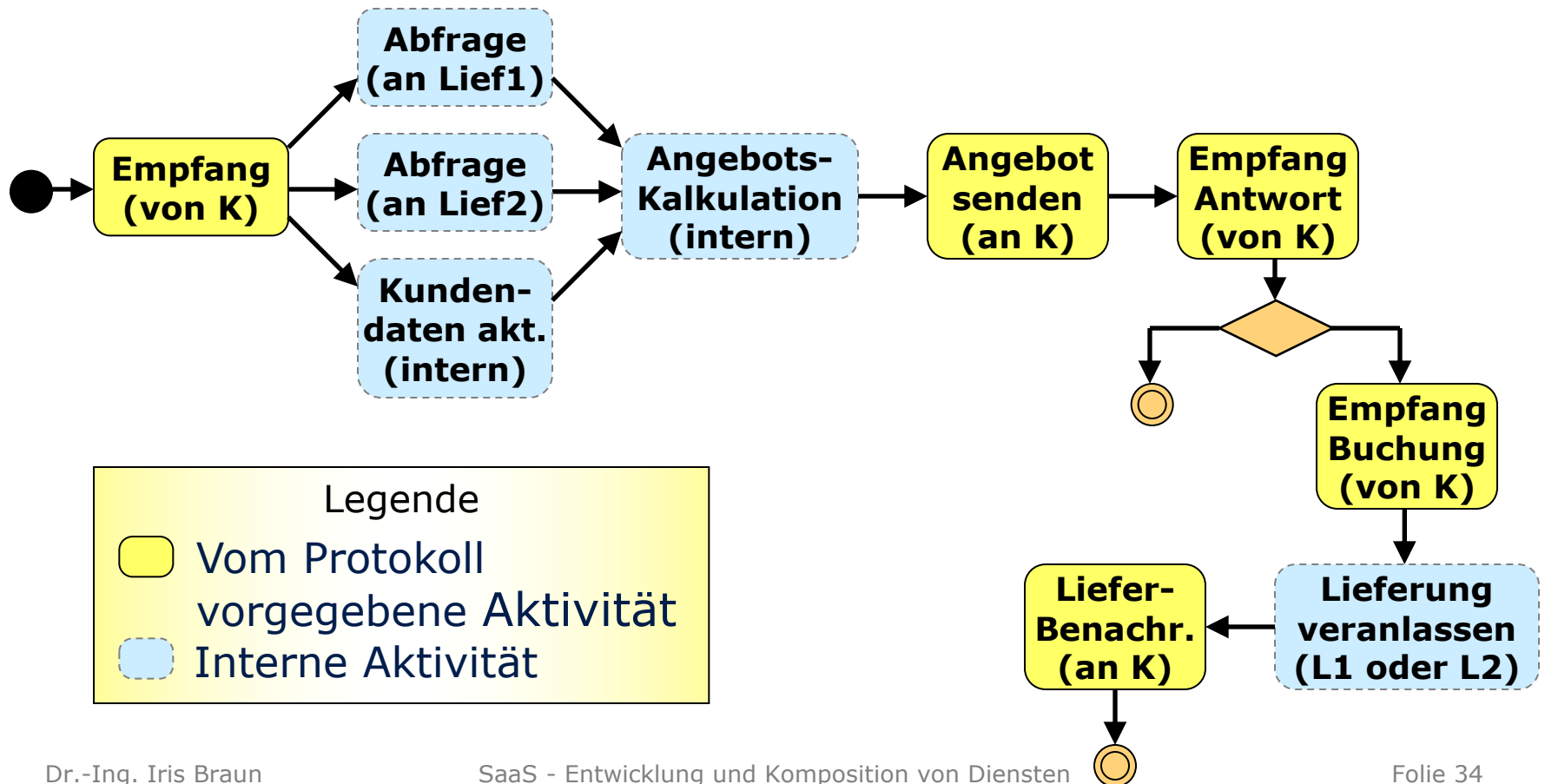




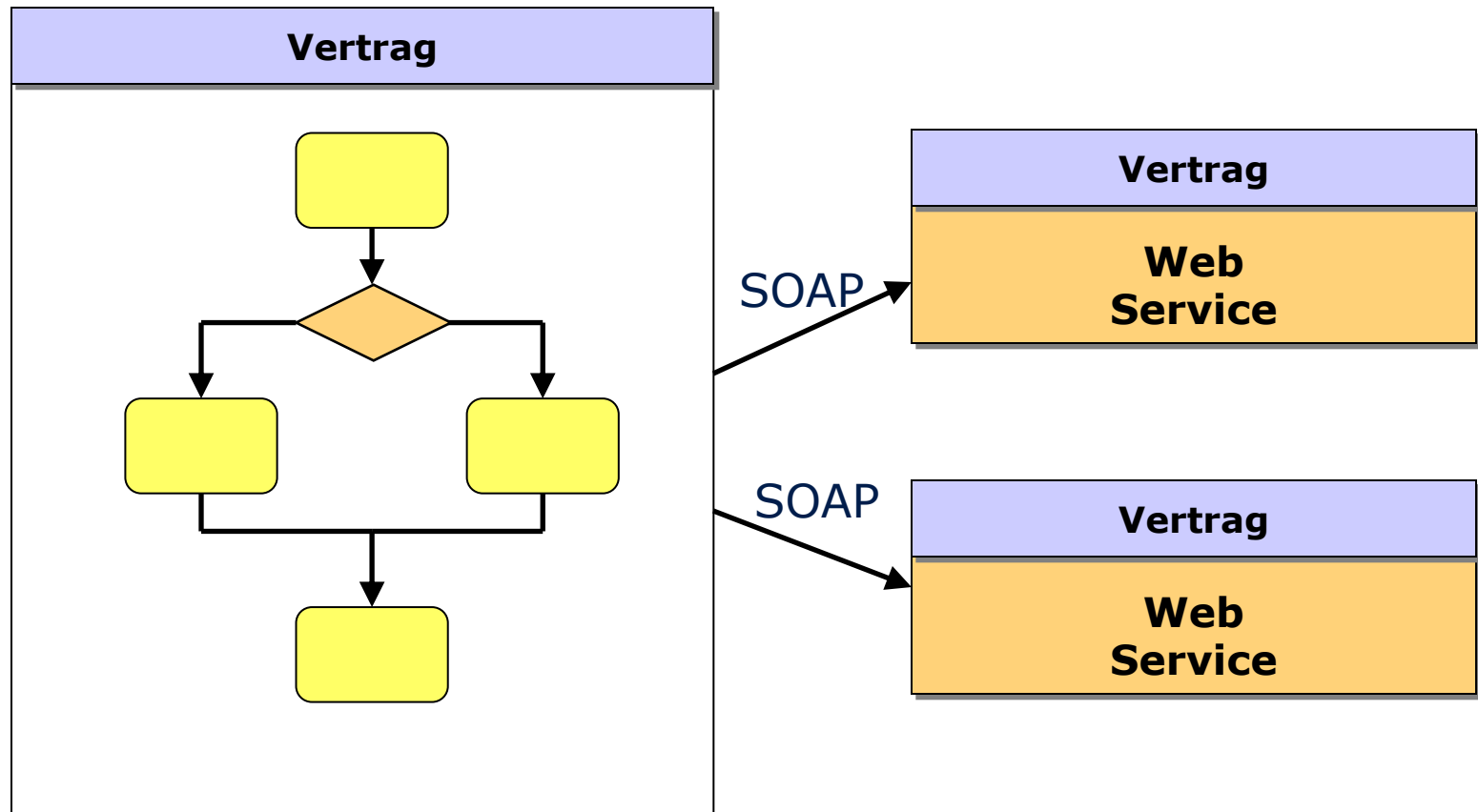
- Protokoll legt Sichtbarkeit von Nachrichtenaustausch-Verhalten fest, ohne interne Implementierungsdetails offenzulegen
- Vorteile dieser Aspekt-Trennung:
  - Firmen wollen interne Geschäftsprozesse und ihr Datenmanagement Geschäftspartnern nicht offenbaren
  - Freiheit, private Details von interner Prozessimplementierung ändern zu können, ohne dass die öffentliche Schnittstelle davon beeinflusst wird



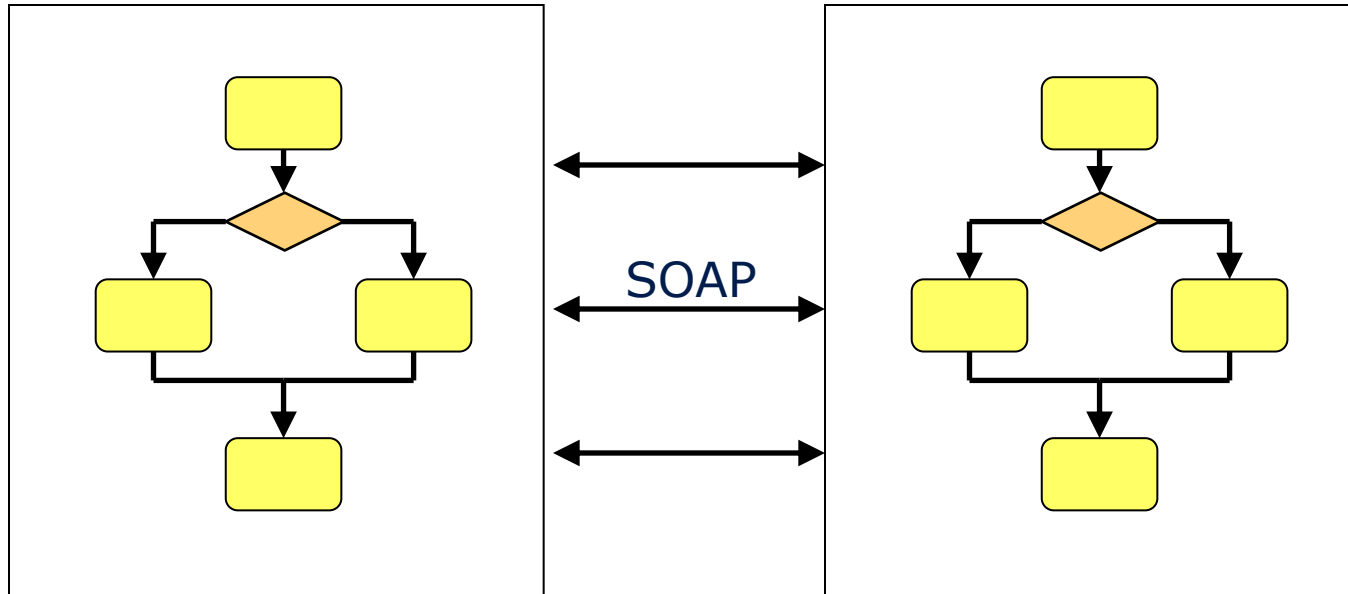
- Interne Aktivitäten für andere Teilnehmer nicht sichtbar



## Orchestrierung:



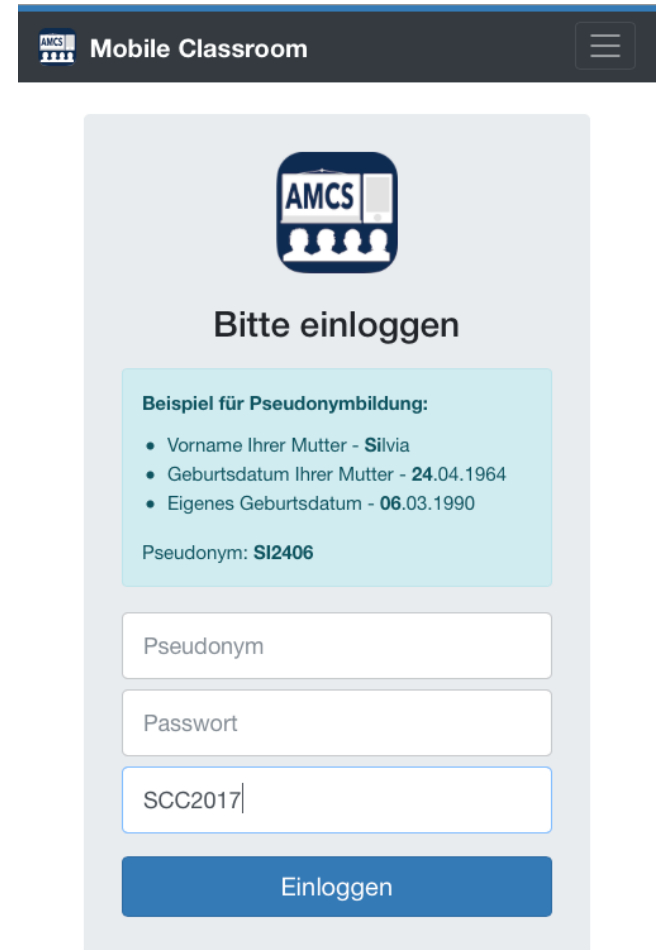
## Choreographie:






Web-Zugang: <https://amcs.website>

PIN: SCC2023



AMCS Mobile Classroom



## Bitte einloggen

**Beispiel für Pseudonymbildung:**

- Vorname Ihrer Mutter - Silvia
- Geburtsdatum Ihrer Mutter - 24.04.1964
- Eigenes Geburtsdatum - 06.03.1990

Pseudonym: SI2406

Pseudonym

Passwort

SCC2017

Einloggen



# Vorlesung "SCC – Service and Cloud Computing"

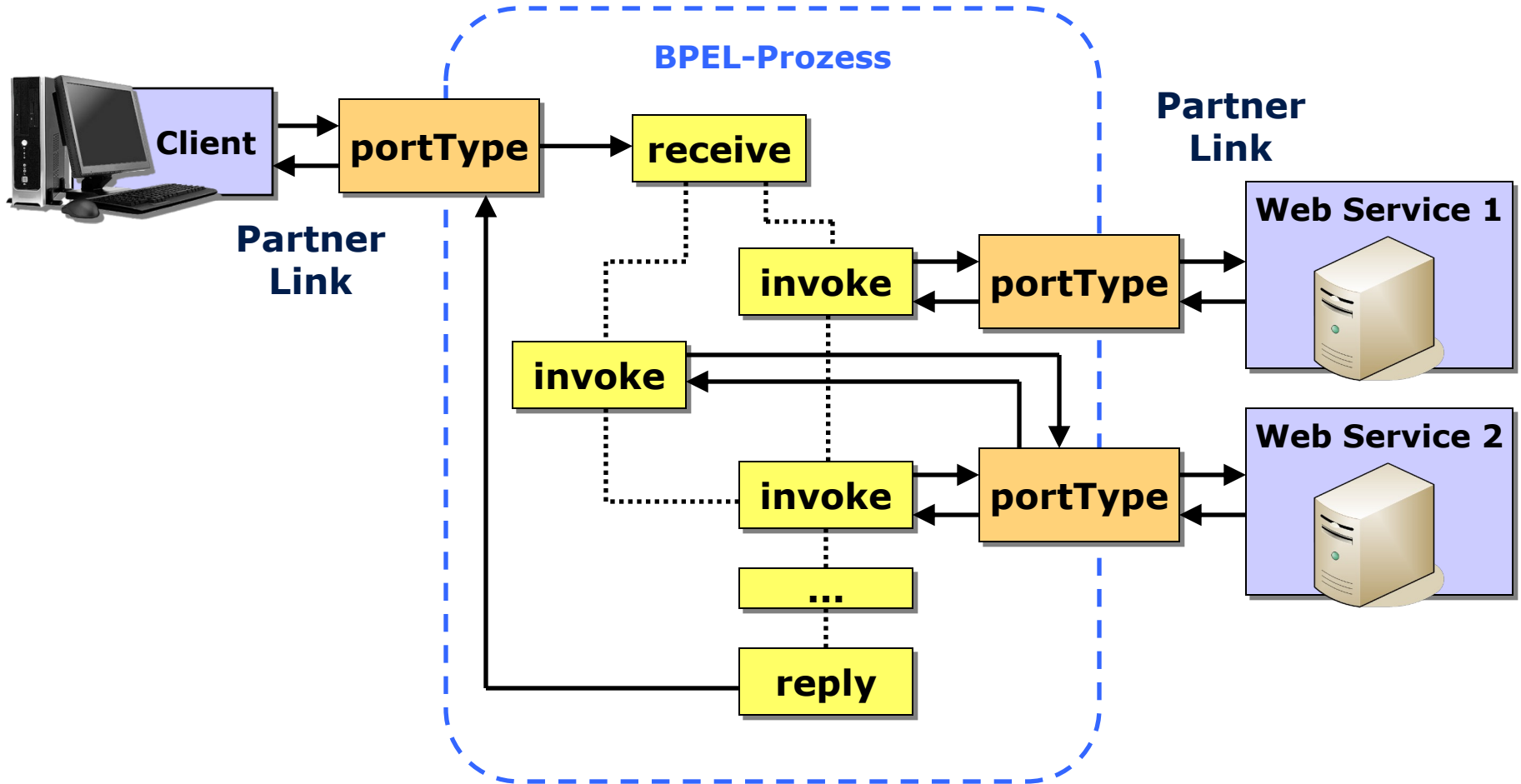
## 5. SaaS - Entwicklung und Komposition von Diensten

5.3 Kompositionssprachen (BPEL, BPMN)

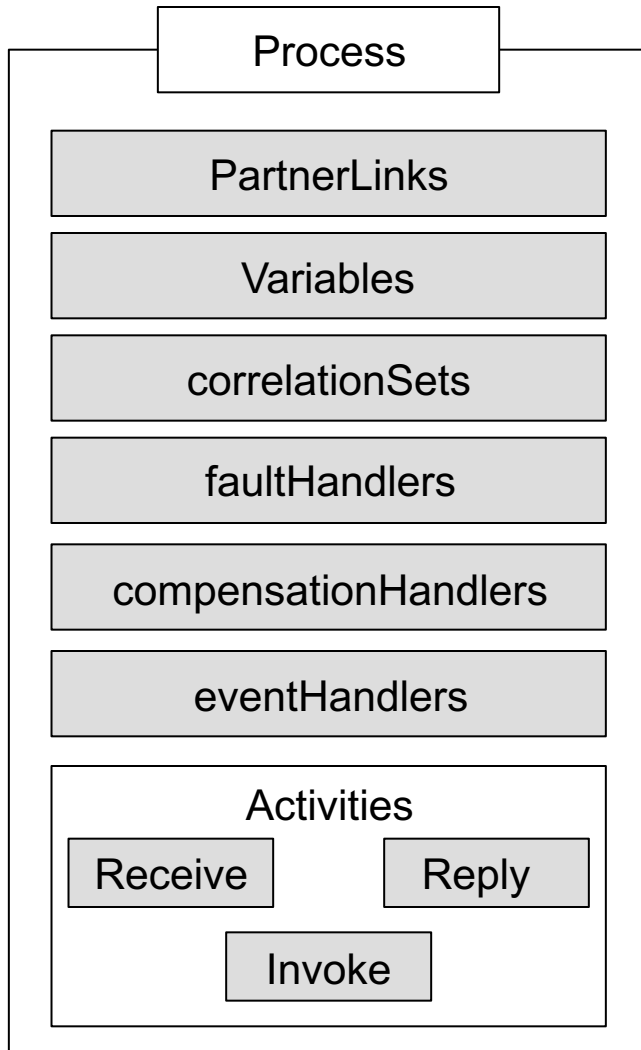
Dr.-Ing. Iris Braun

- Wiederholung Software as a Service (SaaS)
- Konzeption serviceorientierter Anwendungen
  - Abbildung von Geschäftsprozessen
  - Entwicklung von Diensten
  - Integration
- Modellierung von Geschäftsprozessen
- Komposition – Motivation, Begriffsdefinitionen
- Choreographie vs. Orchestration
- **Kompositionssprachen**
  - BPEL
  - BPML
- **Server-Komponenten**
- **Design-Werkzeuge**

- 2002 von Microsoft, IBM und BEA vorgestellt als BPEL4WS
- 2003 Release 1.1 zur Standardisierung an OASIS übergeben
- 2007 Version 2.0 standardisiert, aber inkompatibel zu 1.1
  
- Abbildung von Geschäftsprozessen auf Basis von Web Services
- BPEL-Prozess wird durch Interaktion der WSDL-portTypes realisiert
- Aktivität = ein Prozessschritt innerhalb des Gesamtprozesses
- WSDL-Datei beschreibt Sicht von außen auf komplexen WS
  - Beschreibung möglicher Rollen
  - Port mit aufrufbaren Operationen
- BPEL-Datei beschreibt Ablauf des Geschäftsprozesses
  - Reihenfolge der Aktivitäten
  - Randbedingungen
  - Fehlerbehandlungen











- **Executable** Process:
  - Orchestrierung spezifischer Aktivitäten und bestimmter Dienste, die ausgeführt werden müssen
  - Über eine Execution Engine ausführbar
  - ⇒ Definition einer **Orchestrierung**
  
- **Abstract** Process:
  - Spezifikation des Nachrichtenaustauschs mehrerer Teilnehmer
  - Keine Definition der internen Prozess-Details
  - Schnittstelle definiert durch Menge aller Receive und Reply
  - ⇒ Definition einer **Choreographie**



```

<process name=„SellThings“ targetNamespace=...
  xmlns=...   abstractProcess="no" >
  <!-- Definition und Rollen der Teilnehmer -->
  <partnerLinks> ... </partnerLinks>
  <!-- Daten/Zustand des Prozesses -->
  <variables> ... </variables>
  <!-- Eigenschaften, die Konversationen definieren -->
  <correlationSets> ... </correlationSets>
  <!-- Exception-Handling -->
  <faultHandlers> ... </faultHandlers>
  <!-- Fehler-Recovery - Aktionen rückgängig machen -->
  <compensationHandlers> ...</compensationHandlers>
  <!-- Ereignisse mit eigenem Prozess -->
  <eventHandlers> ... </eventHandlers>
  <!-- Business Process Flow: Mindestens eine
  Activity -->
  <!-- (Activities)+ -->
</process>
  
```

-  **Receive:** Blockierendes Warten auf Nachrichten
-  **Invoke:** Aufrufen von Web-Service-Aktivitäten
-  **Reply:** Generierung einer Antwort auf empfangene Nachricht
-  **Wait:** Warten für eine bestimmte Zeit
-  **Assign:** Kopieren von Daten von einer Variable zu einer anderen
-  **Throw:** Fehlerbehandlung (Erzeugung einer Fehlernachricht)
-  **Terminate:** Beenden der kompletten Prozessinstanz
-  **Empty:** Einfügen von NOOP (no operations)



**Sequence:** Reihenfolge von Schritten



**Flow:** Definition parallel abzuarbeitender Prozessschritte



**Switch:** Definition von Verzweigungen auf Basis von Fallunterscheidungen



**While:** Definition von Schleifen

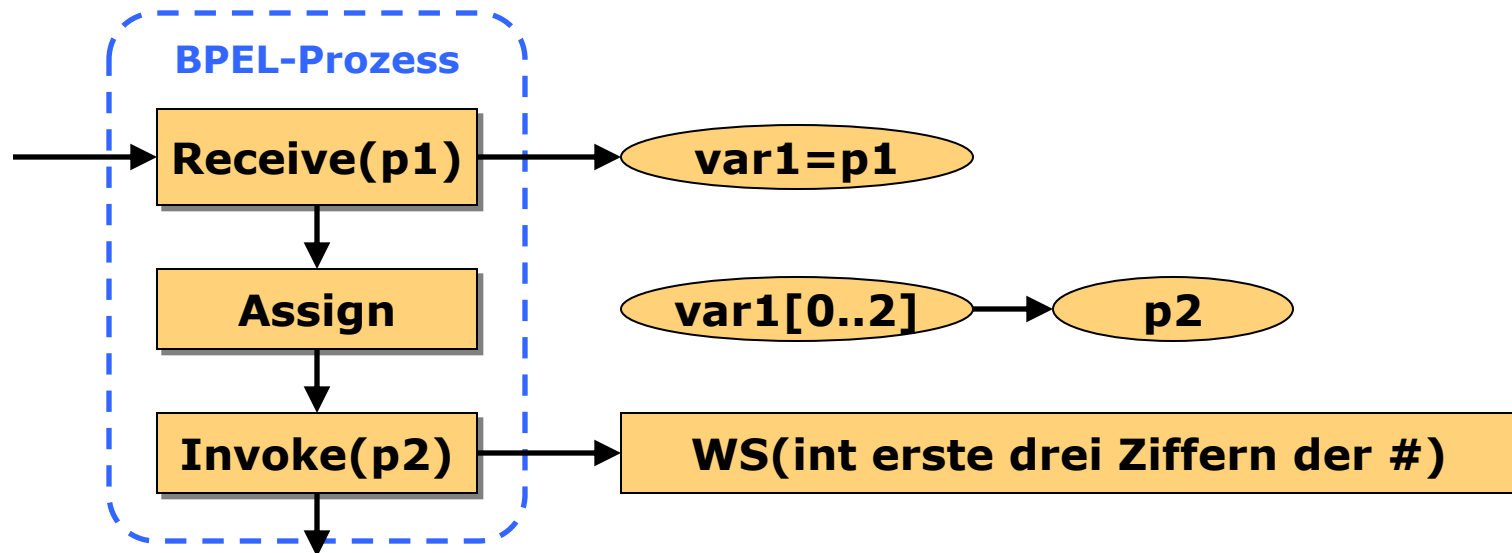


**Pick:** Blockierendes Warten auf eine passende Nachricht oder Timeout, abhängig davon Auswahl der nächsten Aktivität

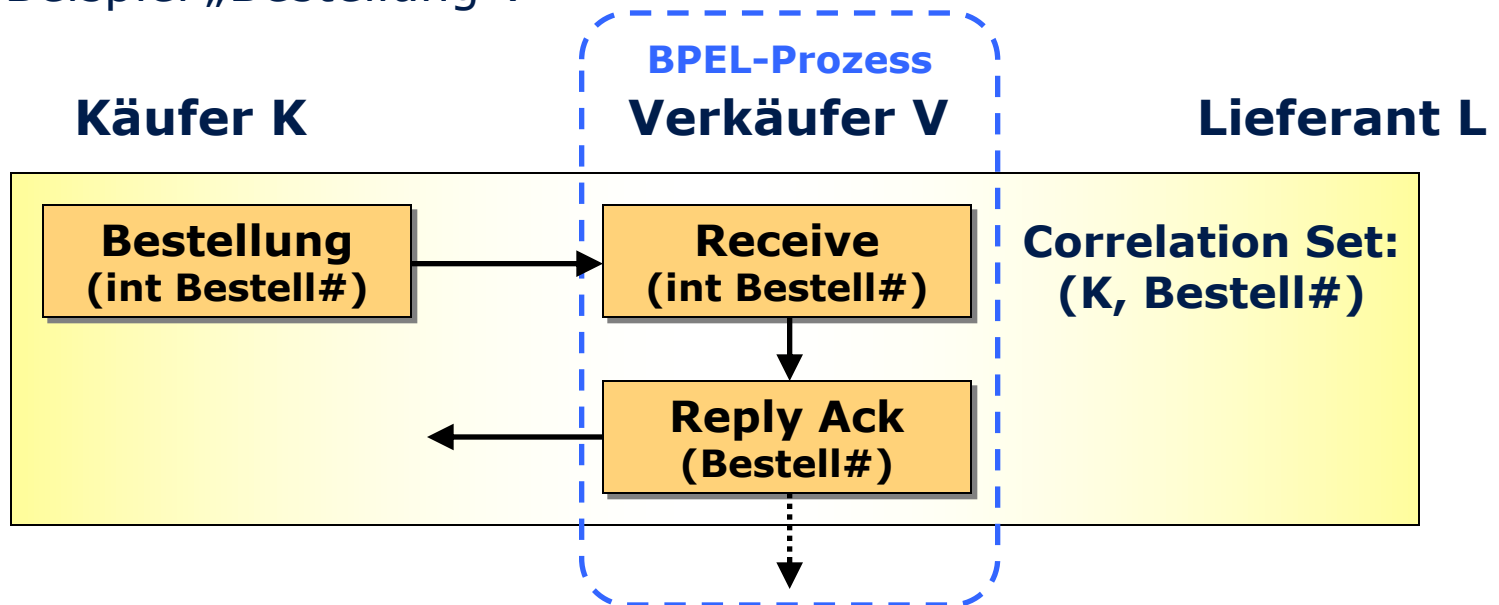


**Scope:** Gruppieren einer Menge von Aktivitäten zu Transaktion

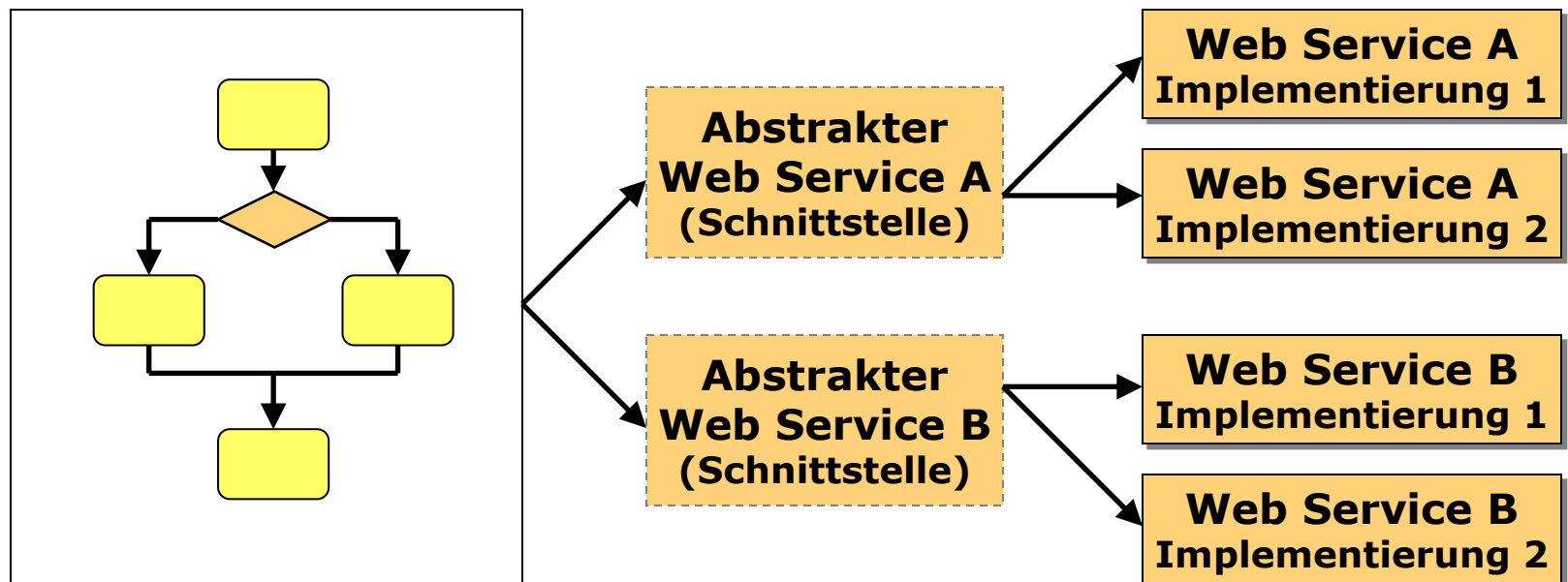
- Variablen speichern Daten, die im Prozess ausgetauscht werden
- **Receive**: Prozess legt empfangene Daten in der zugehörigen Variable ab, folgende Aktivitäten können darauf zugreifen
- **Assign**: Erstellung neuer oder Änderung existierender Nachrichten mit Teilen anderer Nachrichten, XPath-Ausdrücken, XSLT-Ausdrücken oder Literalen (konstanten Werten)
  - ⇒ Verwendung empfangener Daten als Eingabe v. Operationen



- Geschäftsprozess: meist Konversationen mit mehreren Partnern
- Notwendig: Anwendungslogik, welche die Nachrichten der Konversationen dem jeweiligen Geschäftsprozess zuweist
- Correlation Sets: deklarative Beschreibung einer Gruppe von Eigenschaften, die zusammen die Konversation beschreiben
- Beispiel „Bestellung“:



- Aufruf von Web Services: nur über abstrakten portType, nicht über tatsächlich implementierte Deployments (Bindings)
- Erlaubt Wiederverwendbarkeit der Prozessdefinition bei verschiedenen Deployments kompatibler Dienste



1. WSDL-Messages definieren
  - Verwendete Nachrichten
2. WSDL-PortTypes definieren
  - Verfügbare Schnittstellen zu Diensten
3. PartnerLinkTypes definieren
  - Rollen der PortTypes
4. PartnerLinks definieren
  - verknüpfen WSDL-portTypes mit BPEL-Prozess-Definitionen
5. Verwendete Variablen definieren
6. Eigentlichen Prozess definieren

- Erste Aktivität eines Prozesses immer Receive
- Strukturierung der Aktivitäten mit Sequence, Flow, Switch usw.
- Letzte Aktivität meist Reply, um die Antwort zurückzugeben

```
<sequence>
```

```
  <receive partnerLink="kaufen" portType="lns:purchaseOrderPT"  
    operation="bestellungSenden" variable="Bestellung">
```

```
  </receive>
```

```
  <sequence>...</sequence>
```

```
  <flow>...</flow>
```

```
  <reply partnerLink="kaufen" portType="lns:purchaseOrderPT"  
    operation="bestellungSenden" variable="Rechnung">
```

```
  </reply>
```

```
</sequence>
```

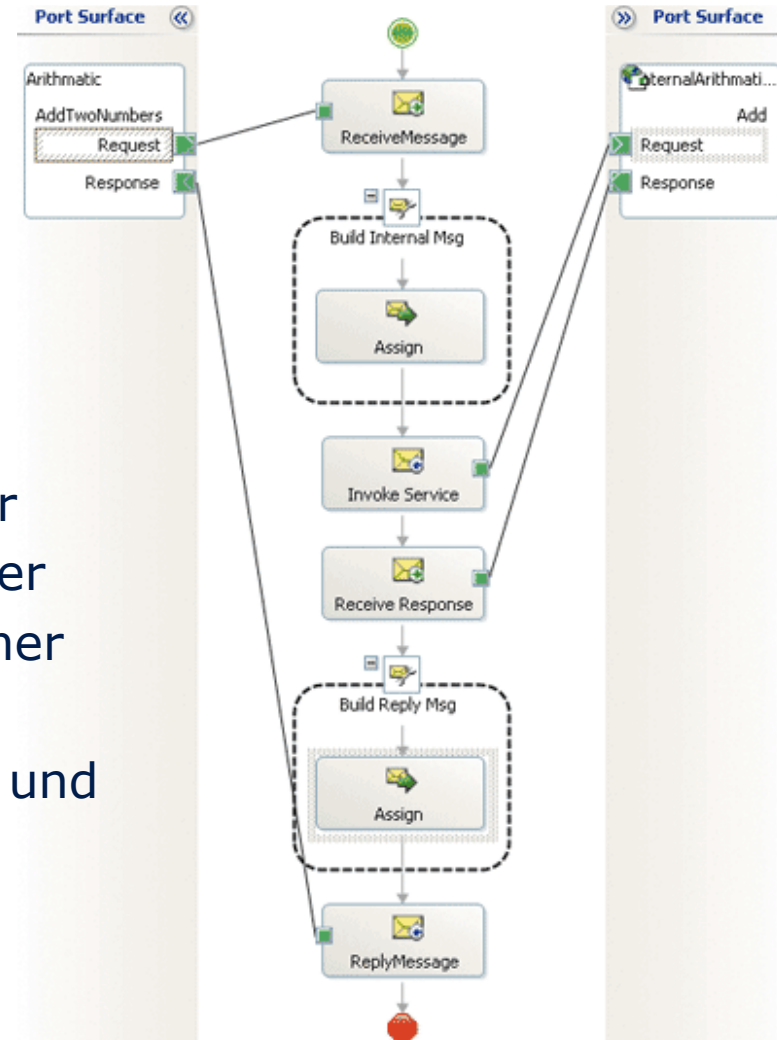
➤ **BPEL-Engine:** Laufzeitumgebung zur Interpretation des BPEL-Dokumentes und Ausführung des Prozesses

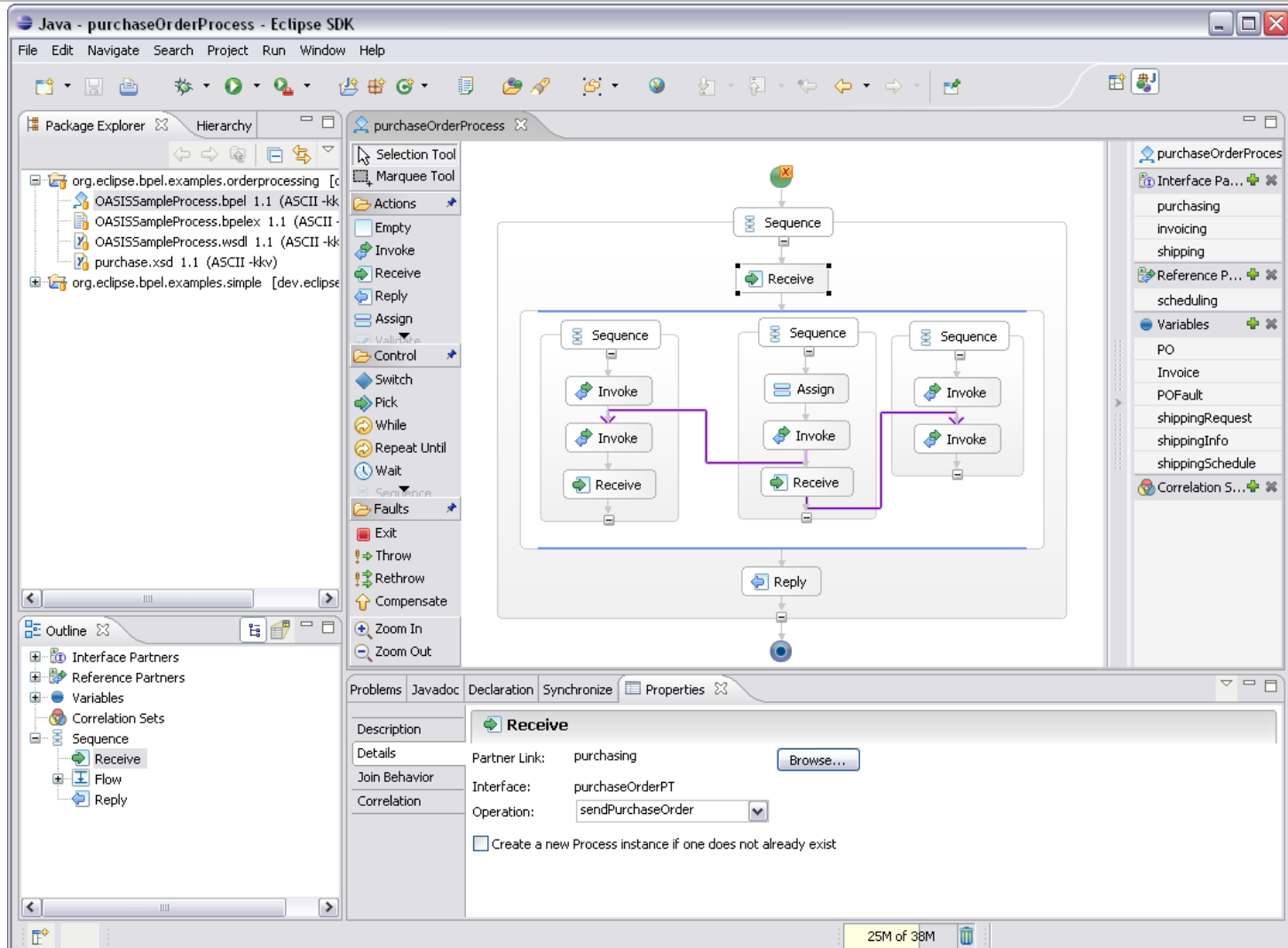
- Apache ODE (Orchestration Director Engine)
- BPEL SE (Bestandteil des OpenESB)
- jBPM (Bestandteil des JBoss Application Servers)
  
- ActiveVOS (früher ActiveBPEL von ActiveEndpoints)
- IBM WebSphere Process Server + Choreographer
- Microsoft BizTalk Server
- Oracle BPEL Process Manager (früher Collaxa BPEL Server)
- Intalio n3 Server
- ...



➤ Bieten grafische Unterstützung beim Design von BPEL-Workflows

- Eclipse BPEL Designer
- JBoss IDE 2.0
- NetBeans IDE 5.5
- IBM WebSphere Integration Developer
- IBM WebSphere Process Choreographer
- Microsoft BizTalk Orchestration Designer (siehe Bild)
- Oracle BPEL Designer (für JDeveloper und Eclipse)
- Active Endpoints ActiveVOS Designer
- ...





The screenshot displays the Eclipse BPEL Designer interface for a project named 'purchaseOrderProcess'. The main workspace shows a BPEL process diagram with the following structure:

- Start:** A 'Sequence' block containing a 'Receive' activity.
- Flow:** Three parallel 'Sequence' blocks are connected to the 'Receive' activity.
- Block 1 (Left):** Contains an 'Invoke' activity, followed by another 'Invoke' activity, and a 'Receive' activity.
- Block 2 (Middle):** Contains an 'Assign' activity, followed by an 'Invoke' activity, and a 'Receive' activity.
- Block 3 (Right):** Contains an 'Invoke' activity, followed by another 'Invoke' activity.
- End:** A 'Reply' activity is connected to the 'Receive' activities of all three parallel blocks.

The Properties view at the bottom shows the configuration for the selected 'Receive' activity:

| Receive       |                                                                                      |
|---------------|--------------------------------------------------------------------------------------|
| Description   |                                                                                      |
| Details       | Partner Link: purchasing <a href="#">Browse...</a>                                   |
| Join Behavior | Interface: purchaseOrderPT                                                           |
| Correlation   | Operation: sendPurchaseOrder                                                         |
|               | <input type="checkbox"/> Create a new Process instance if one does not already exist |

The Package Explorer on the left shows the project structure, including files like 'OASISSampleProcess.bpel', 'OASISSampleProcess.bpelx', 'OASISSampleProcess.wsdl', and 'purchase.xsd'. The Outline view on the bottom left shows the hierarchical structure of the BPEL process, including 'Interface Partners', 'Reference Partners', 'Variables', 'Correlation Sets', 'Sequence', 'Receive', 'Flow', and 'Reply'.

- Unterstützt verschiedene Arten der Komposition:
  - *executable processes* für Orchestration
  - *abstract processes* für Choreographie
  - *callbacks* und *correlation sets* für Konversation
- Breite Unterstützung durch Industrie, standardisiert bei OASIS
- Basiert auf vorhandenen WS-/XML-Standards, z.B. auf WSDL, XML Schema 1.0, XPath 1.0 und WS-Addressing
- Ermöglicht Prozessinteroperabilität
- Umfangreiche Kontrollfluss-Beschreibung, vielseitige Verschachtelung
- Synchron und asynchrone Nachrichtenübermittlung
- Unterstützung für Transaktionen (lang-andauernde mit Kompensation)
- Ausnahmebehandlung, Kompensationen und Fehlerbehandlung

Trotz der vorgestellten Ansätze:

- Komposition aufwändig und zeitraubend
- Erfordert Fachkenntnisse

Ursachen:

- Suche und Eignungsprüfung von Web Services (Matchmaking)
- Korrekte Verschaltung der Web-Service-Operationen komplex

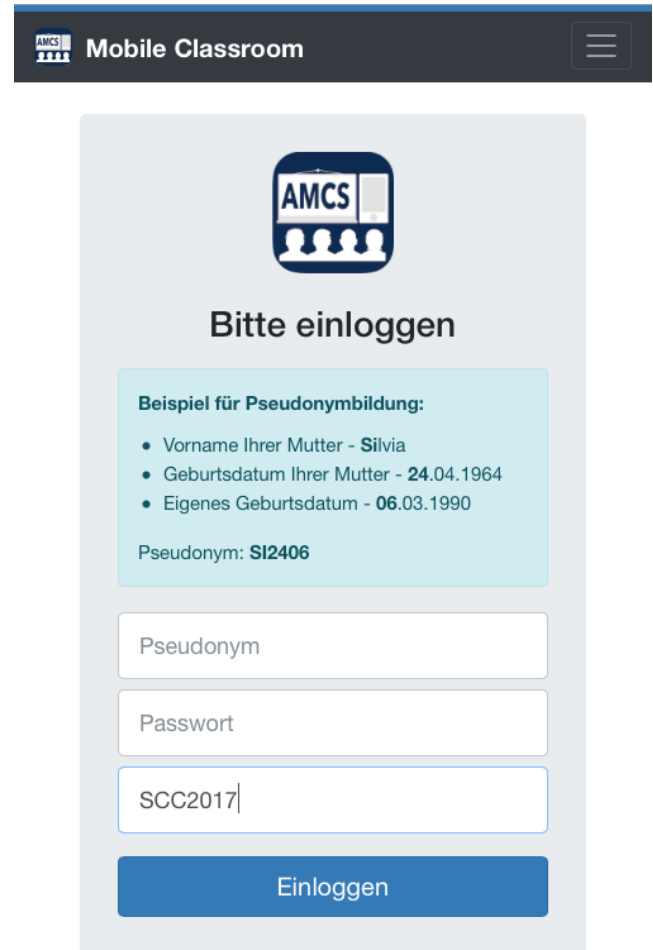
Mögliche Lösungen:

- Einbeziehung semantischer Beschreibung, automatisierte Kompositionsvorschläge mit verschiedenen Qualitätsmerkmalen (Kosten, Sicherheit, Verfügbarkeit, ...)
- Überwachung der Komponenten, bei Ausfall automatischer Einsatz passender Ersatzkomponenten




Web-Zugang: <https://amcs.website>

PIN: SCC2023



AMCS Mobile Classroom



## Bitte einloggen

**Beispiel für Pseudonymbildung:**

- Vorname Ihrer Mutter - Silvia
- Geburtsdatum Ihrer Mutter - 24.04.1964
- Eigenes Geburtsdatum - 06.03.1990

Pseudonym: SI2406

Pseudonym

Passwort

SCC2017

Einloggen