

Technische Universität München

Hauptseminar im SS2003

**Web Services**

# Workflow

Referentin: Bc. Hana Stetinova

Betreuer: Dipl.-Inf. Peter Fischer

# Inhaltsübersicht

## 1. Einführung

### 1.1 Workflow

#### 1.1.1 Modellierung von Workflows

### 1.3 Workflow Management System (WFMS)

## 2. Standardisierung von WFMS

## 3. Architektur von WFMS

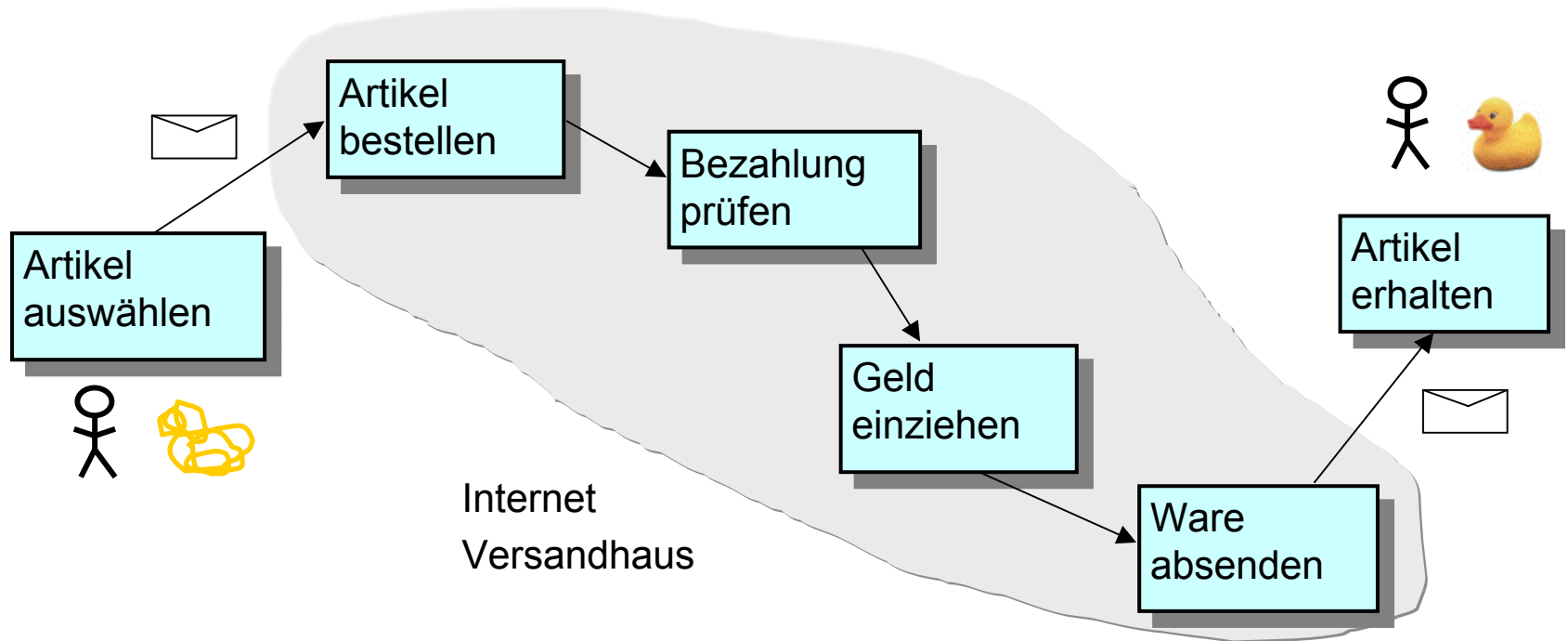
## 4. Implementierung

## 5. Ausblick auf Webservices

## 6. Zusammenfassung

# 1. Einführung

- Komplexität der Geschäftsprozesse steigt
  - Automatisierung und Rechnerunterstützung nötig
- ➔ Workflow Technologie



# 1.1 Was ist Workflow

## Definition von WFMC:

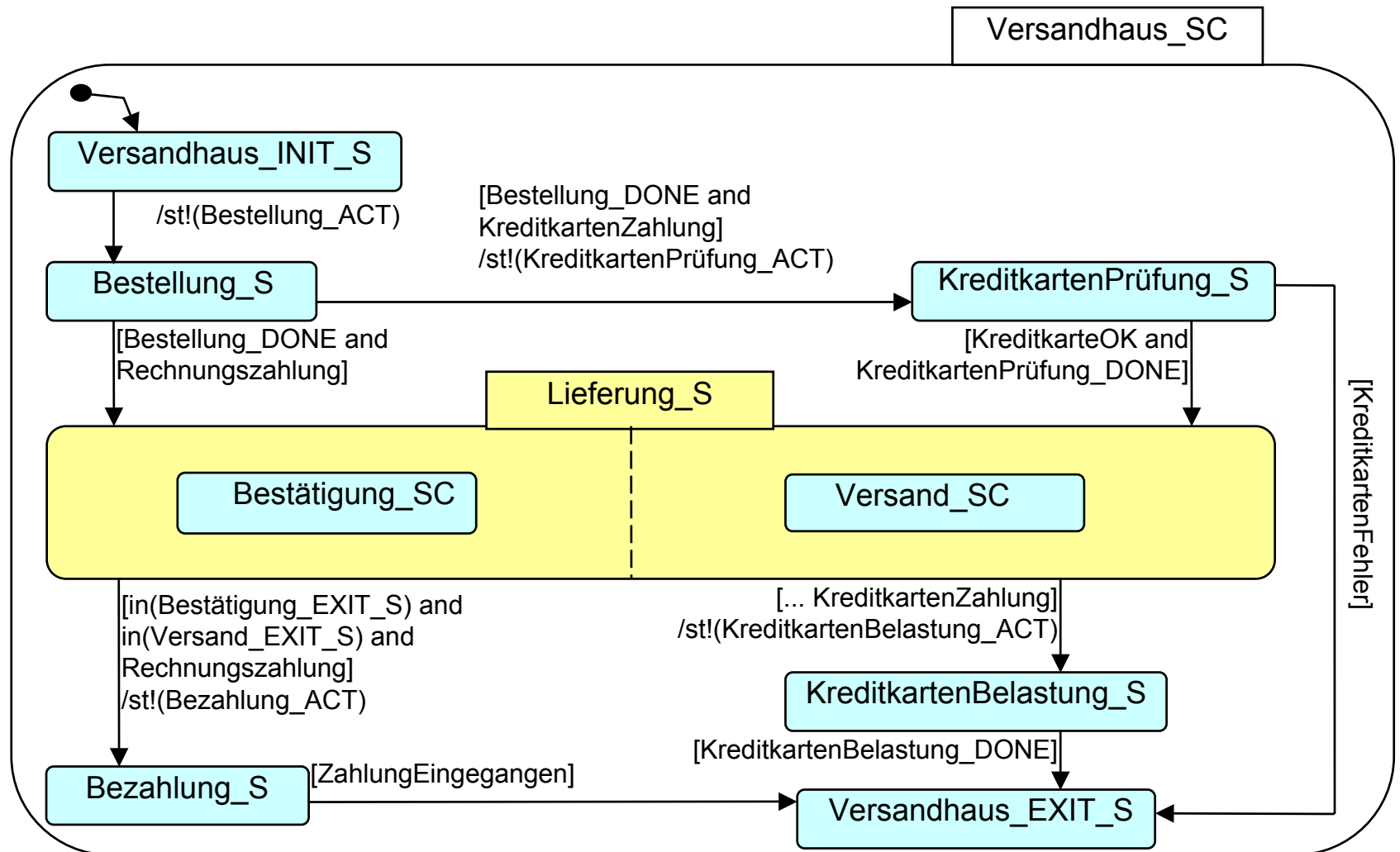
[A Workflow is] „the computerised facilitation or automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.”

**Workflow** = Satz von kooperierenden Aufgaben (tasks), Schritten (steps), Aktivitäten (activities) und Beziehungen zwischen diesen, mit Regeln (Bedingungen, Qualitätskriterien)

# 1.1.1 Modellierung von Workflows

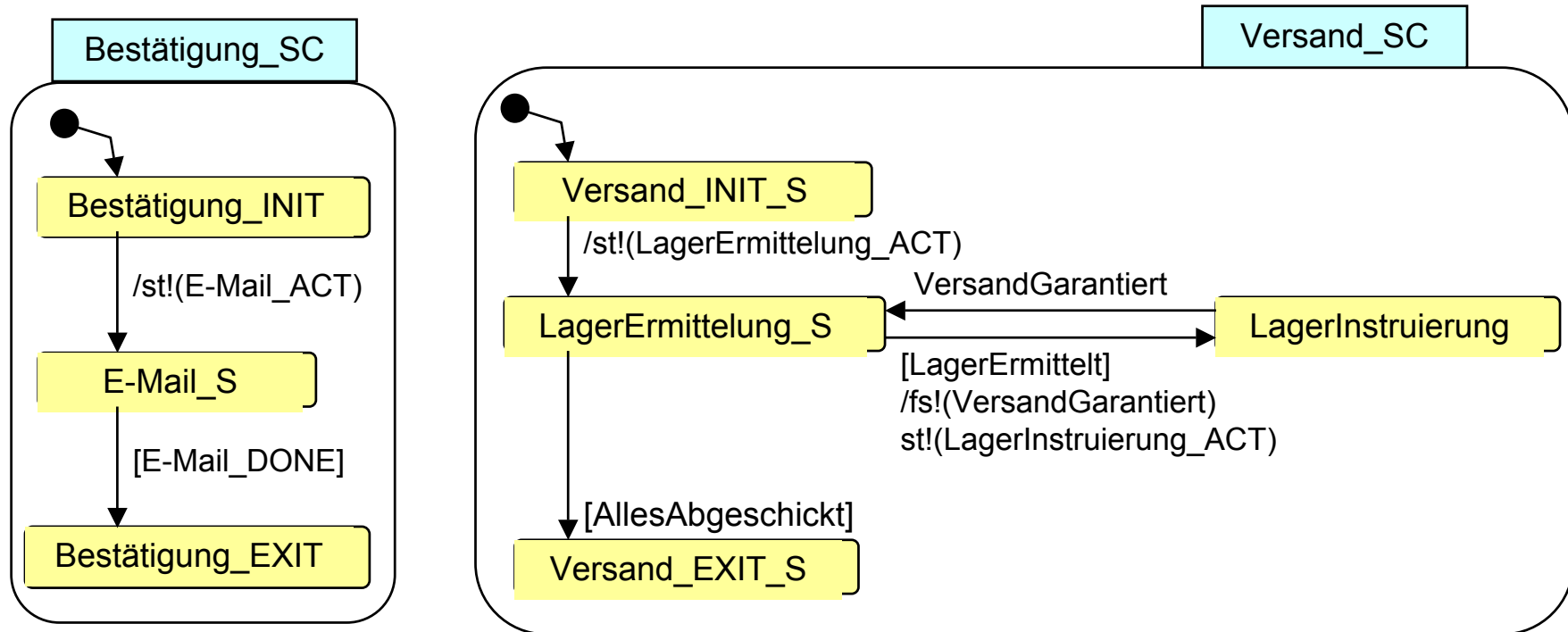
- reeller Geschäftsprozess mittels des Prozessdefinitions-Tools in eine durch den Computer verarbeitbare Form gebracht = Workflowmodell (Spezifikation)
- Modell = Netz von (verteilten) Aktivitäten und den Beziehungen zwischen diesen
- Beispiel: Statecharts, Aktivitätscharts, temporale Logik, script Sprachen, Prozess Algebra, Petri-Netz Varianten
- Statecharts und Aktivitätscharts in UML Industrie Standard der OMG

# 1.1.1a Statechart

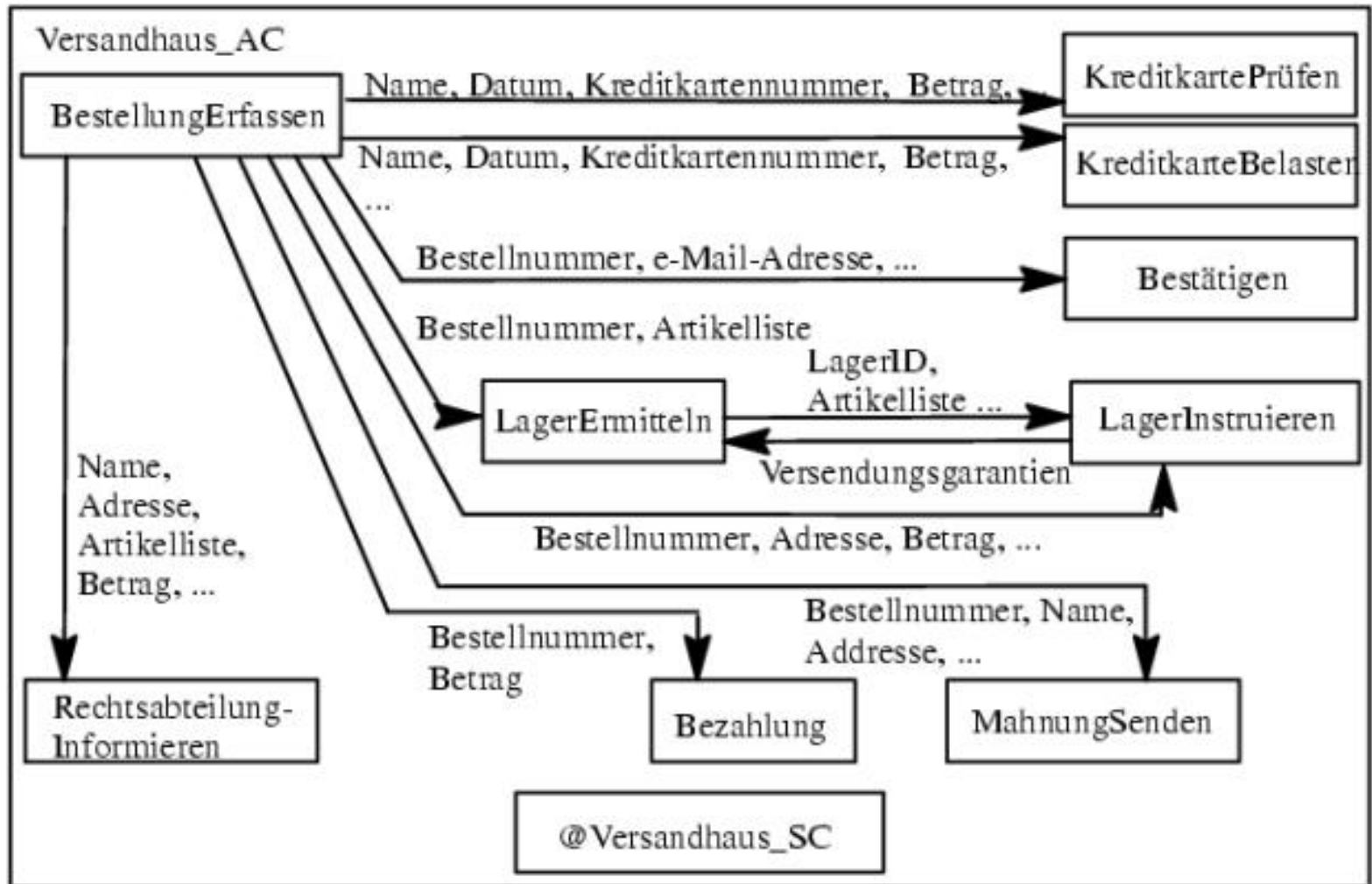


# 1.1.1b Statechart (cont.)

- Verfeinerung von Statecharts



# 1.1.1c Aktivitätschart

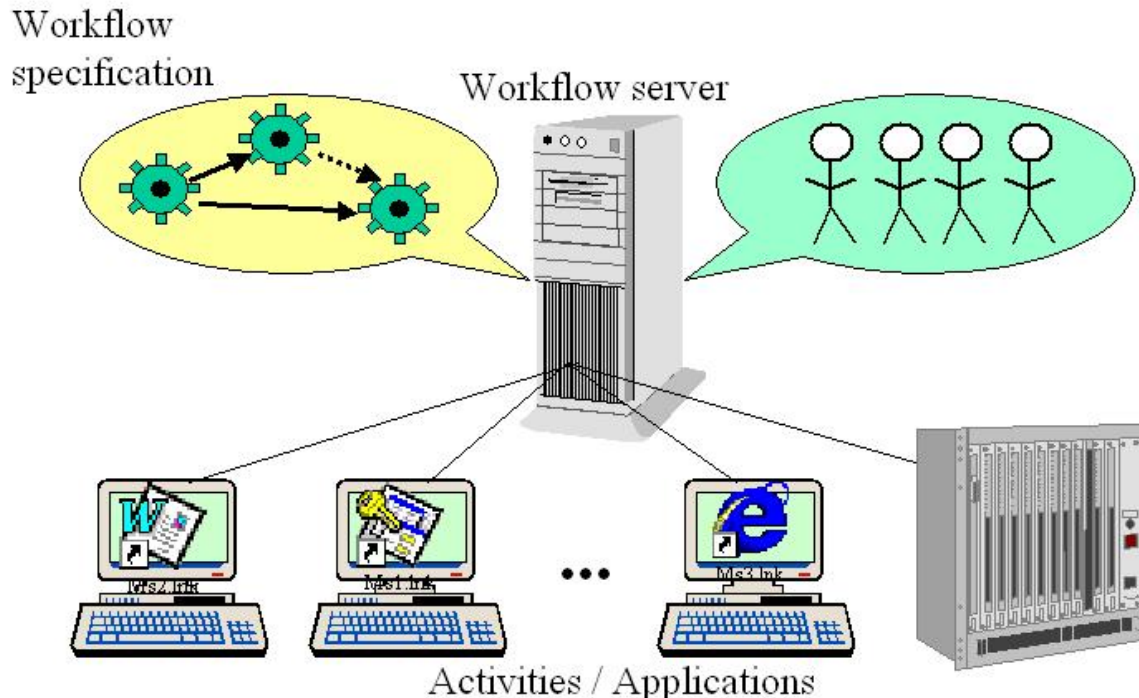




# 1.2 Workflow Management System

**WFMS** = Software System, koordiniert die Arbeitsschritte zwischen den Benutzern und Systemkomponenten, überprüft die Qualitätskriterien (Bedingungen)

- aus Büroautomatisierung und Groupware in 80ern



# Inhaltsübersicht

## 1. Einführung

## 2. Standardisierung von WFMS

- WFMC
- OMG

### 2.1 Model Reference Architektur von WFMC

## 3. Architektur von WFMS

## 4. Implementierung

## 5. Ausblick auf Webservices

## 6. Zusammenfassung

## 2. Standardisierung

**Problem:** Systeme inkompatibel

**Lösung:** Standardisierung

### **Workflow Management Coalition (WFMC)**

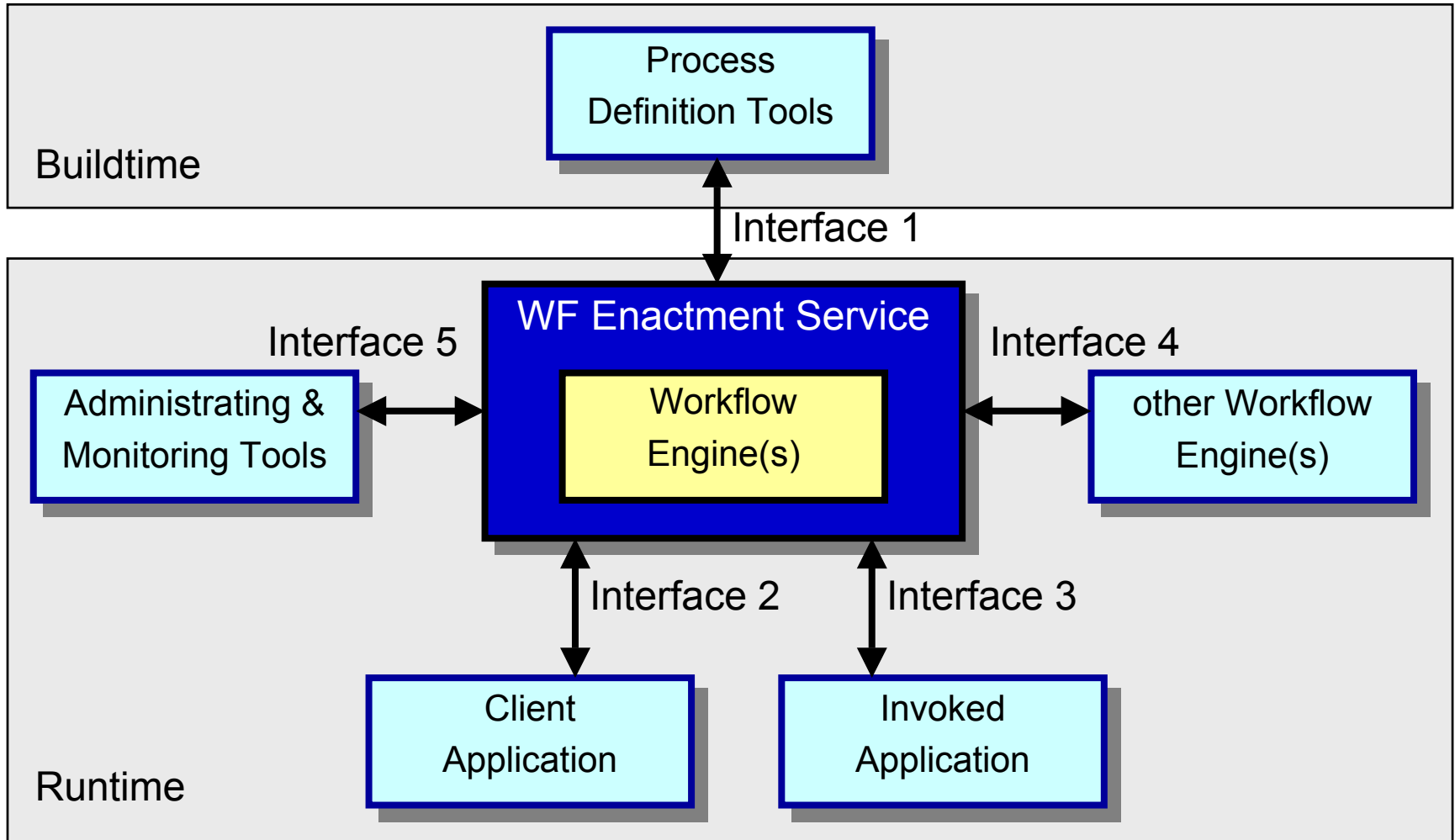
- gegr. 1993 als non-Profit Interessenverband von WFMS Anbietern und Anwendern
- Ziel: Standardisierung und Förderung von WF Technologie
- Reihe von Schnittstellen eingeführt, damit Module von unterschiedlichen Herstellern ersetzbar

### **Object Management Group (OMG)**

- Ziel: Verbreitung von Objekttechnologie
- CORBA als plattformunabhängige Infrastruktur für verteilte Objekte (1995)

# 2.1 Model Reference Architektur (WFMC)

## Model Reference Architektur von WFMC



# Inhaltsübersicht

1. Einführung
2. Standardisierung von WFMS
3. **Architektur von WFMS**
  - 3.1 Zentralisierte Architektur
  - 3.2 Verteilte Architektur
  - 3.3 Bewertung
4. Implementierung
5. Ausblick auf Webservices
6. Zusammenfassung

# 3. Architektur von WFMS

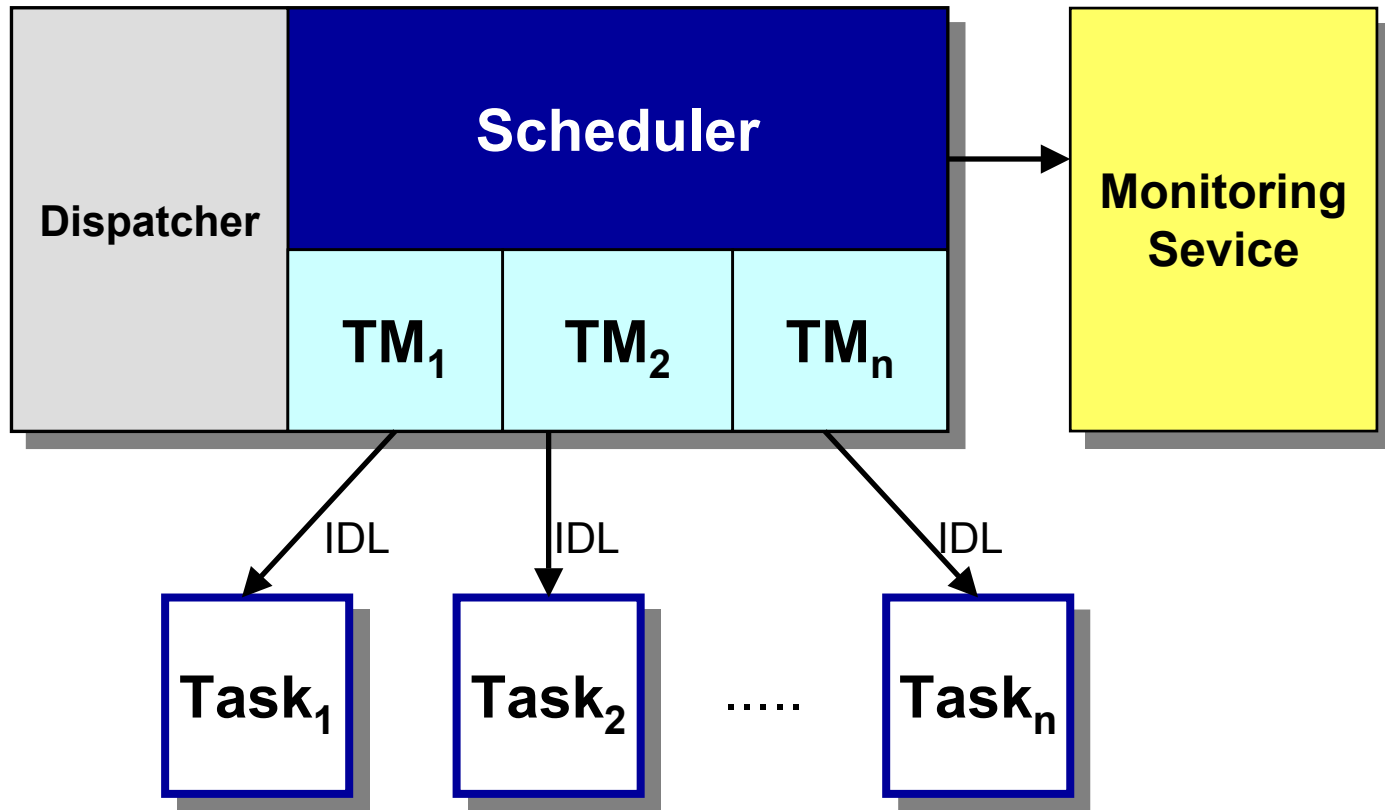
- Workflow Enactment Service kann auch aus mehreren, verteilten Engines bestehen
- Zentralisiert bis zum völlig verteilt
- auch Misch-Architekturen (z.B. semi-verteilte...)

## Hauptkomponenten:

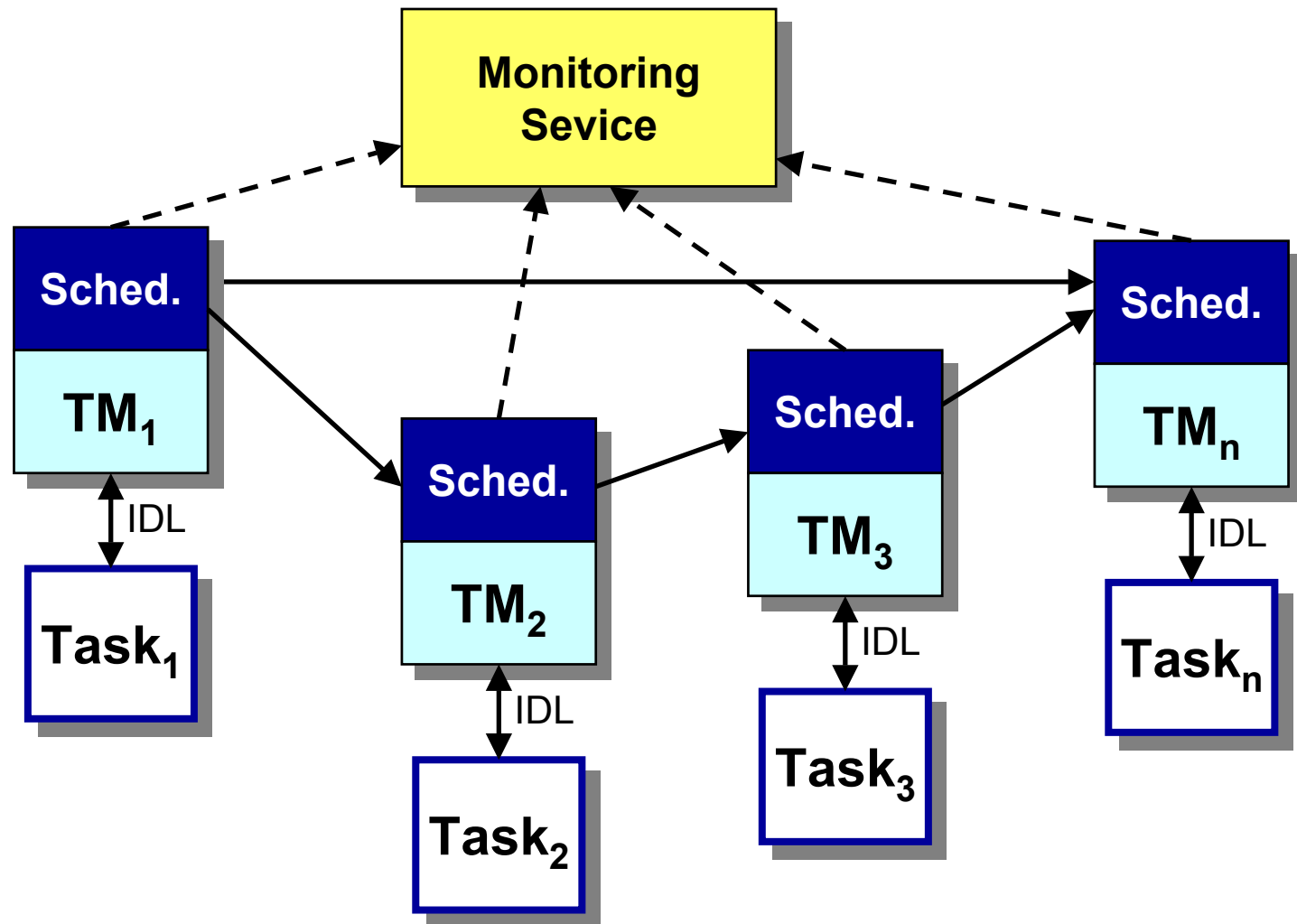
- Scheduler
- Task Managers (TM)
- Tasks
- Monitoring Service

# 3.1 Zentralisierte Architektur

- nur Threads für Task Managers (TM)



## 3.2 Verteilte Architektur





# 3.3 Bewertung

## Pros und Contras

### hoch zentralisierte Architektur

- Flaschenhals Problem von Scheduler
- beim Ausfall total paralysiert
- + leichter testbar und administrierbar

### völlig verteilte Architektur

- + hohe Skalierbarkeit und Performance
- + bei Knotenausfall nur Teil des WF beeinflusst
- hohe Komplexität

➡ nicht eindeutig, was besser ist

# Inhaltsübersicht

1. Einführung
2. Standardisierung von WFMS
3. Architektur von WFMS
4. Implementierung
  - 4.1 CORBA
  - 4.2 Java-basierte Architektur
  - 4.3 XML-basierte Architektur
    - XML Mediator
5. Ausblick auf Webservices
6. Zusammenfassung

# 4. Implementierung von WFMS

**Wie wird die Kommunikation zwischen verteilten Komponenten realisiert?**

- **CORBA**

Standard der OMG

- **JAVA-basierte Architektur**

vorge stellt anhand eines Forschungsprototyps

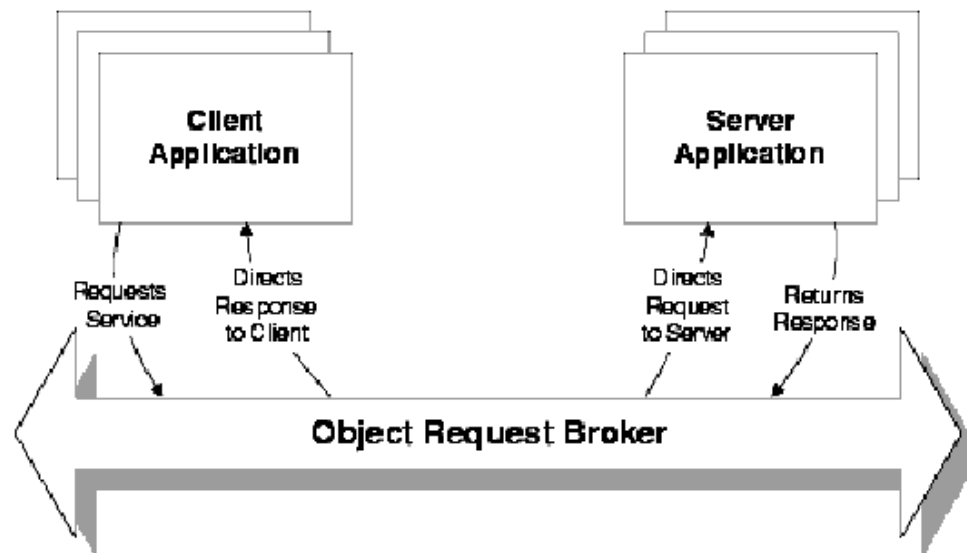
- **XML-basierte Architektur**

mit XML Mediator, vorge stellt anhand eines Forschungsprototyps

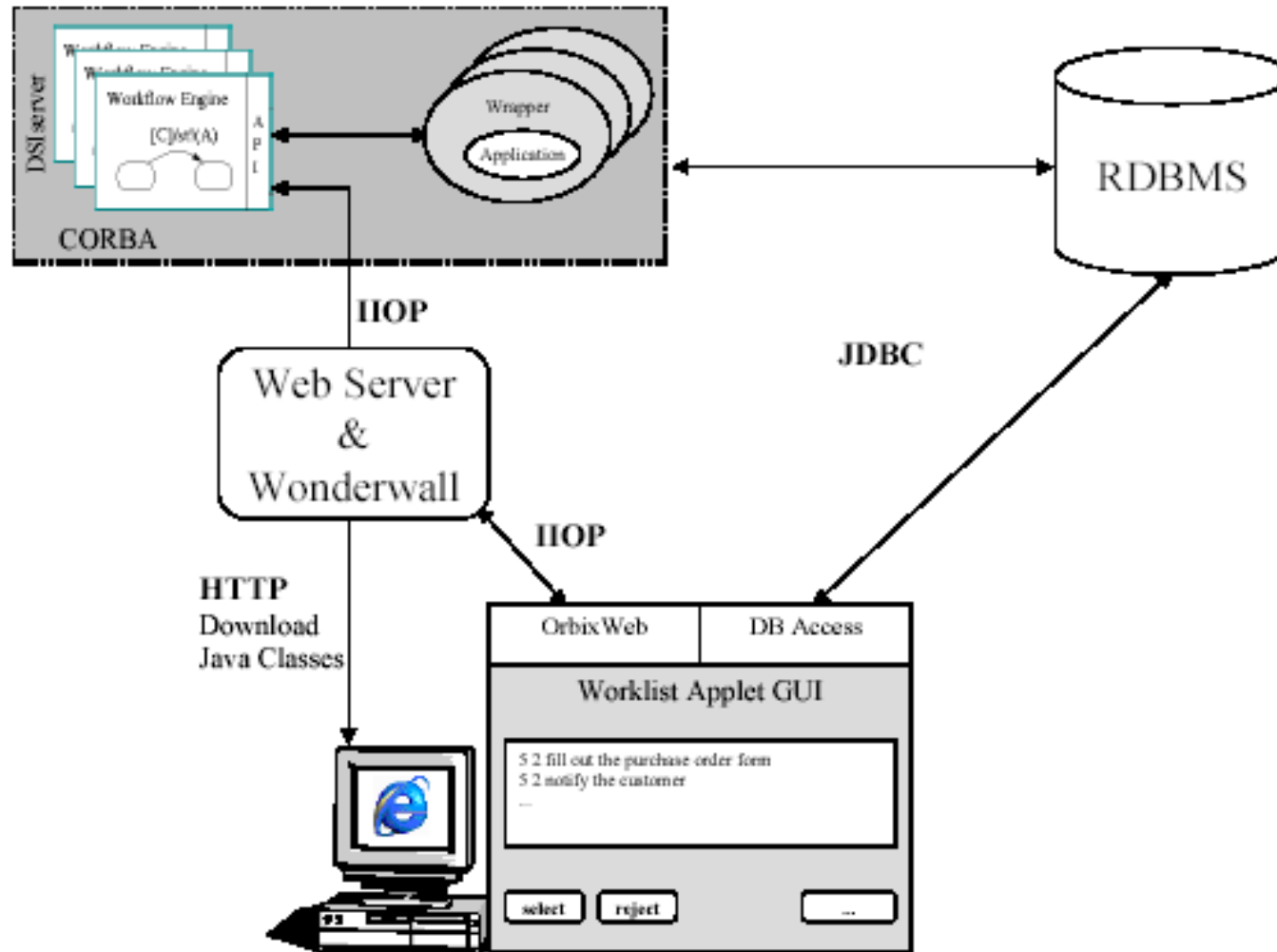
# 4.1 CORBA

## Common Object Request Broker Architecture (CORBA)

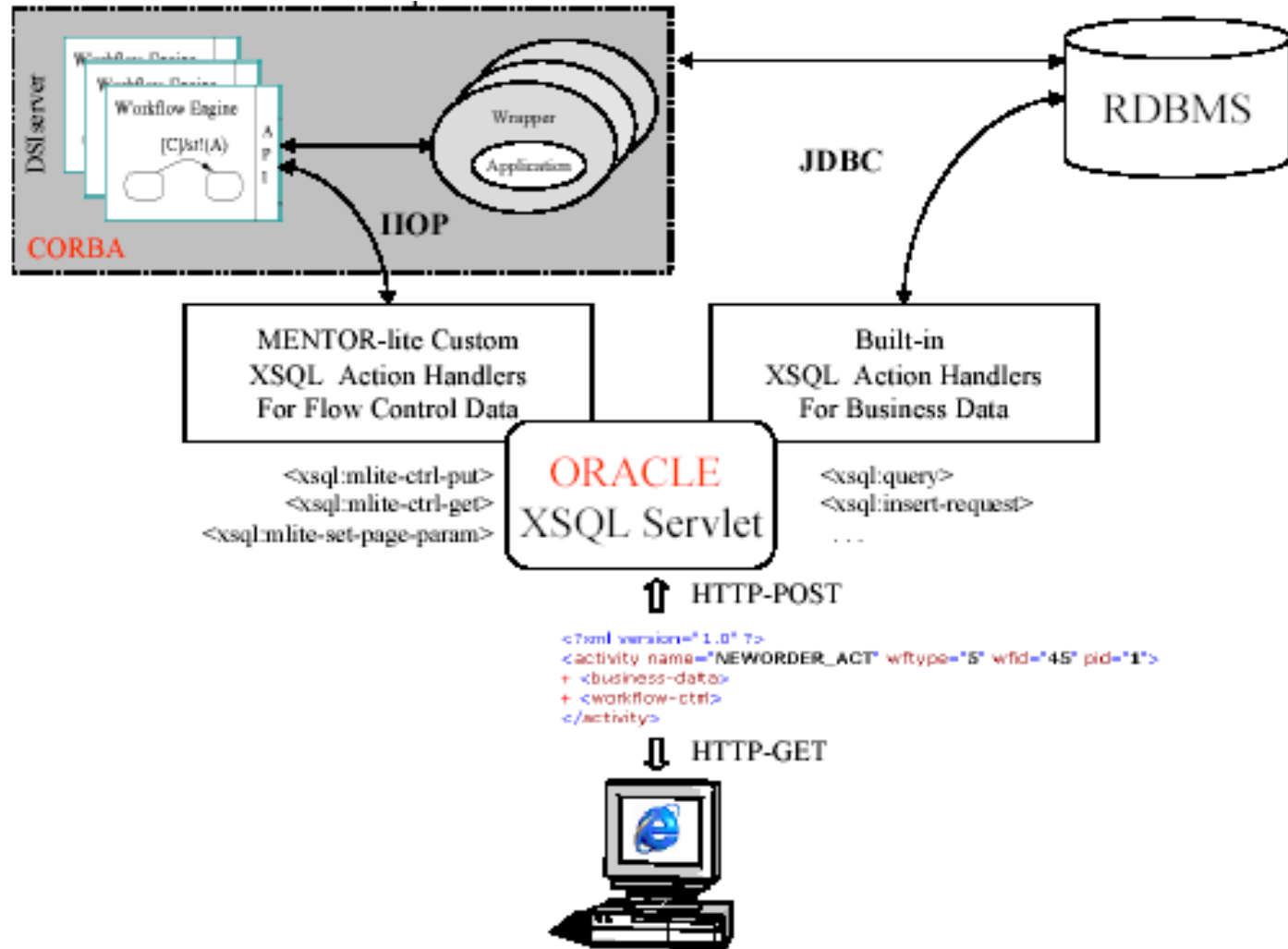
- ein klassischer Ansatz in verteilten Anwendungen
- separiert Client- und Server-Teil der Anwendung
- Remote Procedure Call (RPC) mit Objekten
- IIOP (Internet Inter ORB Protocol)



## 4.2 JAVA-basierte Architektur

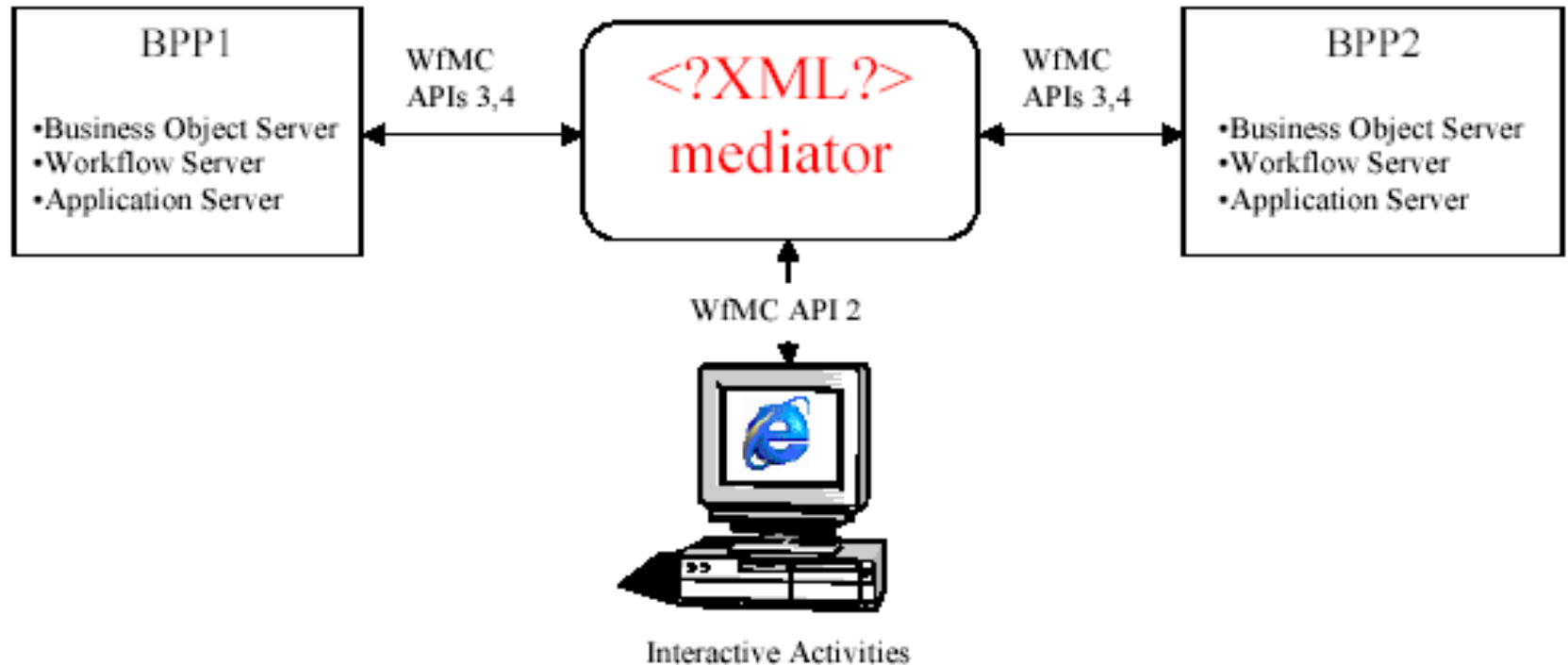


# 4.3 XML-basierte Architektur



## 4.3 XML (cont.)

### XML Mediator



BPP = Business Process Partner

# Inhaltsübersicht

1. Einführung
2. Standardisierung von WFMS
3. Architektur von WFMS
4. Implementierung
5. **Ausblick auf Webservices**
  - 5.1 WSFL
  - 5.2 XLANG
  - 5.3 BPEL4WS
  - 5.4 Bewertung
6. Zusammenfassung



# 5. Ausblick auf Web Services

- Web Services immer komplexer -> Bedarf nach Workflow
- noch kein Standard von W3C
- Streitereien zwischen verschiedenen Herstellern:

- SUN, SAP, BEA ...

**WSCI** - Web Services Choreography Language

- Microsoft, IBM ...

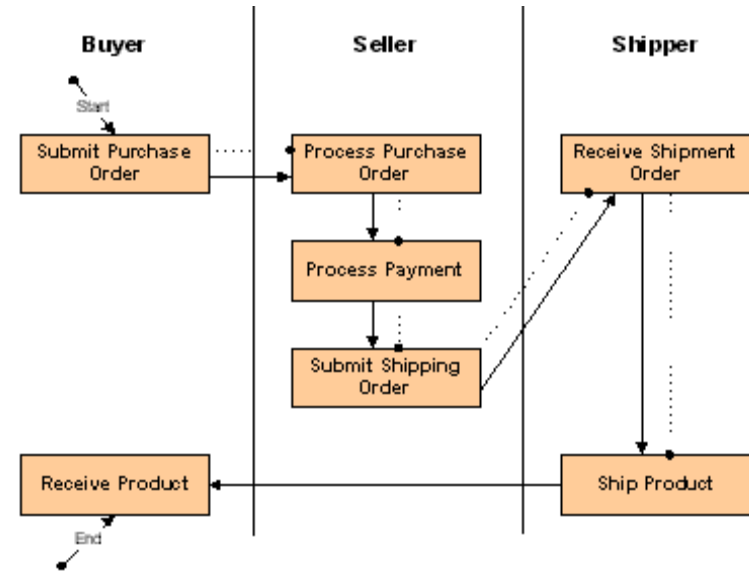
**WSFL** - Web Services Flow Language (IBM)

**XLANG** (Microsoft) WSDL basiert

**BPEL4WS** (IBM + Microsoft)

# 5.1 WSFL- Web Services Flow Language

```
<flowModel name="totalSupplyFlow" serviceProviderType="totalSupply">
  <serviceProvider name="seller" type="seller" />
  <serviceProvider name="shipper" type="shipper" />
  <activity name="submitShippingOrder">
    <performedBy serviceProvider="seller" />
    ...
    <target portType="totalSupplyPT"
      operation="submitShippingOrder" />
    ...
  </activity>
  <activity name="receiveShippingOrder">
    <performedBy serviceProvider="shipper" />
    ...
    <target portType="totalSupplyPT"
      operation="receiveShippingOrder" />
    ...
  </activity>
  <controlLink source="submitShippingOrder" target="receiveShippingOrder" />
  <dataLink source="submitShippingOrder" target="receiveShippingOrder"> ...
  </dataLink>
</flowModel>
```

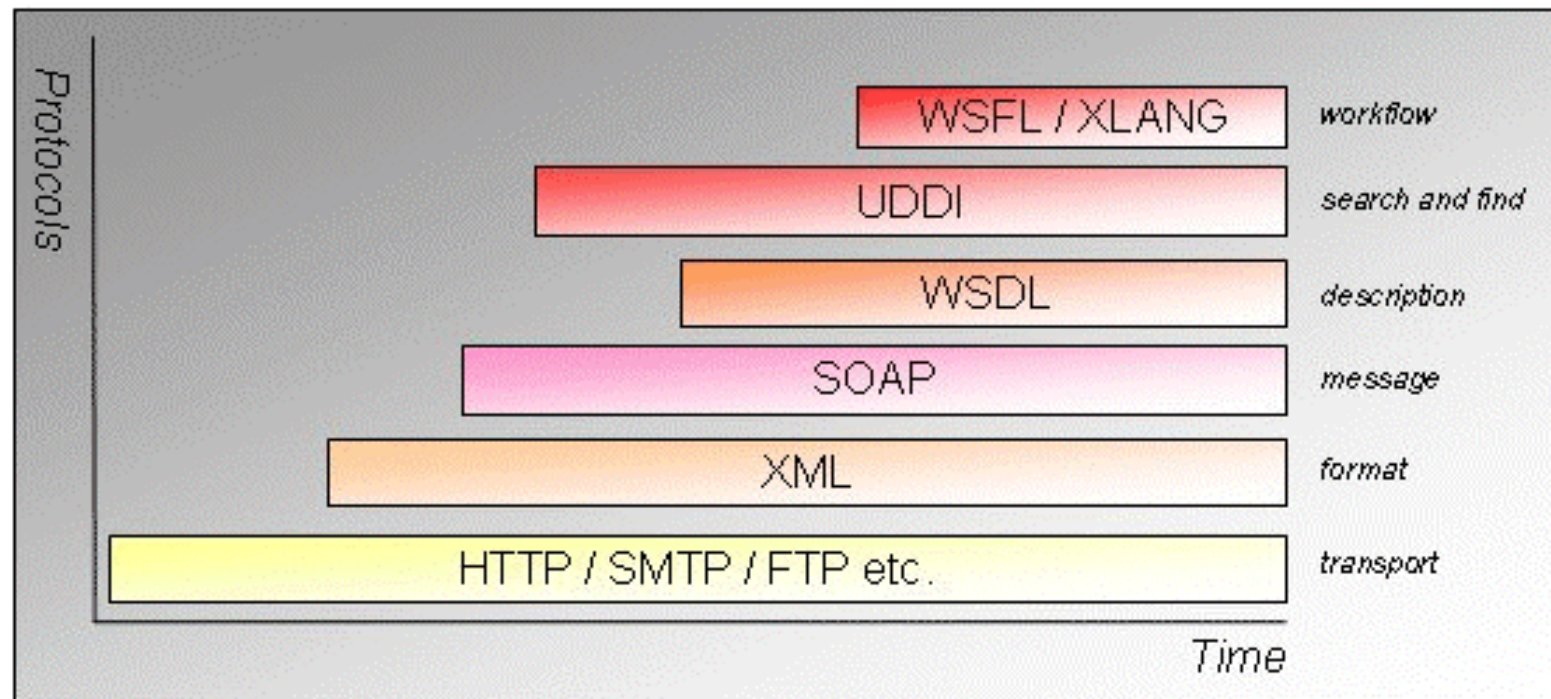


## 5.2 XLANG

```
<definitions name="StockQuoteProvider" ...  
  xmlns:xlang="http://schemas.microsoft.com/biztalk/xlang/"  
  xmlns="http://schemas.xmlsoap.org/wsdl/" ... >  
...  
<service name="StockQuoteProviderService">  
  <port name="pGetRequest" binding="tns:RequestReceivePortBinding">  
    <soap:address location="mailto:quote@example1.com"/>  
  </port>  
  <port name="pSendResponse" binding="tns:ResponseSendPortBinding">...  
  </port>  
  
  <xlang:behavior>  
    <xlang:body>  
      <xlang:sequence>  
        <xlang:action operation="AskLastTradePrice"  
          port="pGetRequest" activation="true"/>  
        <xlang:action operation="SendLastTradePrice"  
          port="pSendResponse"/>  
      </xlang:sequence>  
    </xlang:body>  
  </xlang:behavior>  
  
</service>  
</definitions>
```

## 5.3 BPEL4WS

- Business Process Execution Language for Web Services
- BPEL4WS = WSFL + XLANG
- Schön wäre es ... Vorstellung von Microsoft und IBM:



## 5.4 Bewertung

### Standardisierung von Datenaustauschprotokollen:

- schaut schlecht aus
- WSFL = „Web Services Football League“ (nach IBM)
- August 2001: erster Vorschlag von IBM (WSFL)
- März 2003: Microsoft steigt aus der Choreografie Gruppe des W3C Consortiums aus
- bis heute – keine Vereinbarung über einheitlichen Standard
- vielleicht Morgen?

# 6. Zusammenfassung

- **Workflow** = Satz von Aufgaben, die zusammen kooperieren, um einen (Geschäfts)Prozess umzusetzen
- **Modellierung** (Spezifikation) von Workflows:  
überwiegend Statecharts und Aktivitätscharts verwendet  
(UML Industrie Standard von OMG)
- **Workflow Management System** = Software System, das Erzeugung und automatische Durchführung von Workflows unterstützt und koordiniert
- **Hauptkomponenten** von WFMS:  
Workflow Engine, Prozess Definitionstool, Administration & Monitoring Tools, externe Workflow Engines, Client Applikationen, Aufgerufene Applikationen

## 6. Zusammenfassung (cont.)

- ein **Standard** von WFMC Consortium definiert die Komponenten und Schnittstellen (API1-5)
- Die **Architektur** von WFMS von zentralisiert bis verteilt
- Die **Implementierung** von WFMS:  
CORBA (Standard von OMG), Java-basiert oder XML-basiert
- **Web Services**: bisher kein Standard von W3C gewählt
- wichtigsten Pre-Standards: **WSFL** und **XLANG**  
(zusammen **BPEL4WS**)

**Vielen Dank für Ihre Aufmerksamkeit!**

