

# Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: [ittner@hs-mittweida.de](mailto:ittner@hs-mittweida.de)

WWW: [www.andreas-ittner.de](http://www.andreas-ittner.de)

Tel.: +49(0)3727-58-1288

- Motivation
- Prozesse und Prozess-Management
  - Geschäftsprozesse, Workflow-Prozesse
  - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
  - Zweck, Modellierungselemente und –sprachen
  - Petri-Netze, EPKs, BPMN, ...
- **Prozess-Analyse**
  - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
  - Simulation
- Workflow-Management-Systeme
  - Historie, Infrastruktur, Implementierungen, Standards

## 7. Performance-Analyse / Simulation

1. Fragestellungen bei der Prozess-Analyse,
2. Petrinetze mit Zeit,
3. Simulation,
4. Design: Verschiedene Szenarien.

### 1. Fragestellungen bei der Prozess-Analyse

#### Wdh.: Arten der Analyse:

- Validierung
  - Ist das Modell richtig bzgl. der Realität/Vorstellung?
  - z.B. Kundenbezug, Medien- und Organisationsbrüche
- Verifikation
  - Nachweis der Korrektheit des Prozesses,
  - Struktur (z.B. Vor- und Nachbedingungen für alle Aktivitäten),
  - Verhalten (z.B. Deadlocks, nie ausgeführte Aktivitäten).
- Leistungsbewertung
  - Leistungsfähigkeit des Geschäftsprozesses,
  - z.B. Durchlaufzeit, Kostenrechnung, Ressourcenauslastung, Service-Level.

### Qualitative Fragestellungen

- Kann es zu Deadlocks (toten Markierungen) kommen?
- Können alle Arten von Cases (Fällen) erfolgreich behandelt werden?
- Werden alle Cases irgendwann beendet?
- Können Tasks (Aufgaben) parallel ausgeführt werden?

### Quantitative Fragestellungen

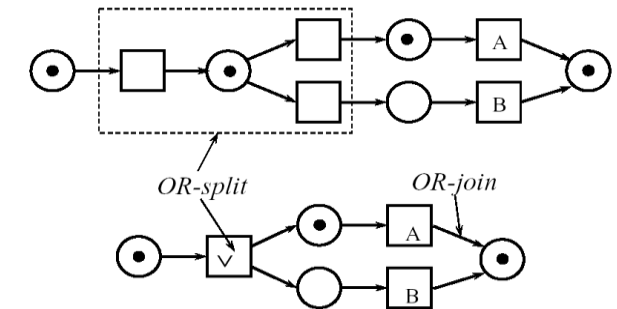
- Wie viele Cases pro Stunde können bearbeitet werden?
- Was ist die durchschnittliche Durchlaufzeit?
- Wie viele Ressourcen werden pro Case benötigt?
- Was ist die durchschnittliche Wartezeit?

### 2. Petrinetze mit Zeit

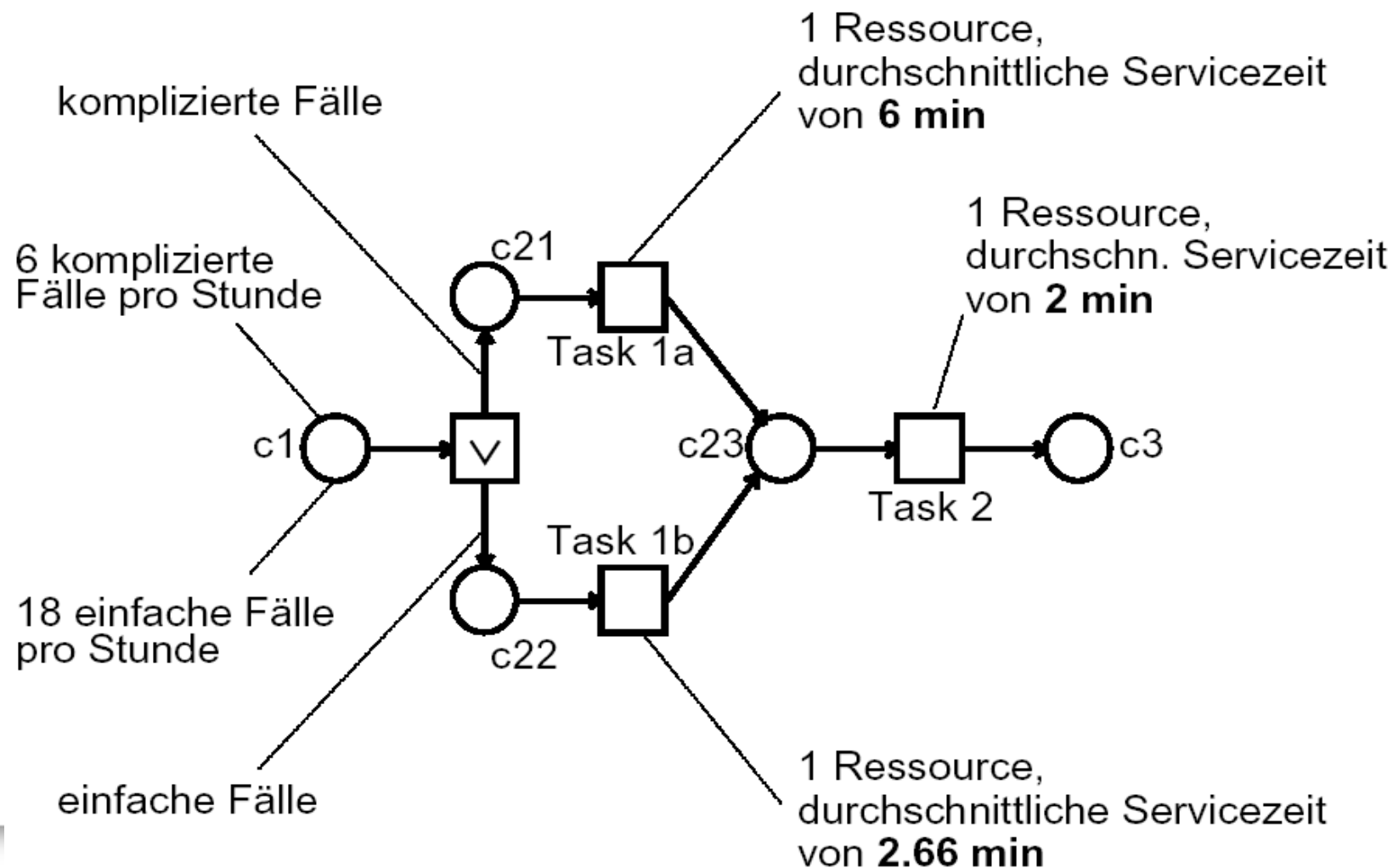
Die Erweiterung mit Zeit:

- für Performance-Analysen werden Zeitdauern, Verzögerungen etc. benötigt,
- jeder Transition  $t$  wird eine Zeitdauer  $\delta(t)$  zugeordnet
  - d.h. zuerst wird  $t$  initialisiert und die Marke verschwindet von der Inputstelle  $s$ ,
  - dann ist  $t$  aktiv während der Zeitdauer  $\delta(t)$ ,
  - Beendigung der Transition  $t$ , die Marke erscheint in der Outputstelle.
- es können für  $\delta(t)$  Zufallsverteilungen angenommen werden, da z.B. die Ankunft der Kunden und die Dauer des Service i.a. nicht bekannt sind.

## 7. Performance-Analyse / Simulation

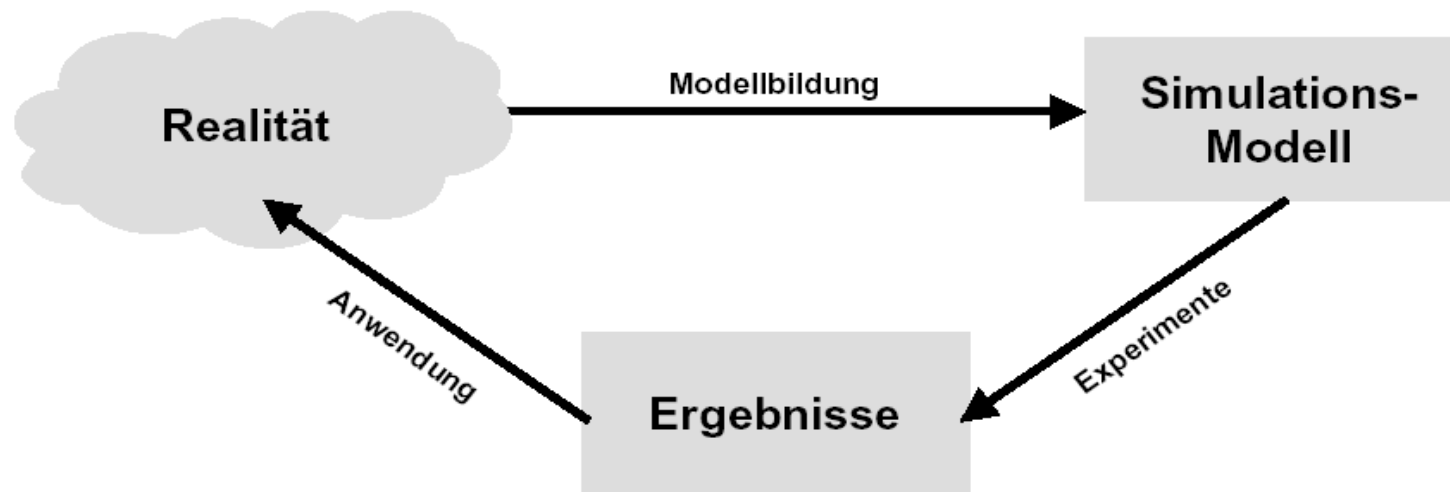


## 2. Petrinetze mit Zeit (Beispiel)



### 3. Simulation

- ist das systematische Experimentieren mit Modellen realer Problemstellungen.



- Prozesse realer Systeme können anhand von Modellen durchgeführt und ausgewertet (simuliert) werden
- Ergebnisse der Simulationsexperimente werden dann auf das reale System übertragen.



### 3. Simulation

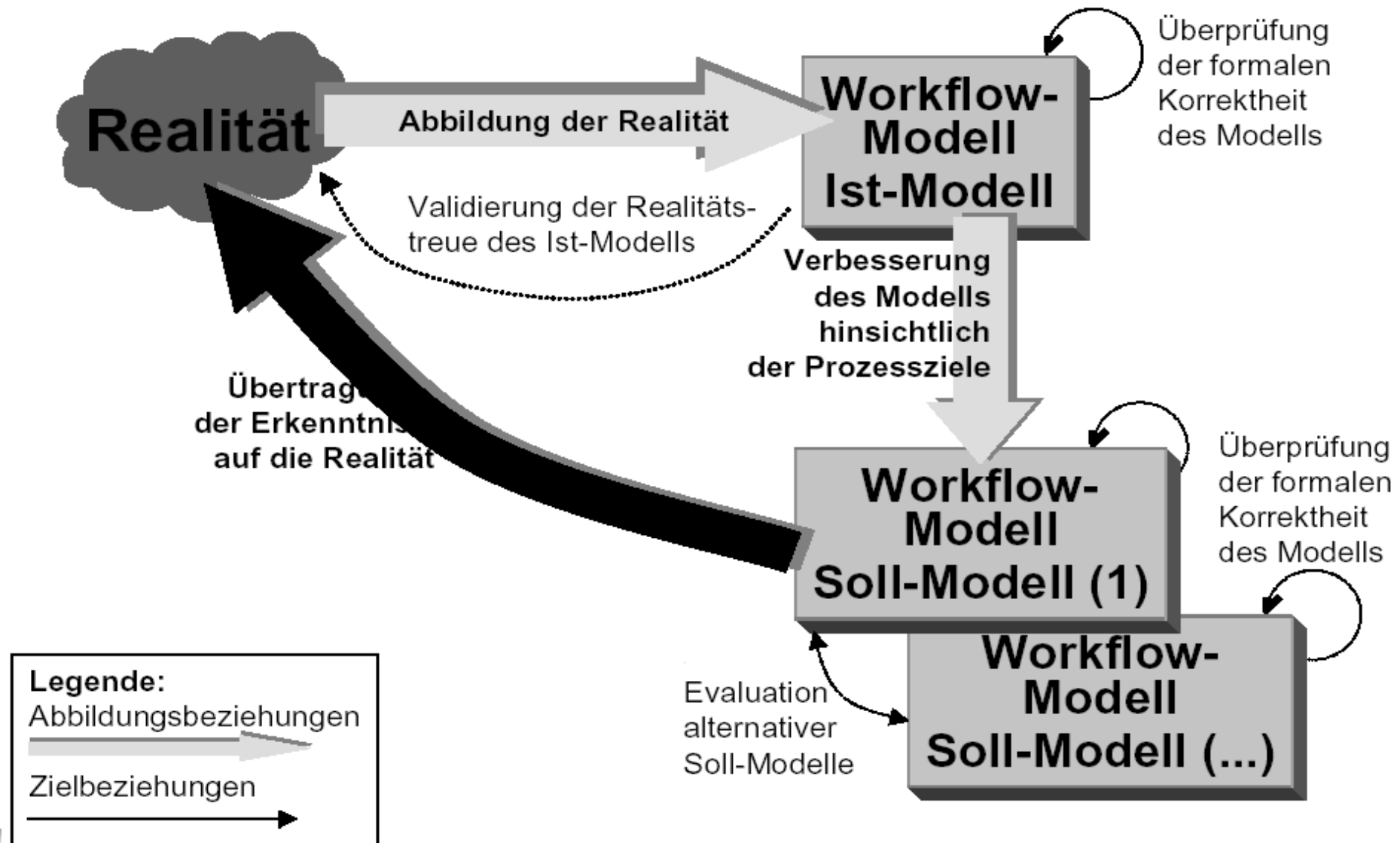
- Simulation = Probebetrieb,
  - in der Regel Computerexperiment unter Verwendung von Pseudozufallszahlen,
  - kann notwendiger Ersatz für Realexperimente sein (unwirtschaftlich, unmöglich),
  - Hilfsmittel, um die Komplexität der Realität zu vereinfachen,
  - erlaubt Blick auf wesentliche Probleme,
  - ist Instrument zur qualitativen Verbesserung und Absicherung der Unternehmensplanung,
  - ist Instrument zur Entscheidungsunterstützung.
- Kreativität zur Lösungsfindung bleibt aber beim Anwender !

## 7. Performance-Analyse / Simulation

### 3. Simulation / Vorteile der Simulation

- Visualisierung von Prozessen,
- Verfolgung von Kosten und Wertschöpfung,
- Qualitätskontrolle der Modelle,
- Risikoreduzierung,
- Erhöhung der Planungsqualität,
- Zeit- und Kostenersparnis,
- Vergleich mehrerer Alternativen,
- Vermeidung von Betriebsunterbrechungen.

### 3. Simulation / Ziele der Workflow-Simulation



### 3. Simulation / Ziele der Workflow-Simulation

- Überprüfung der Ablauffähigkeit von Workflow-Modellen
  - Überprüfung auf formale Korrektheit und Konsistenz,
  - noch keine Aussagen über den Inhalt der Modelle,
- Validierung der Realitätstreue von Workflow-Modellen
  - fachlich-inhaltliche Korrektheit,
  - inwieweit bildet das Workflow-Modell die Realität angemessen ab,
  - Validierung des Inhaltes des Modells (Gegenüberstellung von relevanten Ist-Daten der Realität und Simulationsergebnissen),
- Evaluation alternativer Workflow-Modelle
  - entscheidungsunterstützende Informationen für die qualitative Verbesserung der Workflow-Modelle und somit auch der Abläufe,
  - Vergleich von Simulationsergebnissen und Zielkennzahlen (wie mittlere Durchlaufzeiten, Kapazitätsauslastungen, Prozesskosten)
  - „Was wäre wenn“ Analysen.

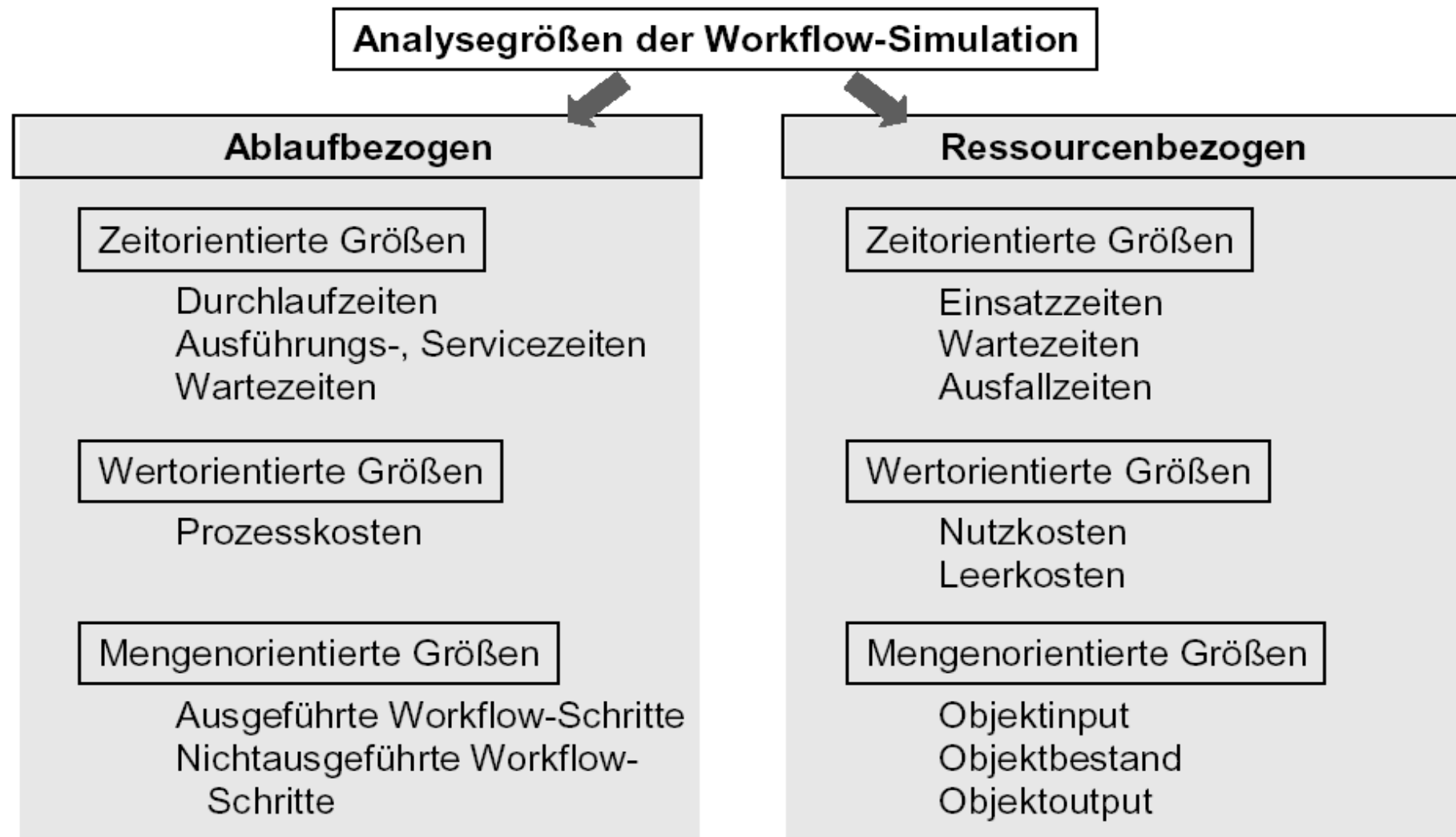
### 3. Simulation / Analysegrößen der Workflow-Simulation

- Unterteilung in
  - ablaufbezogene und
  - ressourcenbezogene Analysegrößen

(Betrachtung der im Rahmen der Simulation erzeugten Workflowinstanzen hinsichtlich des Ablaufverhaltens bzw. der Ressourcen (Bearbeiter, Programme,...)).
- weitere Differenzierung in
  - zeit-,
  - wert- und
  - mengenorientierte Größen.

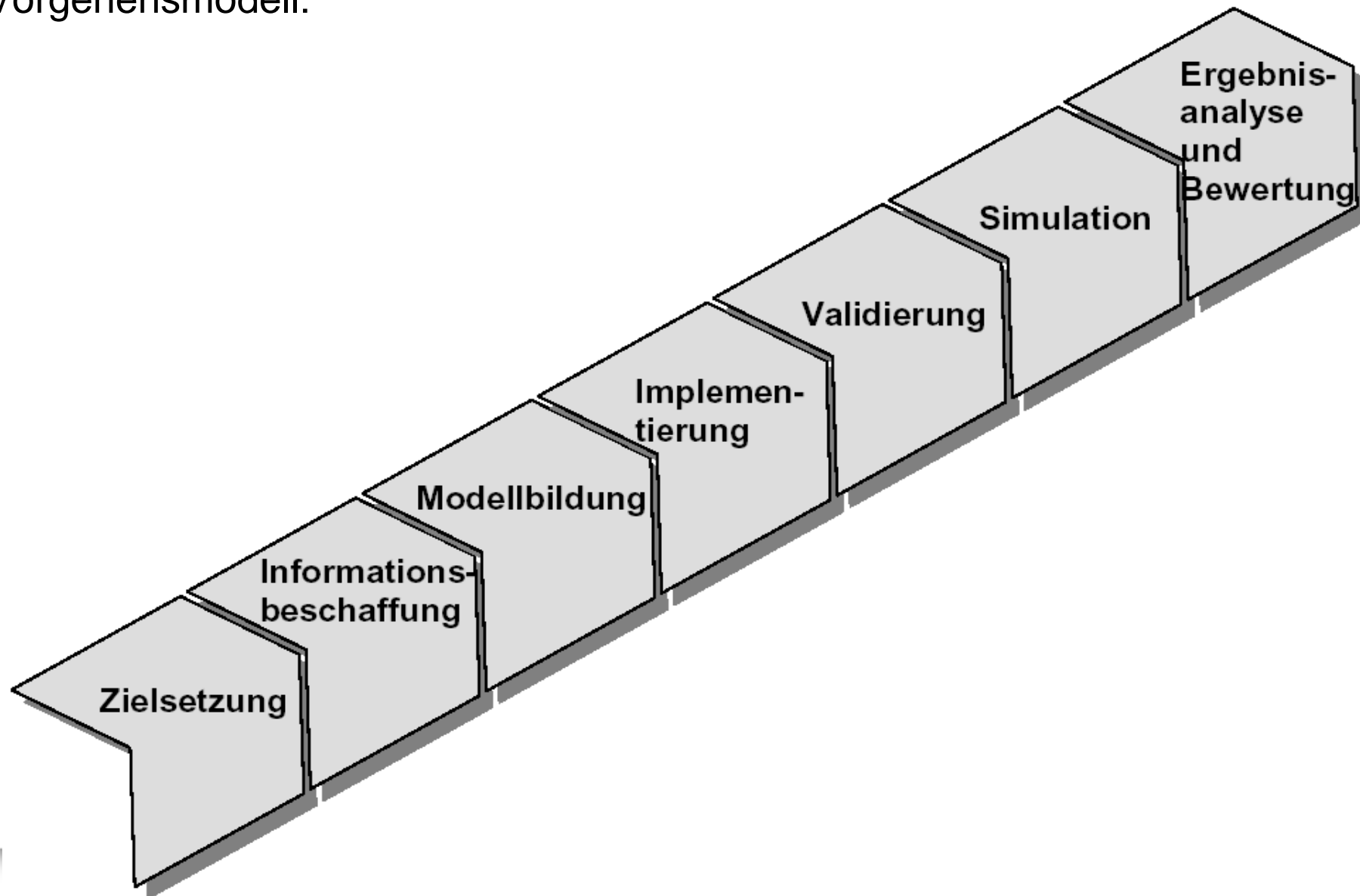
### 3. Simulation / Analysegrößen der Workflow-Simulation

nach Gadatsch



### 3. Simulation / Durchführung der Simulation - 7 Schritte

Vorgehensmodell:



### 3. Simulation / Durchführung der Simulation - 7 Schritte

#### 1. Zielsetzung festlegen

- vor Beginn müssen die Ziele quantitativ, also rechnerisch nachvollziehbar, schriftlich festgelegt/fixiert werden,  
Beispiel: Reduzierung der Durchlaufzeiten um ...%,

#### 2. Informationsbeschaffung

- Erfassung aller relevanten Basisdaten für die Simulationsuntersuchung einschließlich einer Plausibilitäts- und Vollständigkeitsprüfung,  
Beispiel: Bearbeitungszeiten, Kapazitäten, Störgrößen,
- Daten müssen klassifiziert und verdichtet werden, so dass sie den Aufgaben zugeordnet werden können.

#### 3. Modellbildung

- Festlegung „was mit den Daten geschehen soll“,  
Beispiel: Bearbeitungsreihenfolgen, Prioritätsregeln



### 3. Simulation / Durchführung der Simulation - 7 Schritte

#### 4. Implementierung des Modells

- konkrete Erfassung des Modells im Simulator,
- Detaillierungsgrad abhängig von der fixierten Zielsetzung,
- Modellbildung und Implementierung können evtl. gleichzeitig erfolgen, da viele Werkzeuge eine rechnergestützte Modellierung anbieten,

#### 5. Überprüfung des Modells (Validierung)

- Überprüfung der Übereinstimmung mit der abzubildenden Realität,  
Beispiel: Vergleich von Vorab-Simulationsläufen mit bekannten Ergebnissen (evtl. Daten des Vorjahres),

#### 6. Experimentieren mit dem Modell (Simulation)

- Simulationsphase: verschiedene Versuchsreihen mit variierenden Parametern,
- Dokumentation nicht vergessen, die jeden Parameterwechsel protokolliert!
- Beispiel: unterschiedliche Planungszeiträume (Woche, Jahr,...), unterschiedliche Kapazitäten.

### 3. Simulation / Durchführung der Simulation - 7 Schritte

#### 7. Ergebnisanalyse und Bewertung

- numerische Ergebnisse können graphisch dargestellt und analysiert werden (Diagramme, etc.),
- Auswertung kann zu Modelländerungen führen
- → neue Simulationsdurchführung (Beginn bei Schritt 1 oder später wenn möglich).

### 3. Simulation / Durchführung der Simulation – weitere Tipps:

- Überschaubarkeit/Übersichtlichkeit bewahren  
(keine zu große Anzahl von Experimenten)
- pro Simulationslauf nicht mehrere Parameter gleichzeitig ändern !!!  
(sonst sind Ursache und Auswirkung nicht direkt erkennbar)

### 3. Simulation / Warteschlangenmodell:

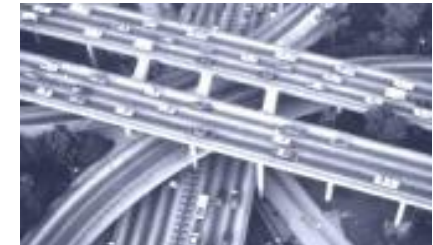
- ist dafür geeignet, eine Leistungsanalyse von Systemen oder aber auch von Geschäftsprozessen durchzuführen,
- Warteschlangen sind charakterisiert durch Personen, die
  - unregelmäßig ankommen,
  - und auf ihre Abfertigung (Service) oder ihren Durchlass warten.
- es gibt einen oder mehrere Server (also Bediener), die den Service anbieten,
- die Ankunftszeitpunkte der Personen sowie die Dauer des Services sind i. Allg. nicht bekannt (Annahme von Zufallsverteilungen),
- Modelle können auch statt auf Personen auf allgemein auch auf zu bearbeitende Fälle, Ausfälle bei Maschinen/Ressourcen, etc. angewendet werden.

## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:

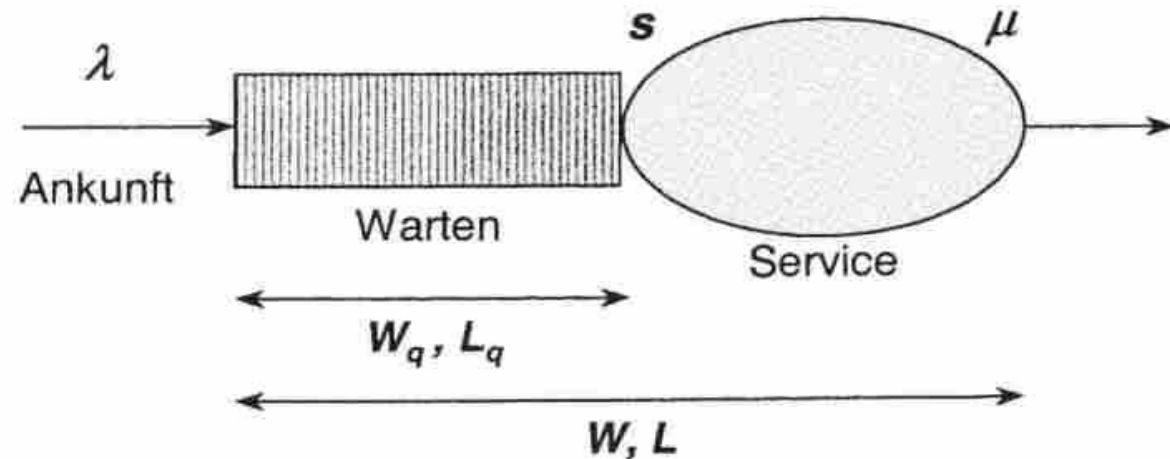
Beispiele:

Kunde	Bedienstation	Warteschlange
Zwischenprodukte	Maschine	Puffer vor der Maschine
Skifahrer	Ski-Lift	Ski-Fahrer vor dem Ski-Lift
Flugzeuge	Startbahn	Flugzeuge auf dem Rollfeld
Flugzeuge	Landebahn	Flugzeuge in der Warteschleife
Touristen	Airline-Schalter	Touristen vor dem Schalter
Reisende	Fahrkartenschalter/-automat	Reisende vor dem Schalter
Kunden	Supermarkt-Kasse	Kundenschlange
Fahrzeuge	Tankstelle oder Ampel	Fahrzeugschlange
Patienten	Arzt	Wartezimmer
Bestellungen	Verkaufsabteilung	Lieferverpflichtungen
Druckaufträge	Drucker	Druck-Jobs
Studenten	Mensa	Studenten vor der Essensausgabe
Maschinen	Instandhaltung	Reparaturaufträge
Transportgüter	Transportsystem	Zwischenlager
Lastwagen	Laderampe	Kolonne
Anrufe	Vermittlungsstelle	Anrufwiederholer
Prozesse	CPU	Prozess-Warteschlange



## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:



- $\mu$  – Anzahl Fälle, die pro Zeiteinheit abgearbeitet werden können (Bedienrate),
- $\lambda$  – durchschnittl. eintreffende Fälle pro Zeiteinheit (Ankunftsrate),
- $s$  – Anzahl der Server, die den Service durchführen,
- $W_q$  – durchschnittl. Wartezeit in der Warteschlange,
- $L_q$  – durchschnittl. Anzahl von Fällen in der Warteschlange,
- $L$  – durchschnittl. Anzahl von Fällen im System,
- $W$  – durchschnittl. Verweildauer im System.

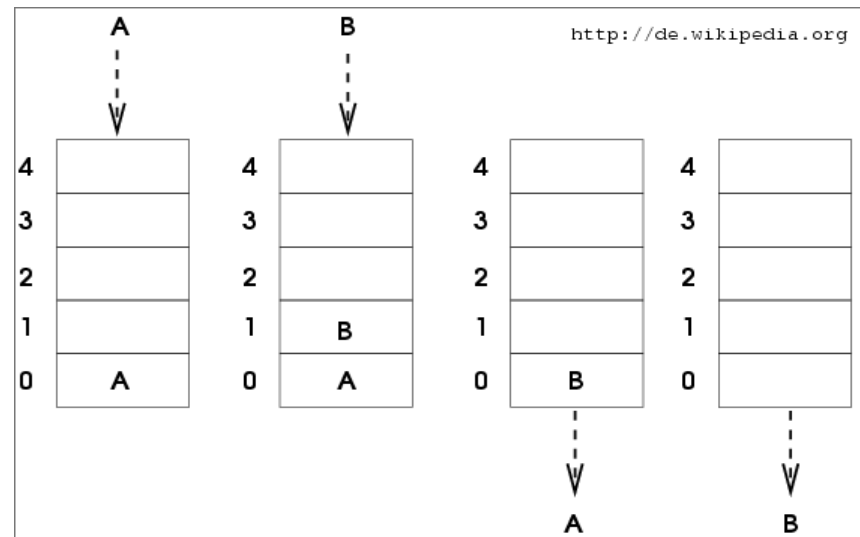
### 3. Simulation / Warteschlangenmodell:

- **Einfachstes** Warteschlangenmodell mit nur einem Server, dem FIFO/FCFS-Prinzip, exponentialverteilten Zwischenankunfts- und Bedienzeiten (Poissonverteilung der Ankunfts- und Bedienrate):
- $\mu$  – Mittlere Bedienrate,
- $1/\mu$  – Mittlere Bedienzeit,
- $\lambda$  – Mittlere Ankunftsrate,
- $1/\lambda$  – Mittlere Zwischenankunftszeit,
- $r = \lambda / \mu$  – Mittlere Auslastung,
- $W = 1 / (\mu - \lambda)$  – Mittlere Durchlaufzeit,
- $W_q = \lambda / (\mu * (\mu - \lambda))$  – Mittlere Wartezeit,
- $L_q = \lambda * \lambda / \mu * (\mu - \lambda)$  – Mittlere Anzahl von Kunden/Fällen in der Warteschlange,
- $L = \lambda / (\mu - \lambda)$  – Mittlere Anzahl von Kunden/Fällen im System.

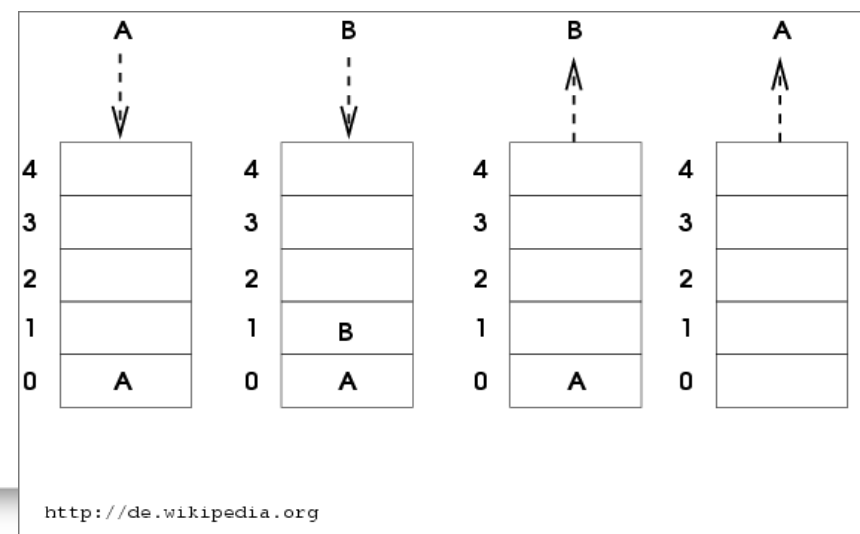
## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:

FIFO:



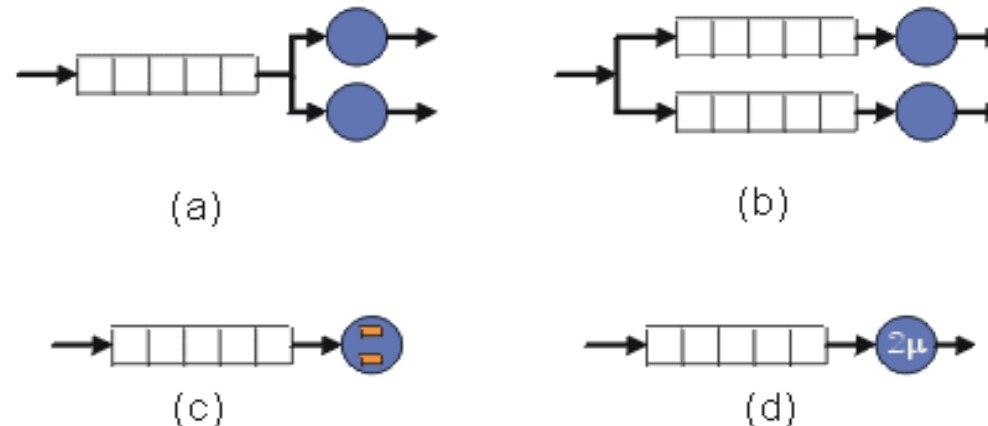
LIFO:





### 3. Simulation / Warteschlangenmodell:

Design von Warteschlangensystemen (exemplarisch 4 typische Muster):

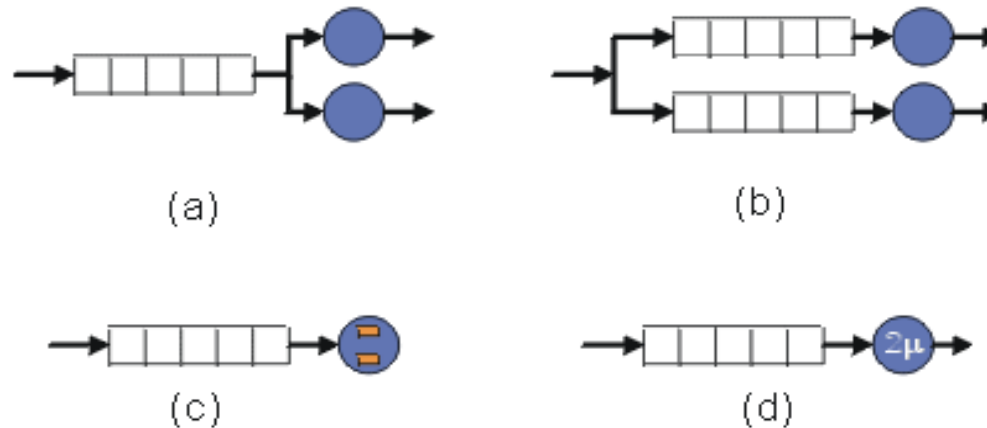


- a) Es stehen zwei Bediener zur Verfügung. Die ankommenden Kunden werden von einem so genannten Dispatcher dem jeweils freiwerdenden Schalter zugewiesen. Nach diesem Prinzip werden z. B. beim Check-In auf dem Flughafen die Warteschlangen abgearbeitet.
- b) Auch hier stehen zwei parallele Bediener zur Verfügung. Allerdings werden die Kunden bereits bei ihrer Ankunft gleichmäßig (d. h. zu je 50%) auf die beiden Warteschlangen verteilt. In dieser Weise stellt man sich an den Fahrkartenschaltern auf Deutschlands Bahnhöfen an.

## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:

Design von Warteschlangensystemen (exemplarisch 4 typische Muster):

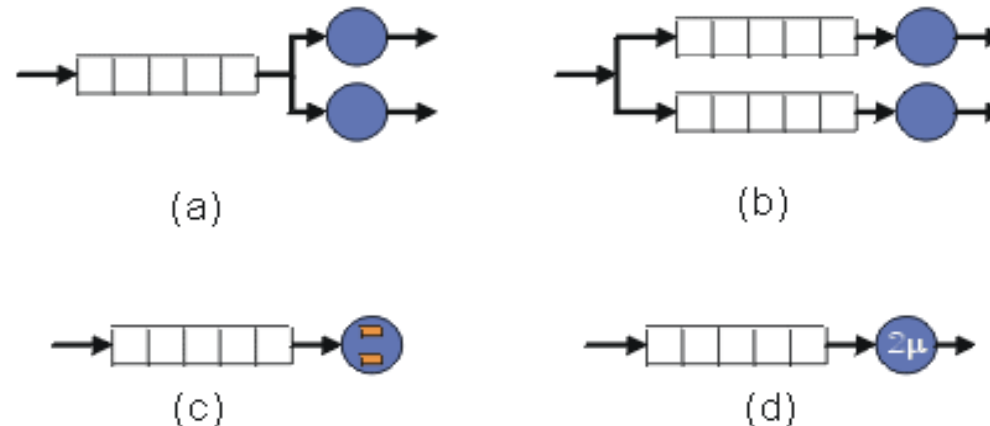


- c) Es gibt nur einen Bediener und eine Warteschlange. Der Bediener kann jedoch zwei Kunden gleichzeitig bedienen.
- d) Es gibt nur einen Bediener und eine Warteschlange. Der Bediener arbeitet jedoch mit doppelter Geschwindigkeit.

## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:

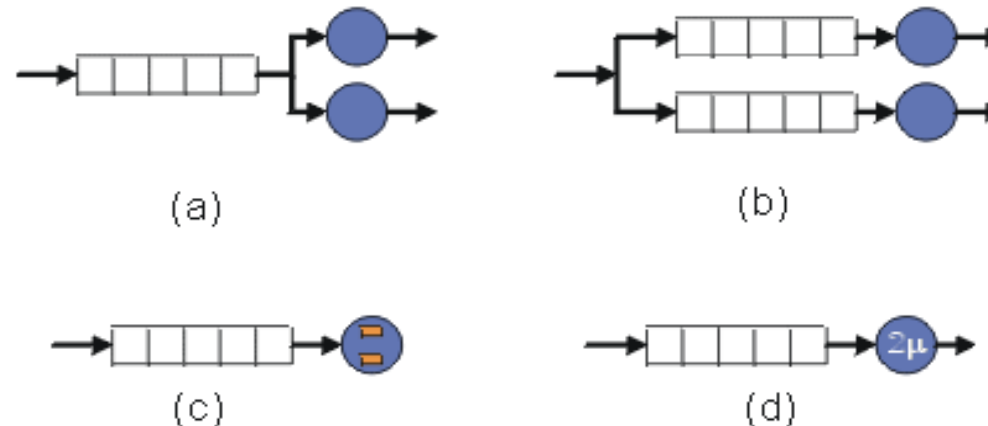
Design von Warteschlangensystemen (exemplarisch 4 typische Muster):



- Aufgabe: Es stellt sich die Frage, welches dieser Systeme in der Praxis am günstigsten ist?
- Zum Vergleich der Systeme verwenden wir als Leistungsmerkmal die mittlere Anzahl Kunden im System ( $L$ ).
- Für den Vergleich der Leistungsgrößen werden folgende Parameter angenommen:  
Ankunftsrate = 1.8, Bedienrate = 1.0, Variationskoeffizient des Ankunftsstroms  $c_l = 1.0$ ,  
Variationskoeffizient des Bedienprozesses  $c_s = 0$ .

### 3. Simulation / Warteschlangenmodell:

Design von Warteschlangensystemen (exemplarisch 4 typische Muster):



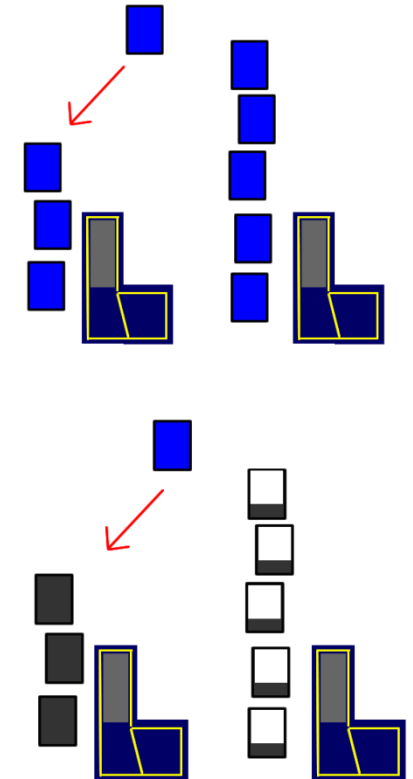
- Lösung: (a)  $L = 5,64$ , (b)  $L = 9,90$ , (c)  $L = 6,35$ , (d)  $L = 4,95$
- Das Beispiel zeigt, dass eine gemeinsame Warteschlange günstiger ist als zwei getrennte und dass eine doppelt so schnelle Maschine besser geeignet ist als zwei parallele, halb so schnelle.
- Andererseits leisten zwei parallele Maschinen jedoch mehr als eine einzige Maschine, die zwei Teile gleichzeitig bearbeiten kann.

## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:

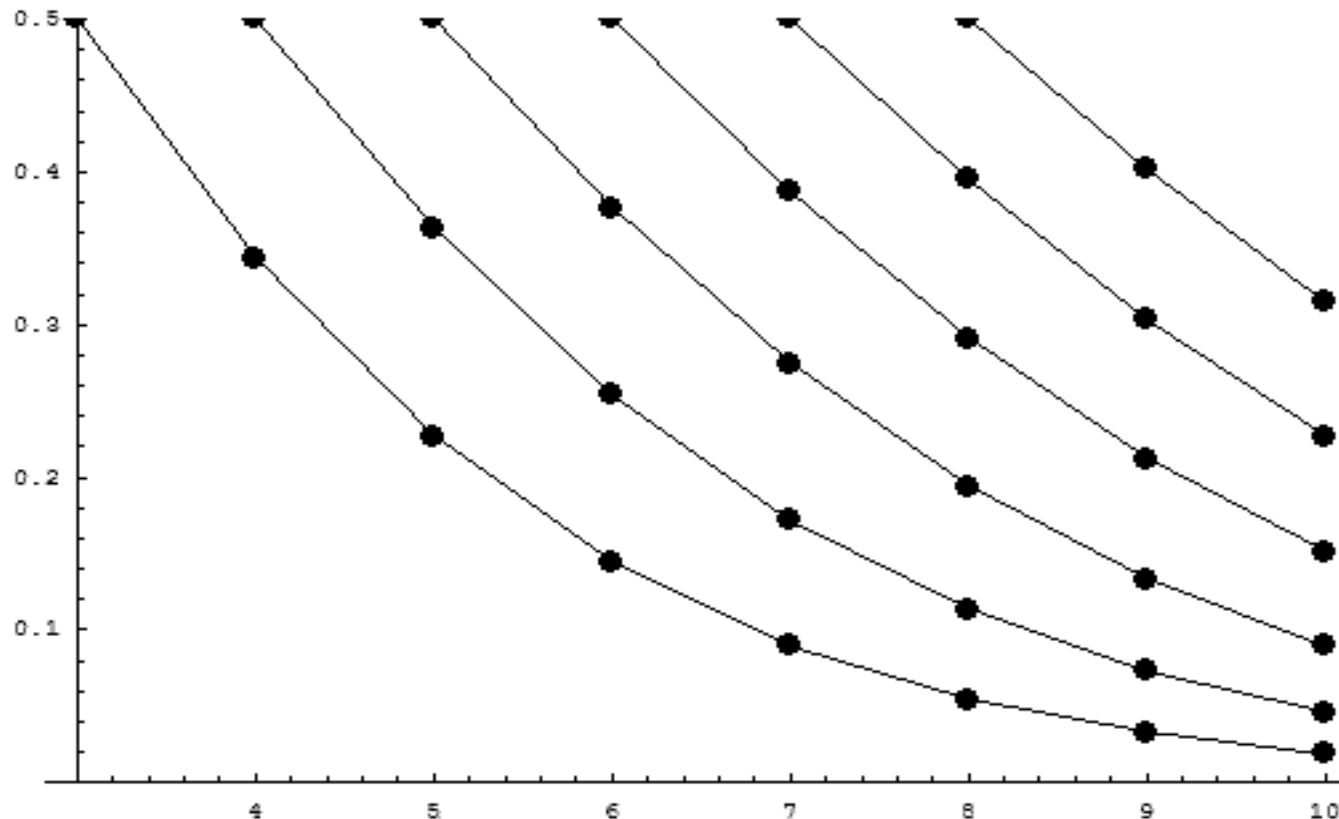
Warum steht man so oft in der falschen Warteschlange ?

- Stehen wie im Supermarkt mehrere Kassen zur Auswahl, so neigt man dazu, sich bei der Kasse mit der kürzesten Warteschlange anzustellen, um die eigene Wartezeit zu minimieren.
- **Problem:** offensichtlich garantiert diese Strategie nicht, dass man auch schneller abgefertigt wird. Da die Bediendauern der einzelnen Kunden zufällig schwanken (der eine hat mehr im Einkaufskorb, der andere weniger), kann es vorkommen, dass in der langen Warteschlange zufällig viele kleine Aufträge aufeinander folgen, während in der kürzeren Schlange große Aufträge vorherrschen. In diesem Fall muss man in der kürzeren Schlange möglicherweise länger warten als in der längeren Schlange.



## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:



- x-Achse ist die Anzahl der Kunden in der jeweils längeren Warteschlange und auf der y-Achse die Wahrscheinlichkeit, länger warten zu müssen, aufgetragen. Die unterste Kurve steht für "3 Kunden in der eigenen Warteschlange" und die oberste für "8" Kunden in der eigenen Warteschlange".
- Der am weitesten links gelegene Punkt jeder Linie liegt bei einer Wahrscheinlichkeit von 50%, was auch plausibel ist: Denn wenn sich 3 Kunden in der eigenen Warteschlange und 3 Kunden in der anderen Warteschlange befinden, so beträgt die Wahrscheinlichkeit, dass man länger warten muss, gerade 50%.

## 7. Performance-Analyse / Simulation

### 3. Simulation / Warteschlangenmodell:

Java Applet unter

<http://www.stochastik.tu-clausthal.de/Presse/Schulen/Rechner>

**Rechner für einfache Warteschlangensysteme**

Mit diesem Rechner können Sie einfache Wartesituationen analysieren:  
(Bitte etwas Geduld, bis das Java-Applet geladen ist)

Ankunftsstrom	Bedienungsprozess	Ausfallverhalten
<input type="text" value="0,9"/> <b>Ankunftsrate (Teile/ZE)</b>	<input type="text" value="1"/> <b>Bedienrate (Teile/ZE)</b>	<b>Verfügbarkeit: 1</b>
<b>Variationskoeffizient 1</b>	<b>Variationskoeffizient 1</b>	<input type="text" value="1,0"/> <input type="text" value="0,9"/>
		<input type="text" value="0"/> <b>Mittlere Ausfallzeit (ZE)</b>
	<input type="text" value="1"/> <b>Batchgröße</b>	<b>Variationskoeffizient 1</b>
	<input type="text" value="1"/> <b>Anzahl paralleler Bediener</b>	

**Leistungsgrößen des Warteschlangenmodells**

Auslastung	Kunden im System	Verweilzeit (ZE)
<b>rho = 0,9</b>	E[N]=9	E[V]=10
	E[Q]=8,1	E[W]=9
	E[B]=0,9	E[C]=1

### 4. Design: Verschiedene Szenarien

Bei komplexen Problemen ist es schwierig, eine korrekte Leistungsbewertung durchzuführen: → Simulation

#### Gegeben:

- 2 Tasks sind pro Kunde zu bearbeiten
- 24 Kunden pro Stunde

#### Frage:

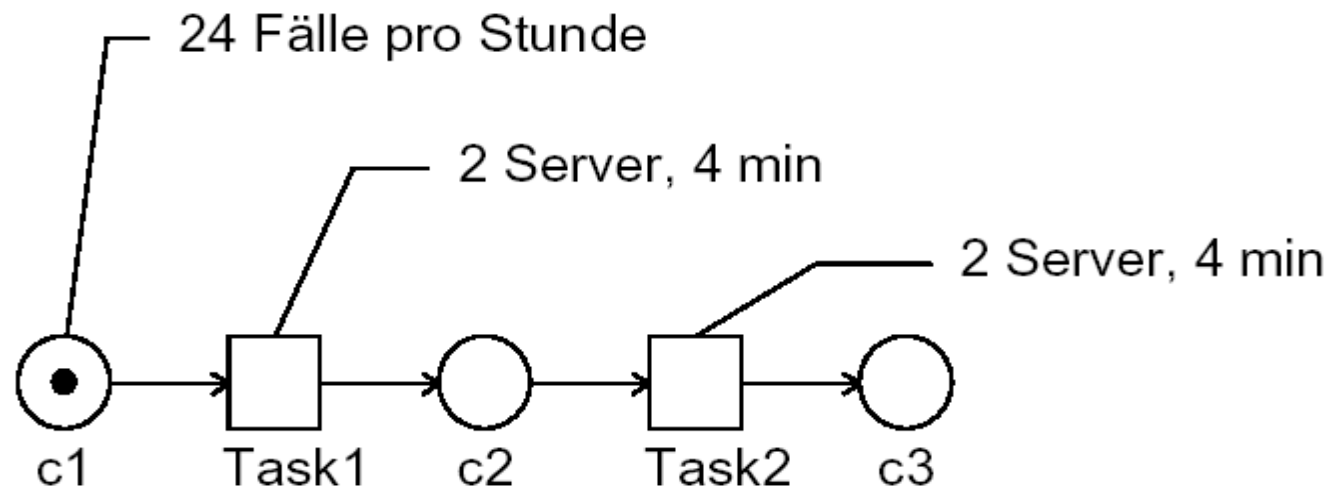
- Wie sieht das beste Design aus?

#### Zu berechnen:

- durchschnittliche Durchlaufzeit,
- durchschnittliche Servicezeit,
- durchschnittliche Wartezeit,
- (Auslastungsgrade,)
- (durchschnittliche Anzahl im System.)



### 4. Szenario 1: sequentiell



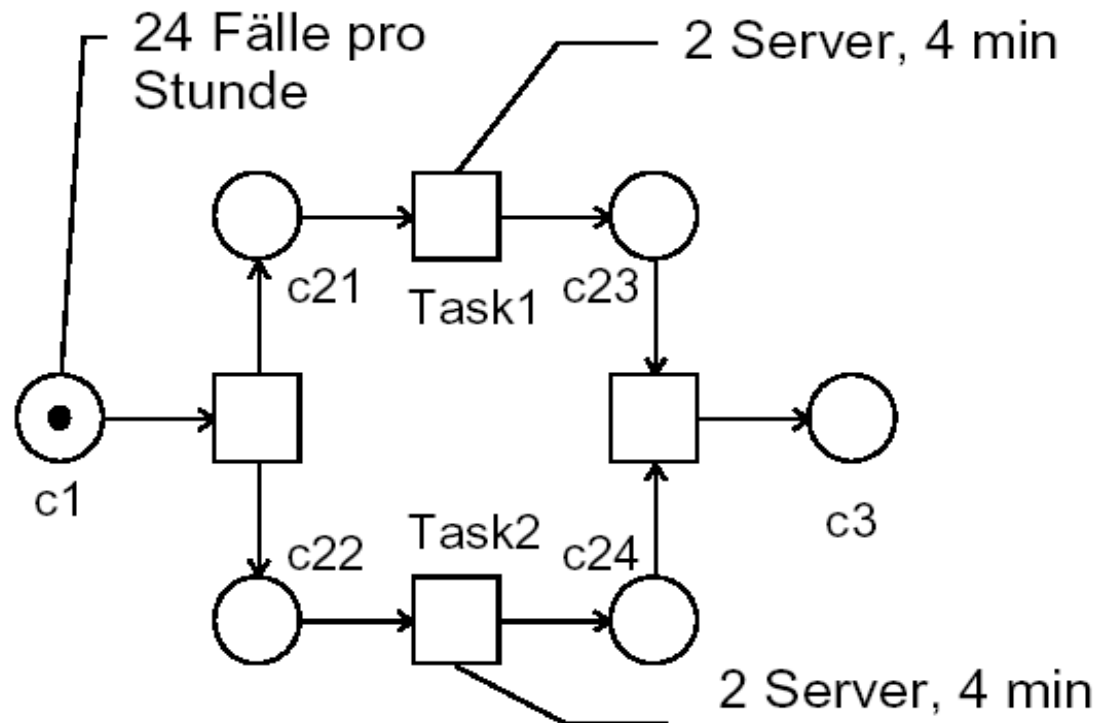
Durchlaufzeit : 22.2 min

Servicezeit: 8.0 min

Wartezeit: 14.2 min

## 7. Performance-Analyse / Simulation

### 4. Szenario 2: parallel

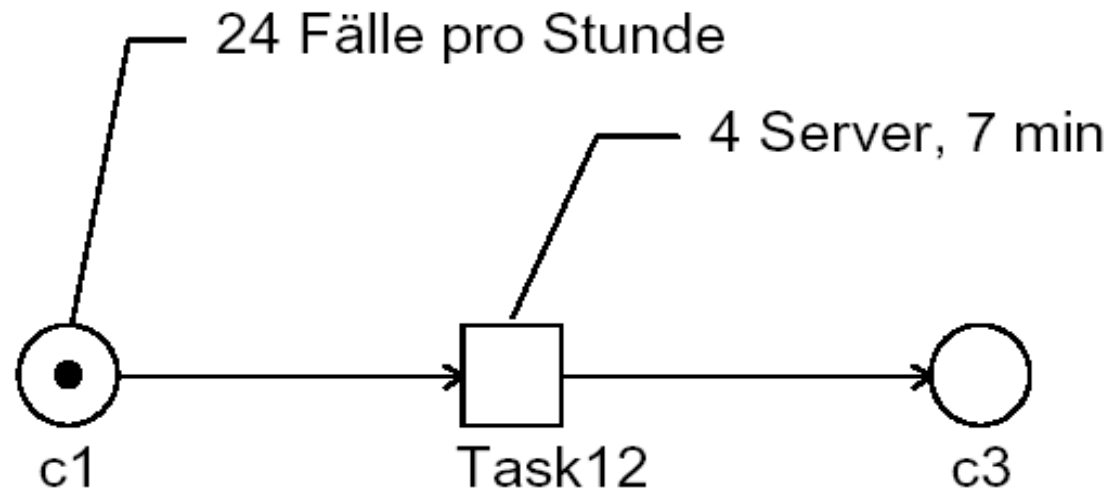


Durchlaufzeit : 15.0 min

Servicezeit: 4.0 min

Wartezeit: 11.0 min

### 4. Szenario 3: Komposition



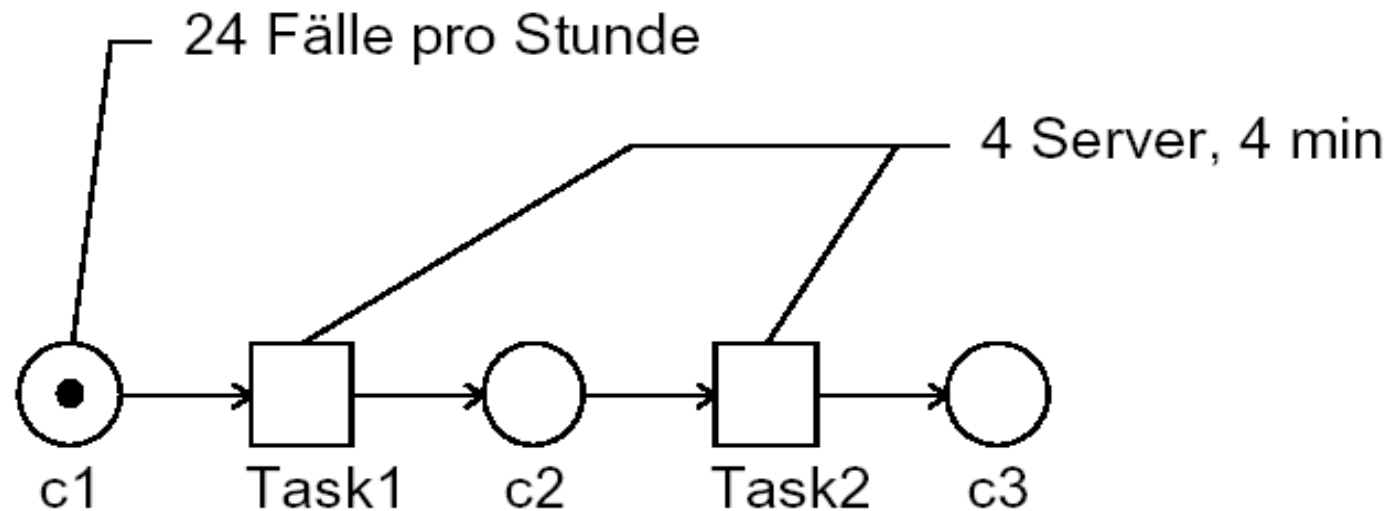
Durchlaufzeit : 9.5 min

Servicezeit: 7.0 min

Wartezeit: 2.5 min

## 7. Performance-Analyse / Simulation

### 4. Szenario 4: flexible Ressourcen

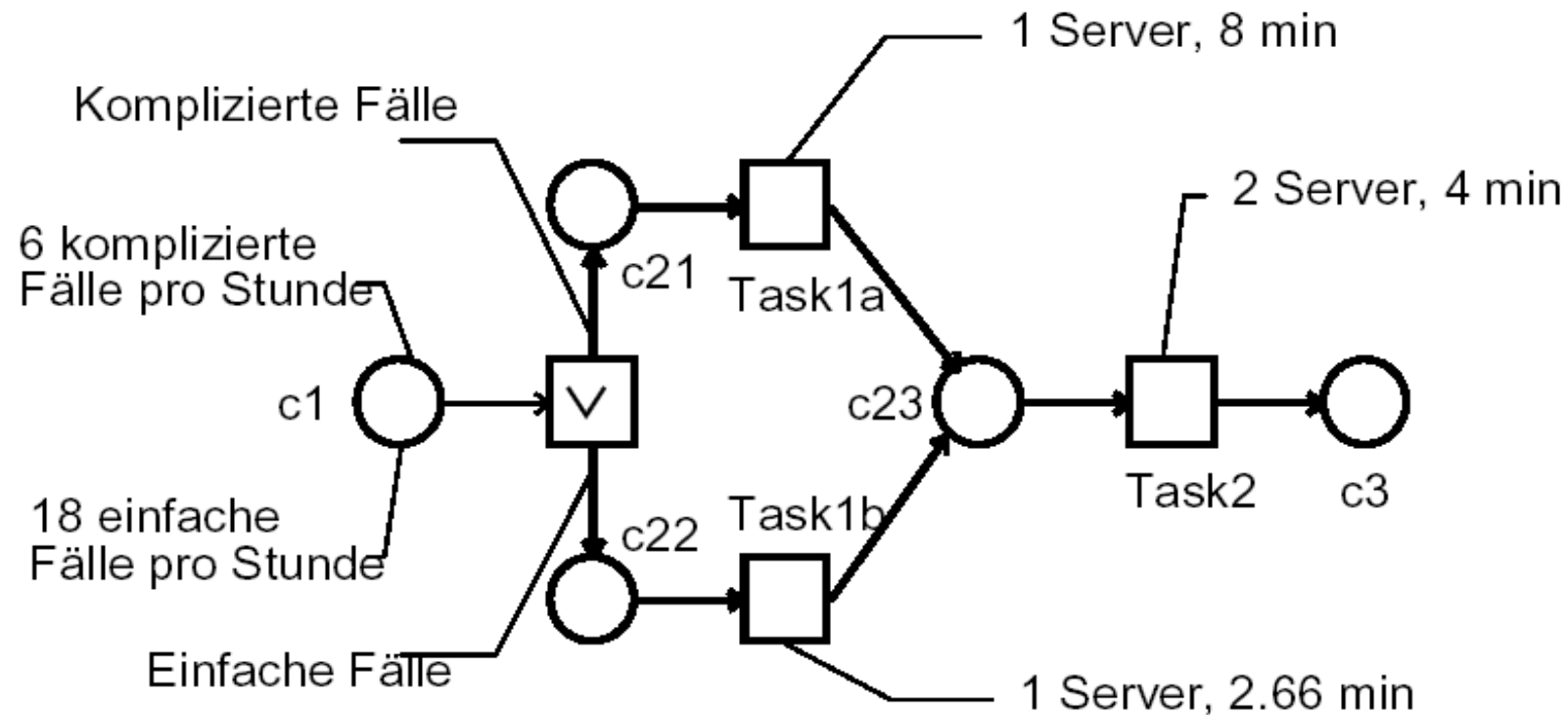


Durchlaufzeit : 14.0 min

Servicezeit: 8.0 min

Wartezeit: 6.0 min

### 4. Szenario 5: Vorauswahl

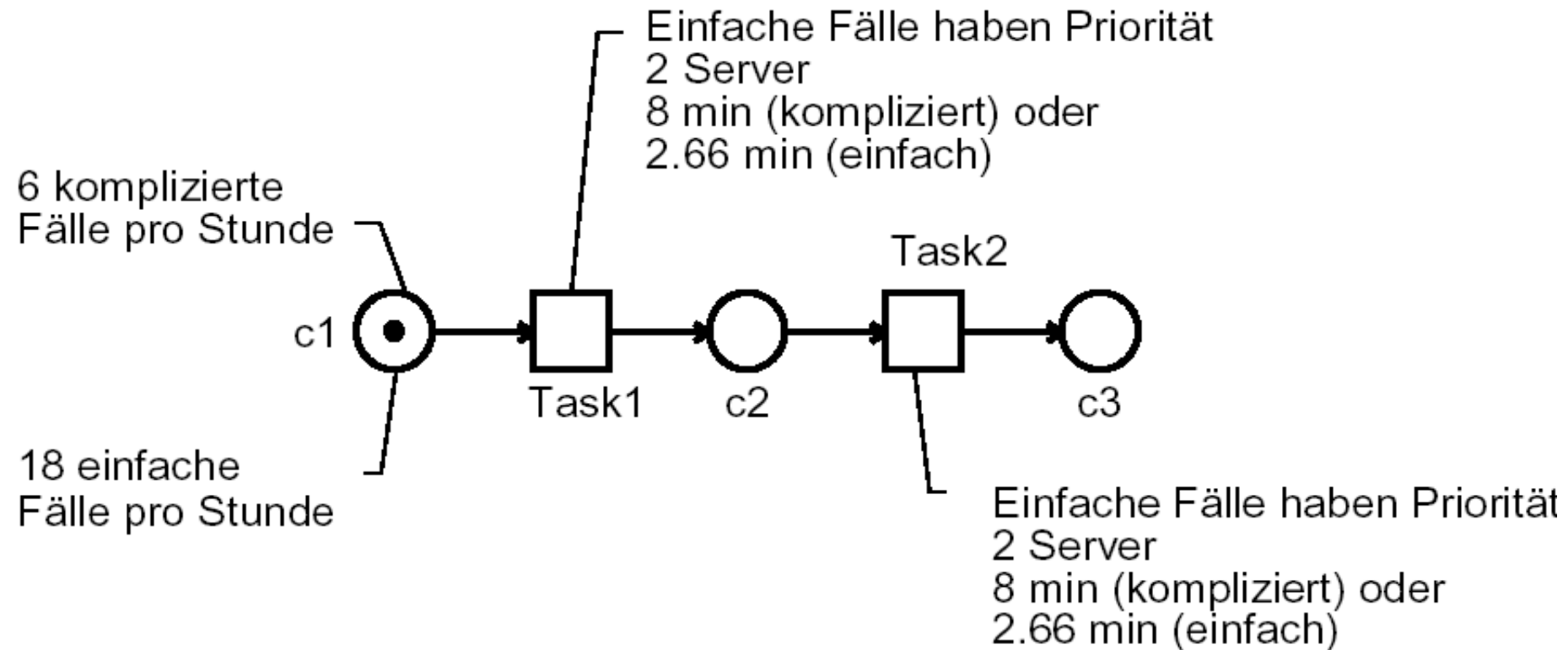


Durchlaufzeit : 31.1 min

Servicezeit: 8.0 min

Wartezeit: 23.1 min

### 4. Szenario 6: Priorität



Durchlaufzeit : 14.0 min

Servicezeit: 8.0 min

Wartezeit: 6.0 min

### 4. Übersicht der Ergebnisse

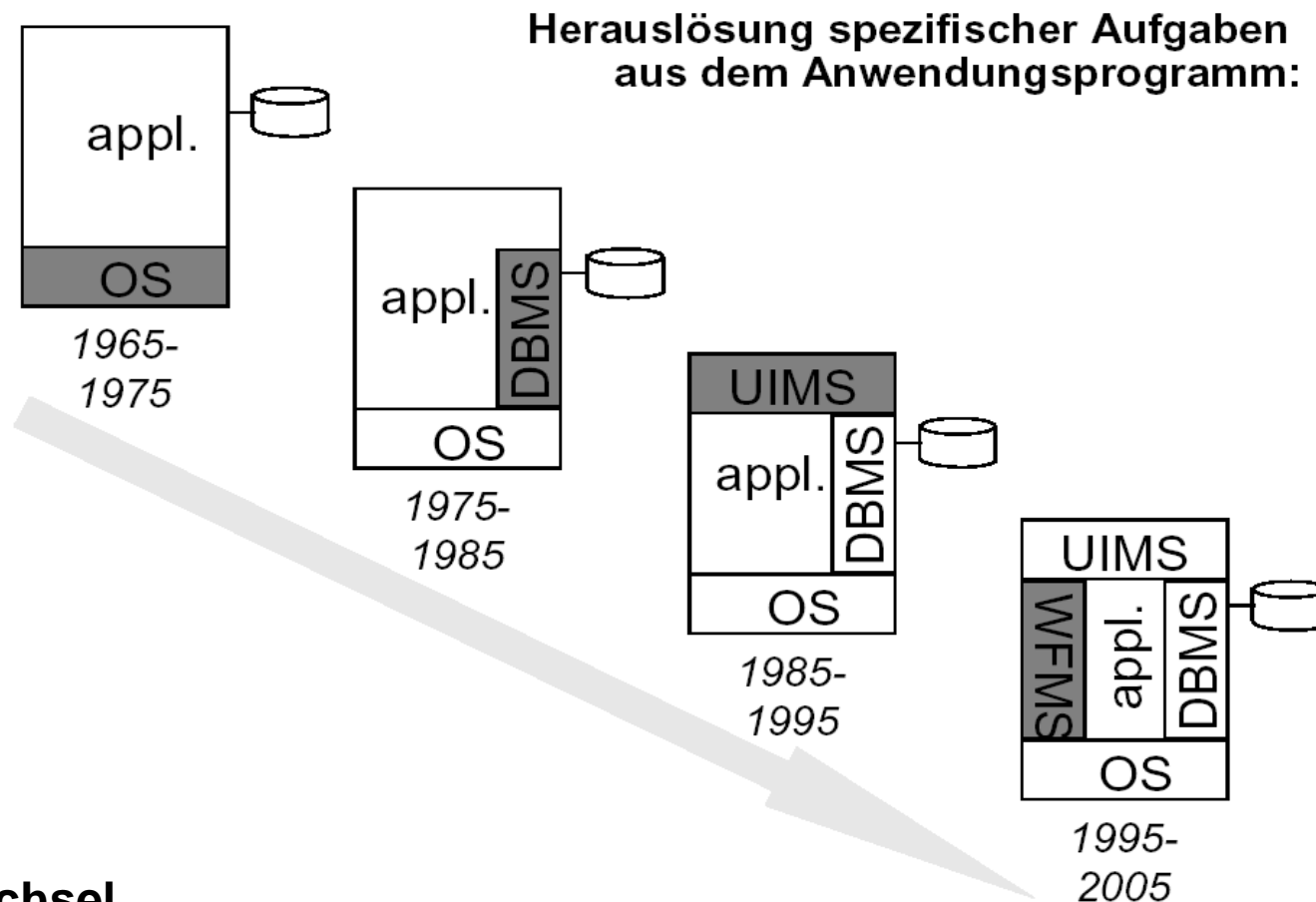
Szenario	Beschreibung	durchschn. Durchlaufzeit	durchschn. Servicezeit	durchschn. Wartezeit
Situation 1	sequentiell	22.2	8.0	14.2
Situation 2	parallel	15.0	4.0	11.0
Situation 3	Komposition	9.5	7.0	2.5
Situation 4	Flexible Ress.	14.0	8.0	6.0
Situation 5	Triage	31.1	8.0	23.1
Situation 6	Priorität	14.0	8.0	6.0

- Motivation
- Prozesse und Prozess-Management
  - Geschäftsprozesse, Workflow-Prozesse
  - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
  - Zweck, Modellierungselemente und –sprachen
  - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
  - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
  - Simulation
- Workflow-Management-Systeme
  - Historie, Infrastruktur, Implementierungen, Standards



1. Historie
2. Aufgaben eines WFMS
3. Build-Time und Run-Time eines WFMS
  1. Bestandteile der Build-Time
  2. Bestandteile der Run-Time
4. Anforderungen an ein WFMS
5. WFMS-Architekturen
6. Implementierung eines Workflow-Designs
7. Konzeptuelle Alternativen
8. WfMC-Referenzmodell

# 1. Historie

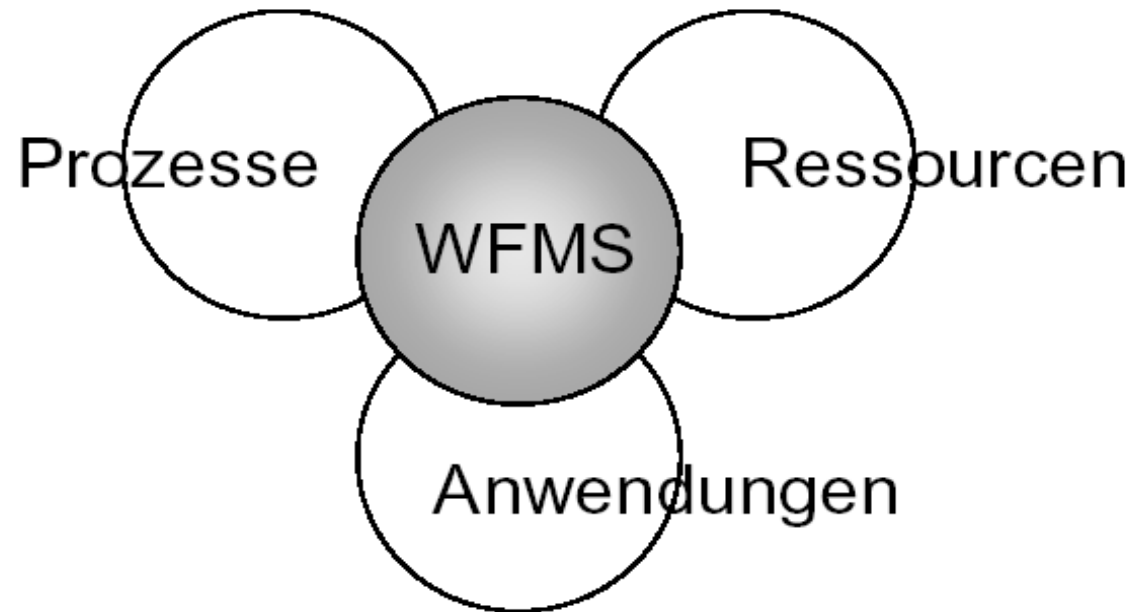


## Paradigmenwechsel

- Früher: Informationssysteme unterstützen die Ausführung einzelner Aufgaben
- Heute: Informationssysteme unterstützen Geschäftsprozesse, „they manage the flow of work“

# 1. Historie

Richtige Kombination von Prozessen, Ressourcen und Anwendungen durch das WFMS



Fokus auf die Logik des Geschäftsprozesses, nicht auf einzelne Aufgaben.

## Einige Entwicklungen

- Netzwerke (Hard- und Software),
- Client-/Server-Systeme,
- Prozessdenken vs. Datendenken (z.B. ISO9000),
- Informationssysteme sind Dynamik ausgesetzt
  - kundenbezogen,
  - Vielzahl an Produkten und Dienstleistungen,
  - Kostenreduktion,
  - Interorganisationale Aufgaben (EDI - electronic data interchange)
- WFM-Systeme erhältlich.

# 1. Historie

## Definitionen:

Ein **Workflow-Management-System (WFMS)** ist ein Softwarepaket zur Unterstützung des Entwurfs und der Ausführung von Geschäftsprozessen.

Ein **Workflow-System (WFS)** ist ein auf einem WFMS basierendes System zur Unterstützung einer bestimmten Menge von Geschäftsprozessen durch Ausführung rechnergestützter Prozessmodelle.

## 2. Aufgaben eines WFMS

### Aufgaben eines WFMS



## 2. Aufgaben eines WFMS

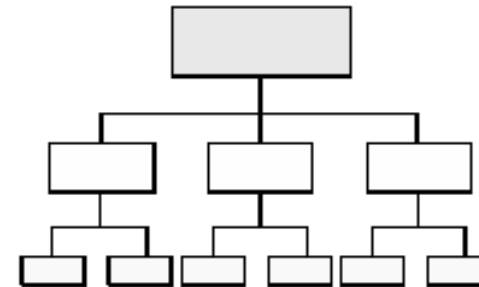
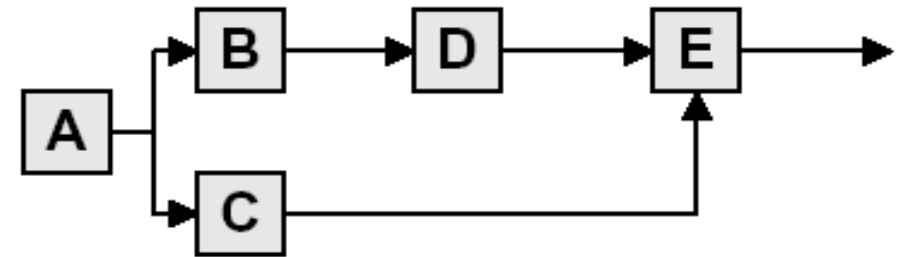
### Modellierung und Simulation von Workflows:

- Spezifizierung der notwendigen Workflowmodelle,
- Koordinierung des Einsatzes von Personen,
- Integration von Anwendungen und Daten,
- Formale und inhaltliche Überprüfung hinsichtlich Lauffähigkeit / Effizienz,
- ...

## 2. Aufgaben eines WFMS

### Modellierung und Simulation von:

- **Vorgangsstruktur:**
  - Vorgangsbeschreibung,
  - Arbeitsanweisungen.
- **Stellenstruktur:**
  - Kompetenzregelung,
  - Vertretungsregelung,
  - Aufgabendefinition.
- **Informationsstruktur:**
  - Dokumentenart,
  - Datenbanken,
  - Archive,
  - Zugriffsberechtigung.





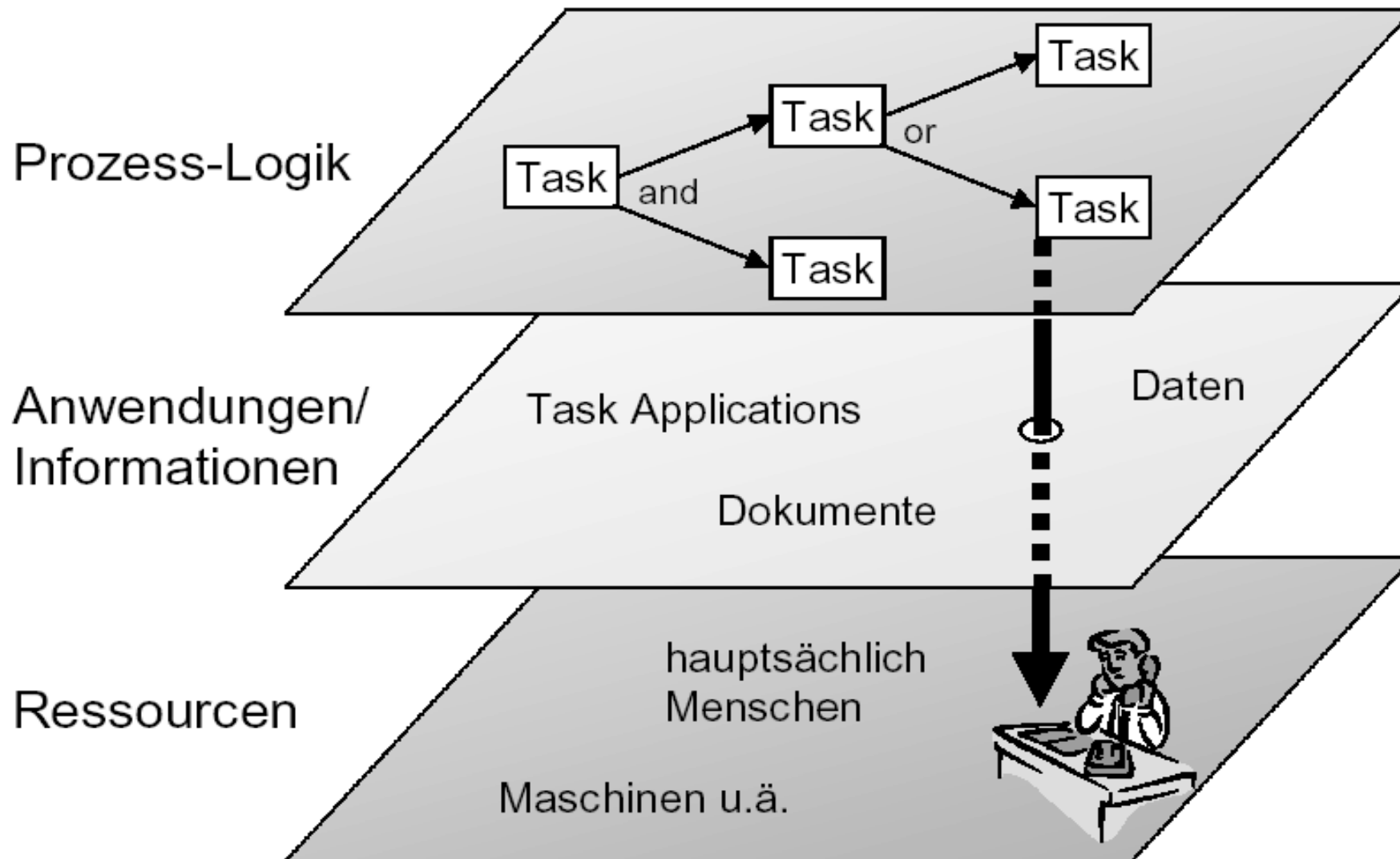
## 2. Aufgaben eines WFMS

### Instanziierung und Ausführung von Workflows:

- WFMS instanziiert konkrete fallbezogene Vorgänge,
- Ermittlung von Bearbeitern und Applikationen für jeden einzelnen Teilschritt,
- Beachtung dynamischer Restriktionen (z.B. Anwesenheitsstatus der Mitarbeiter),
- Benachrichtigung der Bearbeiter (Worklist),
- Aufruf von Applikationen und Datenübergabe,
- ...

## 2. Aufgaben eines WFMS

### Instanziierung und Ausführung von Workflows:

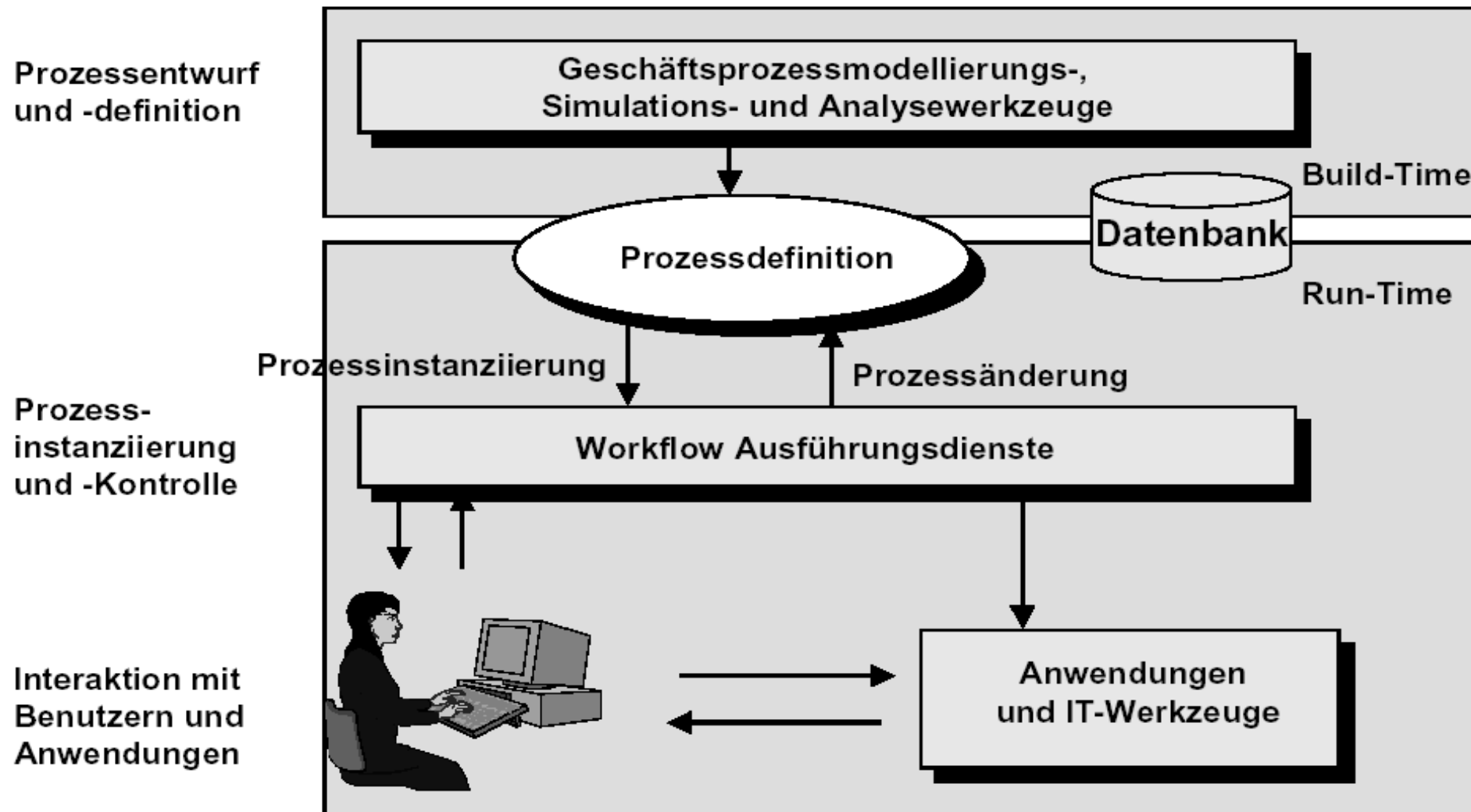


## 2. Aufgaben eines WFMS

### Monitoring laufender Vorgänge und nachträgliche Analyse:

- Bereitstellung von Statusinformationen über Vorgänge und Ressourcen,
- Überwachung von Start- und Ende-Terminen,
- Aktivierung von Ausnahmeroutinen (z.B. bei Krankheit, Urlaub eines Mitarbeiters),
- Sammeln von Ablaufdaten zur statistischen Auswerten (z.B. Data Mining über Prozessdaten),
- ...

### 3. Build-Time und Run-Time eines WFMS

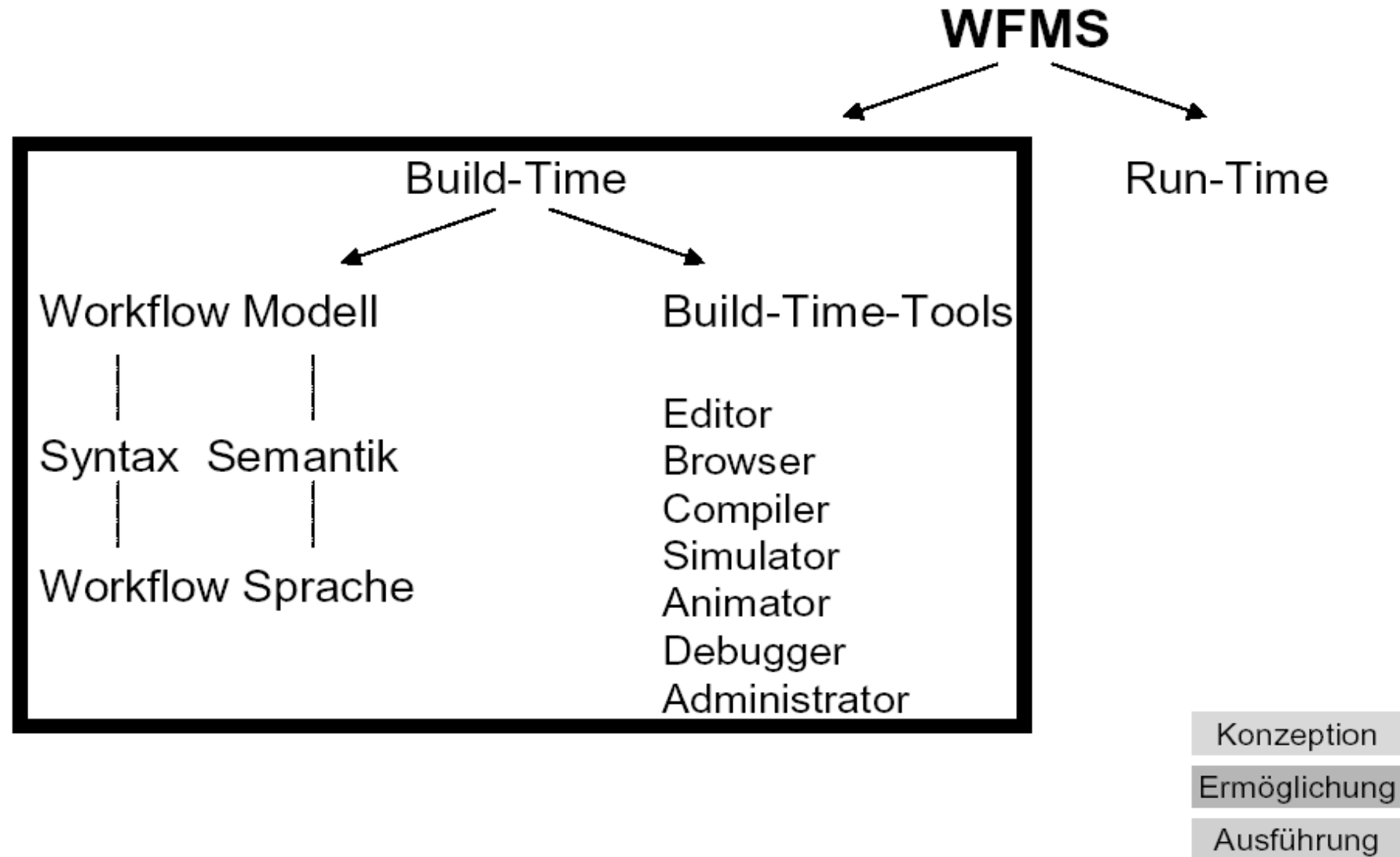


#### Zusammenwirken von Build-Time und Run-Time

(nach Dadam)

### 3. Build-Time und Run-Time eines WFMS

#### Bestandteile der Build-Time:



### 3. Build-Time und Run-Time eines WFMS

## Bestandteile der Build-Time:

### **Workflow Modell:**

- logische Grundlage,
- enthält alle Objekte und Relationen zum Workflow,
- bestimmt die Ausdrucksmächtigkeit

### **Workflow Sprache:**

- in formaler Grammatik,
- spezifiziert den Workflow,
- Konstrukte der WF-Sprache entsprechen denen des WF-Modells,
- Semantik: Transitionen ↔ Aufgaben, etc.
- Syntax: abh. von Präsentationsanforderungen.

### 3. Build-Time und Run-Time eines WFMS

## Bestandteile der Build-Time:

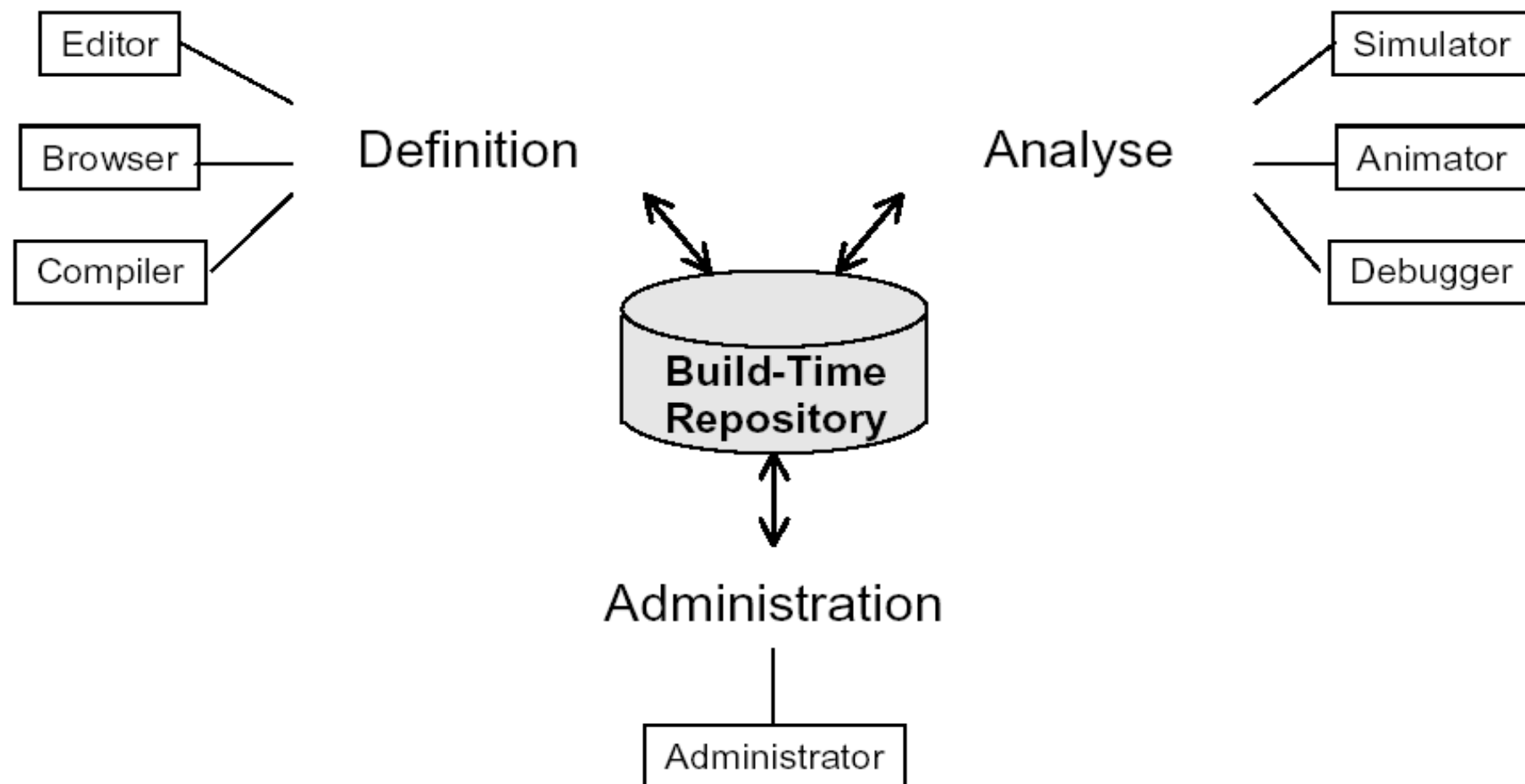
### Build-Time-Tools:

- Editor: zur Spezifikation der Workflows
- Compiler: Umsetzung der Workflow-Spezifikation
- Simulator, Animator: Überprüfung der Realitätstreue
- Administrator: Verwaltung der Workflow-Definition
- ...

### 3. Build-Time und Run-Time eines WFMS

#### Bestandteile der Build-Time:

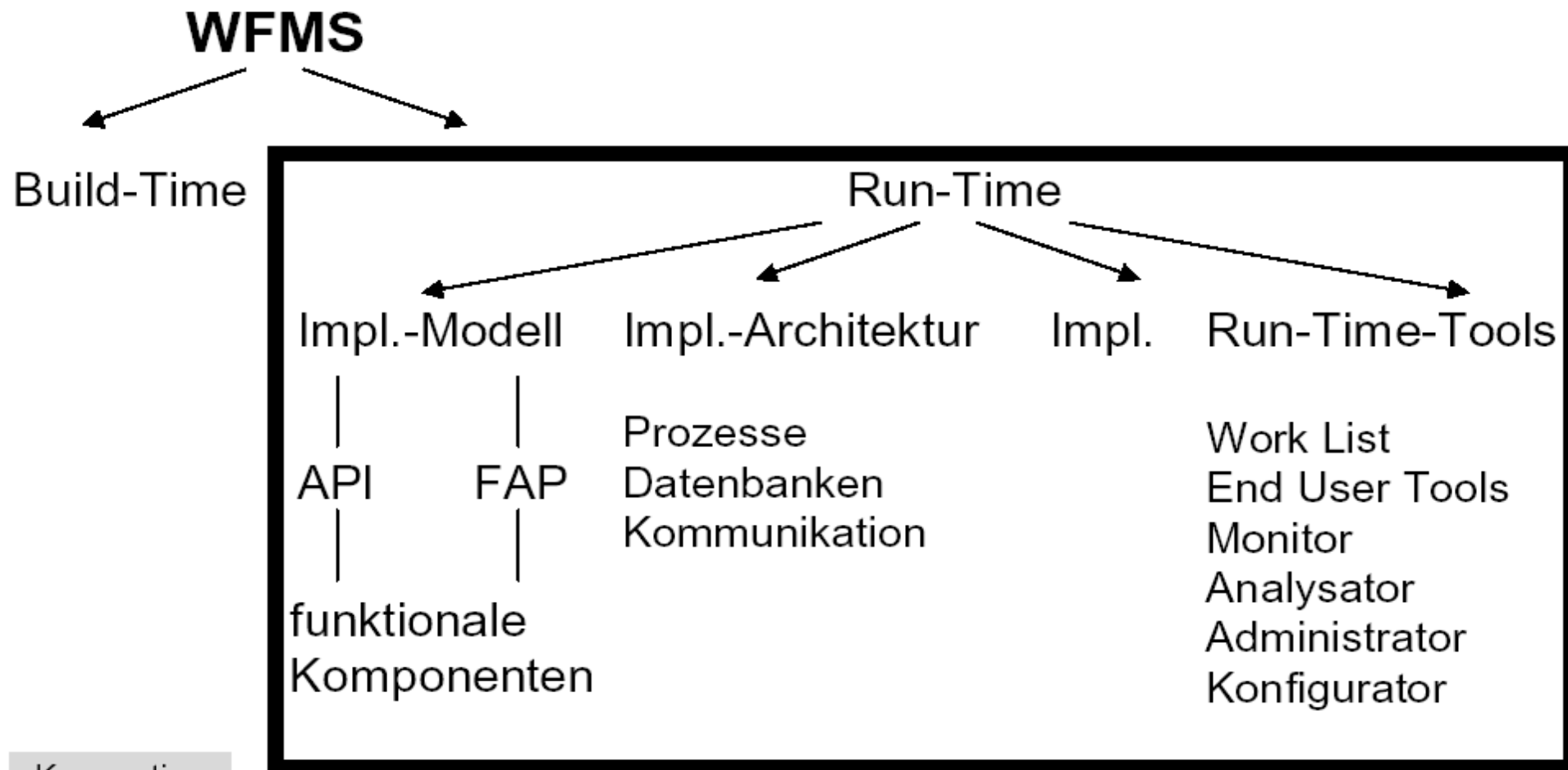
#### Zusammenhänge der einzelnen Build-Time-Tools:





### 3. Build-Time und Run-Time eines WFMS

#### Bestandteile der Run-Time:



Konzeption

Ermöglichung

Ausführung

### 3. Build-Time und Run-Time eines WFMS

## Bestandteile der Run-Time: (Ausführungsinfrastruktur)

### Implementierungs-Modell: (funktionale Architektur)

- Spezifikation funktionaler Komponenten als konzeptionelle Basis,
- Spezifikation der Protokolle (Definition des API und von Formaten und Protokollen (FAP))

### Implementierungs-Architektur:

- Umsetzung der funktionalen Architektur in aktive Elemente,
- Design von Datenbanken für notwendige Daten (Überwachungs- und Ausführungsdaten),
- Auswahl von Kommunikationsmechanismen für die Komponenten

Auswahl ist abhängig von Anforderungen wie Zuverlässigkeit, Skalierbarkeit, Robustheit und Managebarkeit.

### 3. Build-Time und Run-Time eines WFMS

## Bestandteile der Run-Time: (Ausführungsinfrastruktur)

### Implementierung:

- Umsetzung auf konkrete Systeme

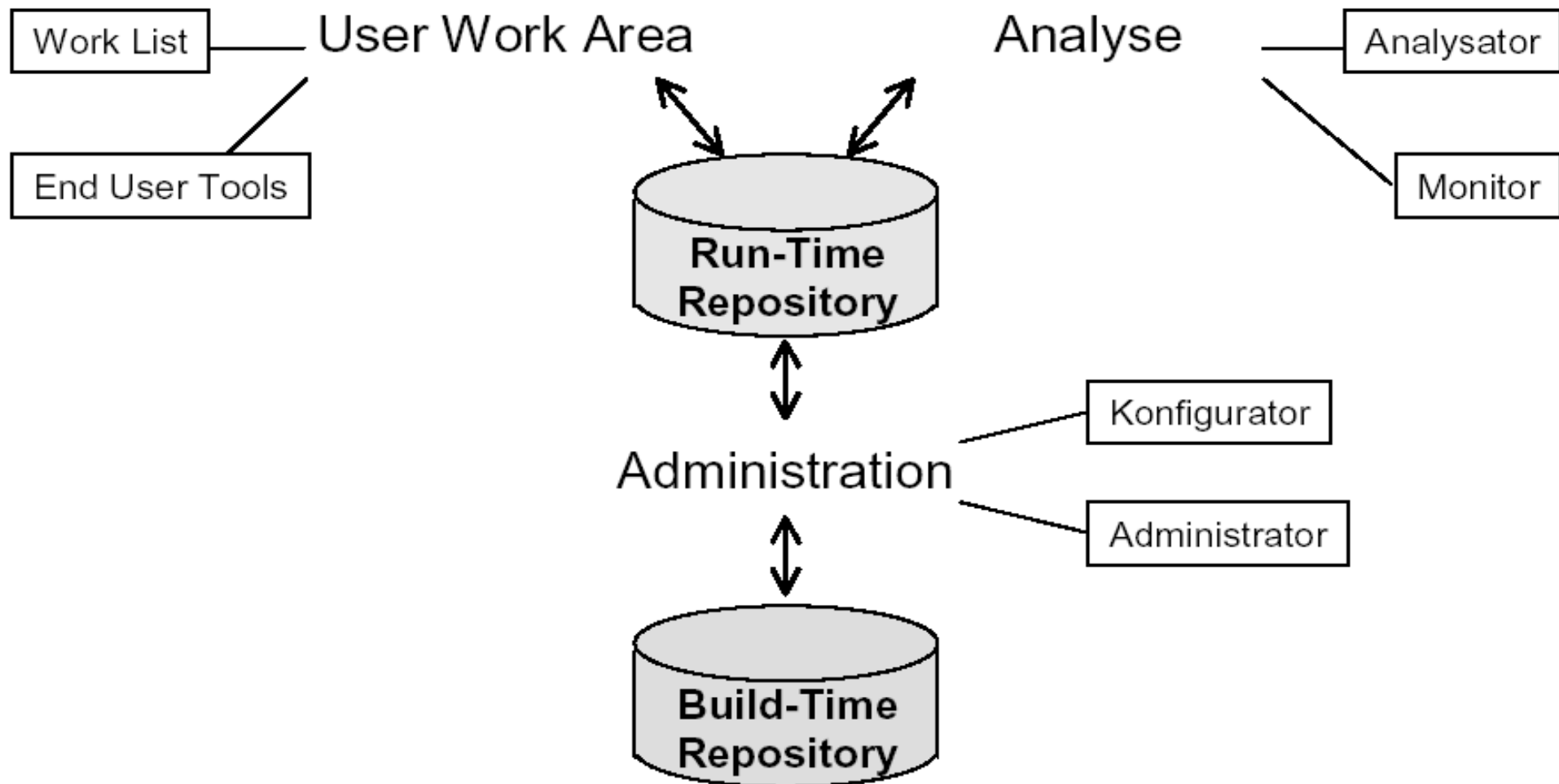
### Run-Time-Tools:

- Monitor: Überwachung und Kontrolle der Workflow-Ausführung,
- Analysator: Bewertung der Effektivität / Effizienz der Ausführung,
- Work List: um Benutzern ihre anfallenden Aufgaben mitzuteilen,
- ...

### 3. Build-Time und Run-Time eines WFMS

## Bestandteile der Run-Time: (Ausführungsinfrastruktur)

### Zusammenhänge der einzelnen Run-Time-Tools:



## 4. Anforderungen an ein WFMS

### WFMS sollten:

- sehr unterschiedliche Abläufe unterstützen können,
- spätere Anforderungen bearbeiten können
- **erweiterbar** sein
- schnell auf Änderungen reagieren können,
- die dynamische Anpassung von Abläufen unterstützen,
- **dynamisch anpassbar** sein.
- auf vorhandene Modellierungen zurückgreifen können (z.B. einen vorh. Workflow als Subworkflow eines neuen WF definieren können)
- Teile sollen **wieder verwendbar** sein

## 4. Anforderungen an ein WFMS

### WFMS sollten:

- bereits existierende Hard- und Softwaresysteme, die noch benötigt werden, einbinden können  
→ **offen** sein
- Prozesse effizient ausführen, besonders auch bei einer hohen Anzahl an Workflow-Instanzen,  
→ **skalierbar** sein.

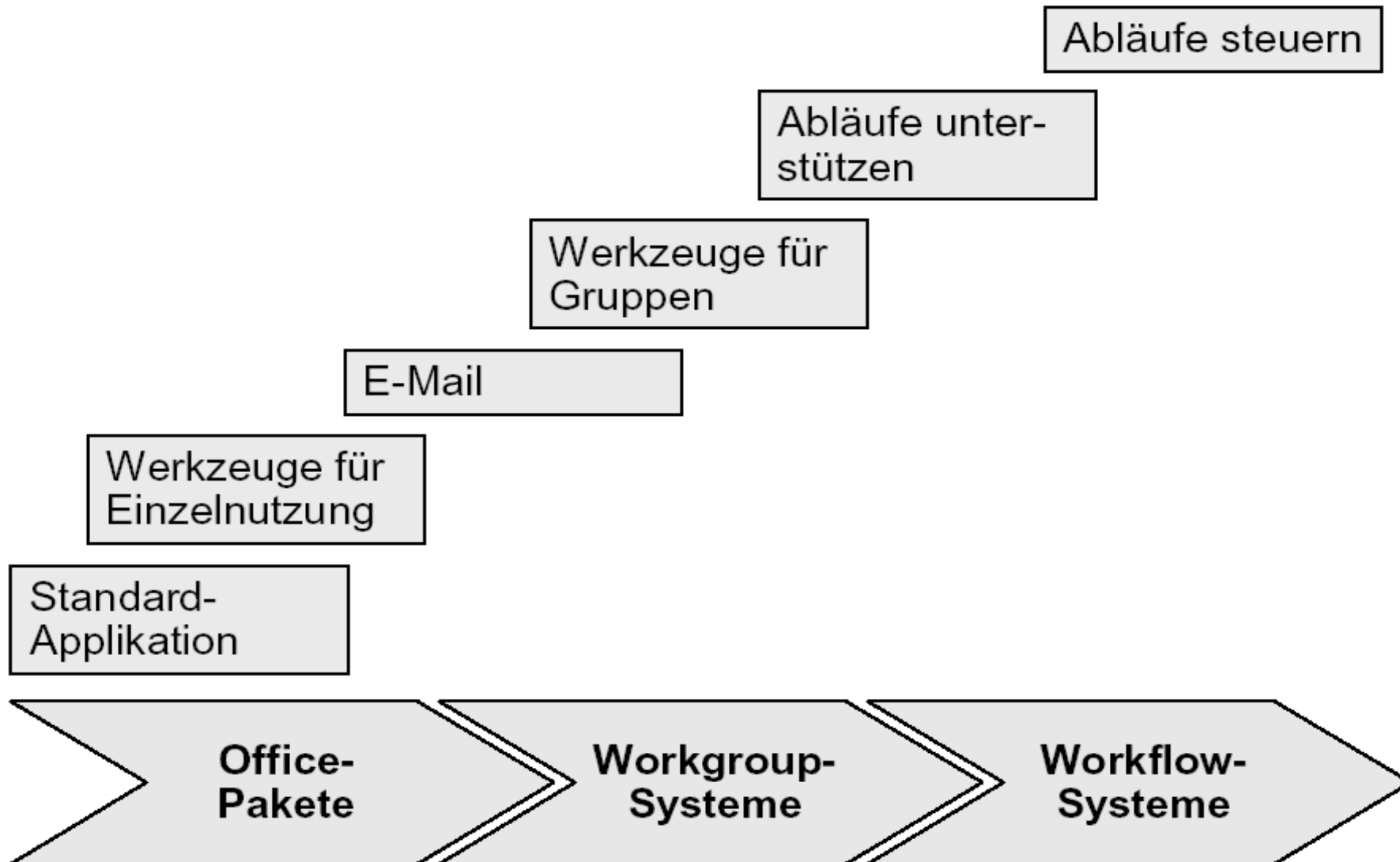
## 4. Anforderungen an ein WFMS

### Anforderungen an die Implementierung:

- **Effizienz**  
(messbar durch: Performance, Antwortzeiten, Durchsatzzeiten)
- **Zuverlässigkeit**  
(Ausfall-, Fehlerwahrscheinlichkeit, Wiederherstellungsmöglichkeit)
- **Korrektheit**  
(Konsistenz, Vollständigkeit, Test-, Überprüfbarkeit, Präzision)
- **Wartbarkeit**  
(besonders für das Anbinden anderer Systemkomponenten)
- **Übertragbarkeit**  
(Anwendung in anderen Infrastrukturen)
- **Modularität.**

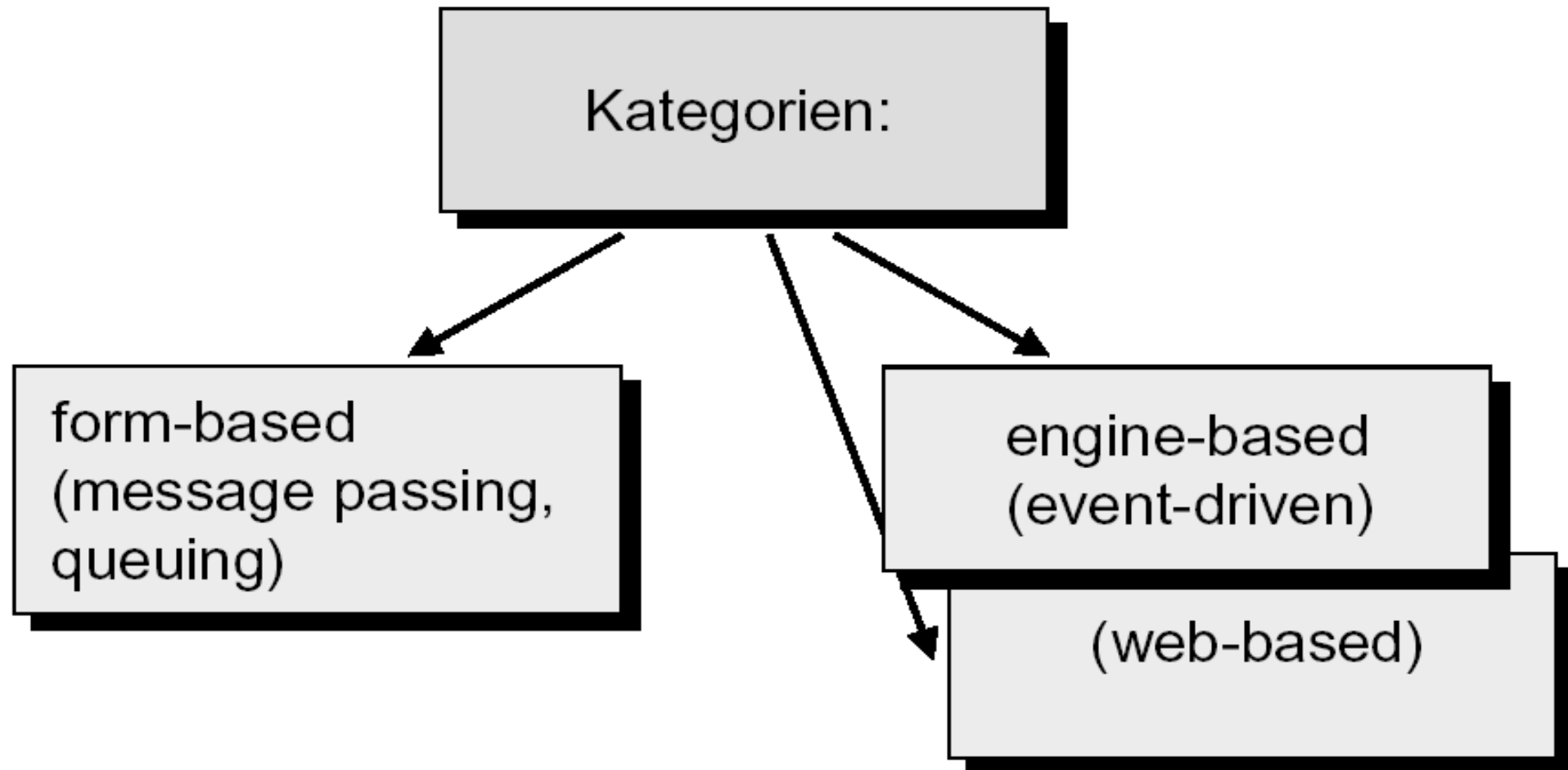
## 5. WFMS-Architekturen

### Geschichte und Funktionalität der Systeme:





## 5. WFMS-Architekturen



### „form-based“ Architekturen:

- Messaging Service leitet durch die vordefinierten Tasks,
- Prozess-Logik geschieht durch Interaktion von Eingabemasken und Mail-System,
- User-Interface ist/sind dabei die Eingabemaske/n,
- Masken können auf eine Datenbank zugreifen,
- die einzige Interaktion zwischen den Tasks geschieht durch persistente Messages, die über die Beendigung der vorhergehenden Task informieren.

### „form-based“ Architekturen:

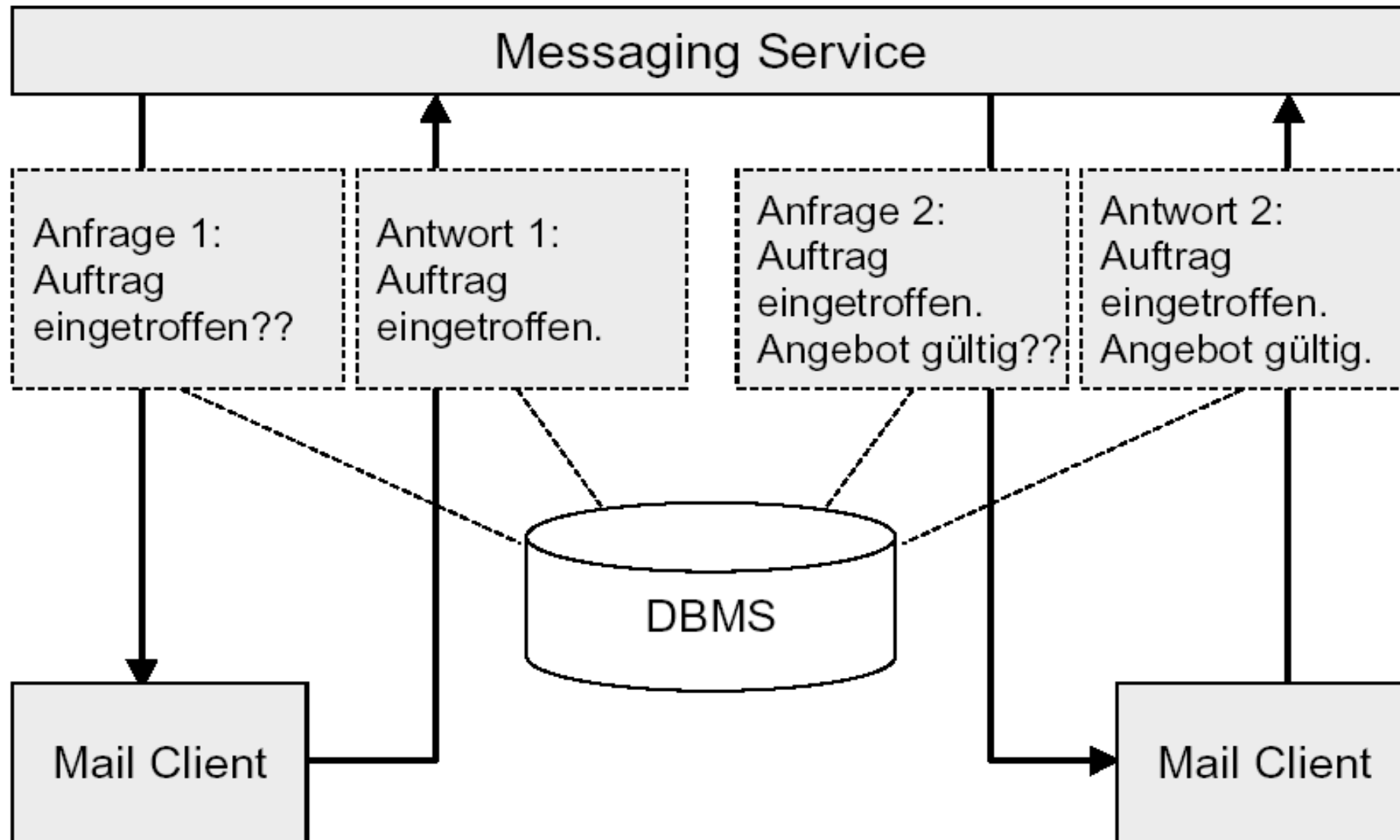
- Local Queue wird bereitgestellt,
- Anwendungen legen dort Messages ab, bzw. erhalten sie von dort,
- Queues sind persistent, so dass die Messages bei Ausfällen nicht verloren gehen,
- Queues machen asynchrone Kommunikation zwischen Tasks möglich,
- Queues erlauben Kommunikation zwischen verschiedenen Plattformen und Protokollen.

### „form-based“ Architekturen:

- + verhindert Performance-Engpass, der während der Ausführung durch die Kommunikation mit dem Server entsteht,
- + stabil gegen Störungen: Ausfall einer einzigen Task beeinflusst die anderen aktiven Tasks nicht,
- + Unabhängigkeit jeder einzelnen Task,
- keine Möglichkeit, Ereignisse zu protokollieren und damit zu überwachen,
- sehr geringe Management-Möglichkeit.

## 5. WFMS-Architekturen

Beispiel: ‚form-based‘



## 5. WFMS-Architekturen

### „engine-based“ Architekturen:

- Datenbanken enthalten
  - Prozess-Definition,
  - Zustand der Prozess-Instanzen in der Prozess-Umgebung,
- Server-basierter Enactment-Service (d.h. Engine),
- Client-Software auf den Desktops,
- fast immer „zugeschnittene“ Systeme.

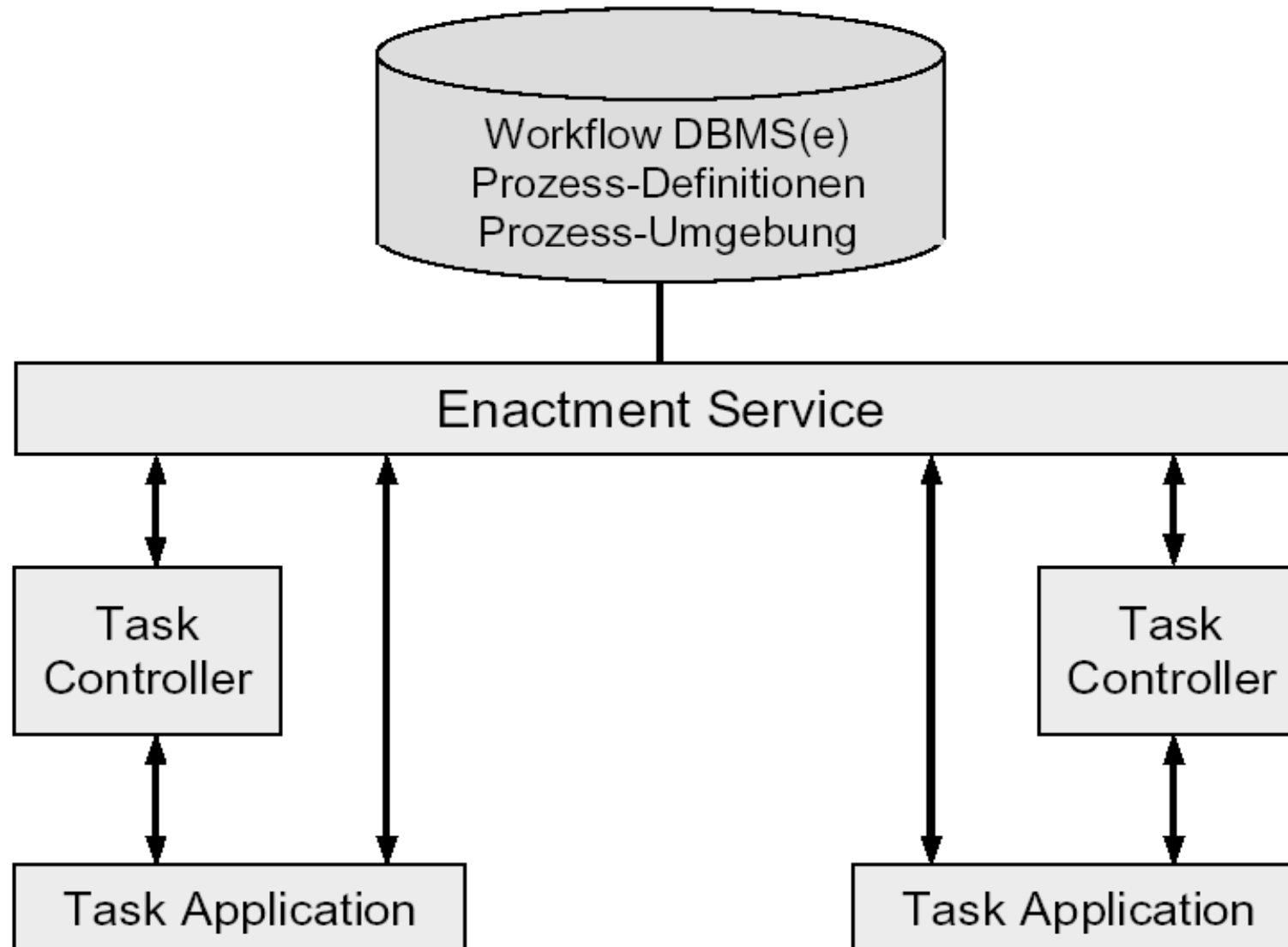
## 5. WFMS-Architekturen

### „engine-based“ Architekturen:

- + einfache Definition des Workflows durch den User,
- + Workflow kann danach exakt ausgeführt werden,
- + Analyse des gesamten Workflows möglich,
- Modifikation des aktiven Workflow endet in System-Fehlfunktion.

## 5. WFMS-Architekturen

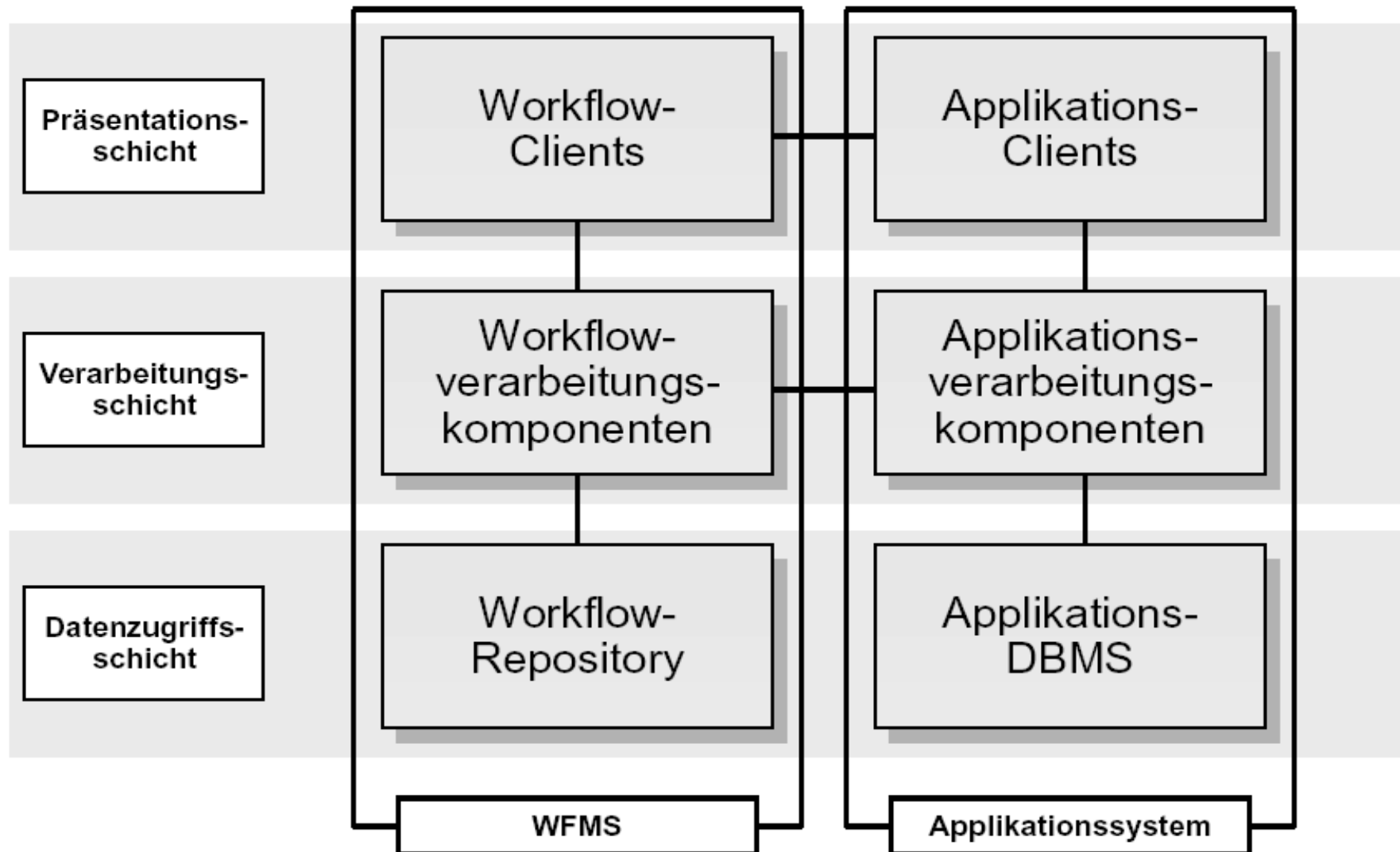
Beispiel: ‚engine-based‘





## 5. WFMS-Architekturen

### Client/Server-Schichtenmodell für WFMSe



### Client/Server-Schichtenmodell für WFMSen

Aufgabenteilung zwischen prozesssteuernden WFMSen und unterstützenden Applikationen:

#### **Präsentationsschicht:**

- Beinhaltet Bedieneroberfläche und dient der Benutzerführung/-interaktion
- WFMS: Dialogprogramme zur Prozess-Steuerung,
- Applikation: Dialogprogramme zur problembezogenen Ausführung,

#### **Verarbeitungsschicht:**

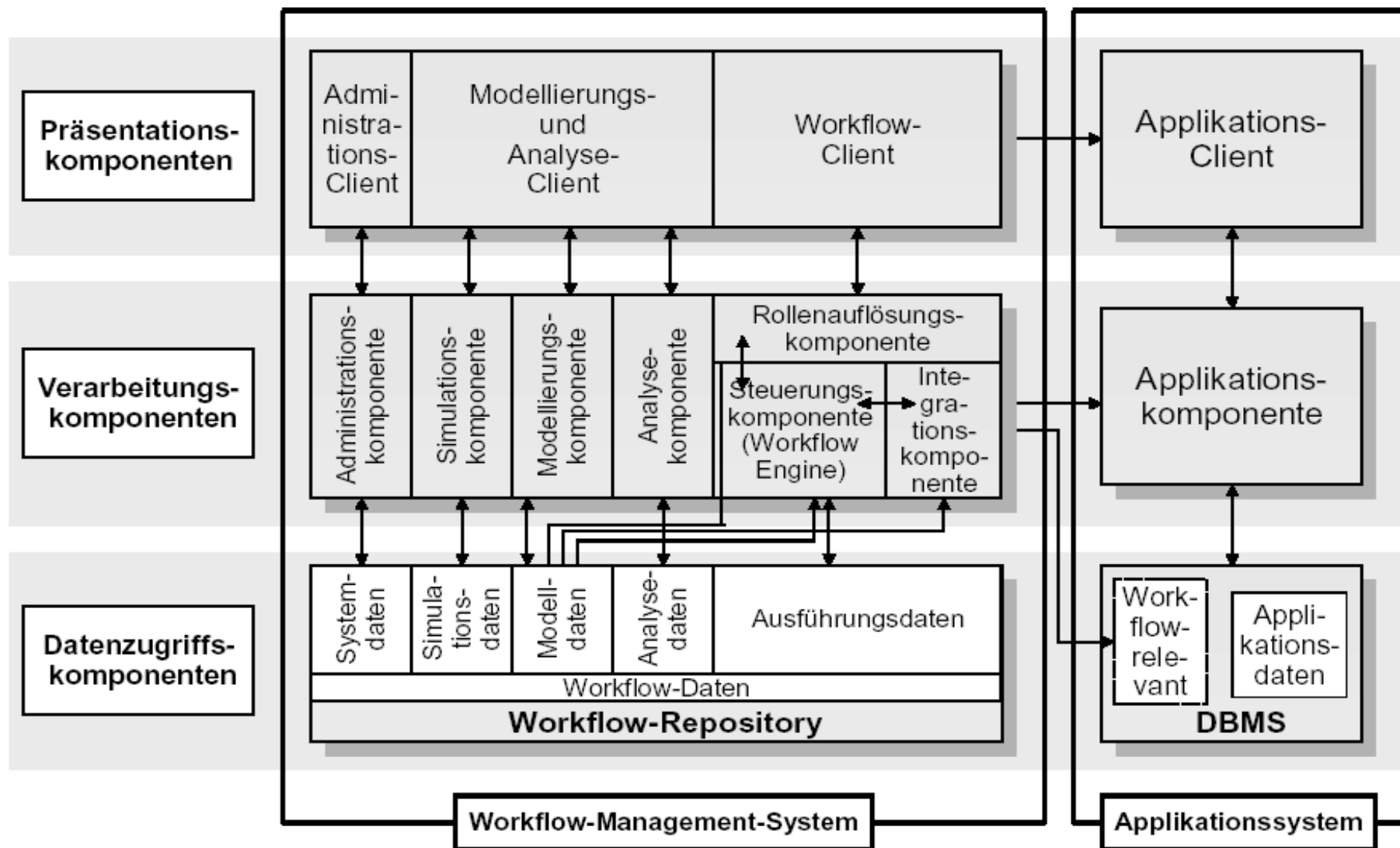
- Stellt Funktionen für Prozess-Steuerung bzw. Ausführung der Aufgaben bereit,
- Workflow-Verarbeitungsmodule rufen Applikationsverarbeitungsmodule auf

#### **Datenzugriffsschicht:**

- Dient der Verwaltung der Workflow-Daten des WFMS sowie der Applikationsdaten der Applikationssysteme.

## 5. WFMS-Architekturen

### Rahmenarchitektur für WFMSe



## 6. Implementierung eines Workflow-Designs

### Vorgehensmodell (allgemein)

	Projekt- vorbereitung	Analyse	Konzeption	Realisierung
Aufgabe	<ul style="list-style-type: none"> <li>• Projektauftrag definieren</li> <li>• Projektorganisation festleg.</li> <li>• Projektablauf planen</li> </ul>	<ul style="list-style-type: none"> <li>• Anforderungen definieren (IT, WF)</li> <li>• Erheben und Auswerten von Infos u. Daten</li> </ul>	<ul style="list-style-type: none"> <li>• Alternative Konzepte entwickeln und bewerten</li> <li>• Auswahl einer Alternative</li> </ul>	<ul style="list-style-type: none"> <li>• Feinkonzept entwickeln und implementieren</li> <li>• Tests (System, Aktivitäten, WF)</li> </ul>
Tätigkeiten	<ul style="list-style-type: none"> <li>• Gespräche mit Unternehmensleitung und Mitarbeitern aus dem Planungsumfeld</li> </ul>	<ul style="list-style-type: none"> <li>• Erheben der Infos u. Daten</li> <li>• Auswerten der Infos u. Daten</li> </ul>	<ul style="list-style-type: none"> <li>• Aktivitätenplanung, etc.</li> <li>• Aufbau Bewertungsschema / Bewertung der Alternativen</li> </ul>	<ul style="list-style-type: none"> <li>• ggf. Ausschreibung durchführen</li> <li>• Umsetzung des Konzepts</li> <li>• Tests durchführen</li> </ul>
Beteiligte	Unternehmensleitung	Mitarbeiter des Planungsumfeldes	Mitarbeiter des Planungsumfeldes	Mitarbeiter aus dem Planungsumfeld und Entwickler / Lieferanten
Ergebnis	Genehmigter Projektauftrag	Anforderungen / Schwachstellen erkannt	Genehmigtes Konzept	System ist im Produktivbetrieb

## 6. Implementierung eines Workflow-Designs

Vorgehen bei der Einführung eines existierenden WFMS:

- Bereitstellung der technischen Infrastruktur,
- Installation des ausgewählten WFMS,
- Konfiguration des WFMS,
- Implementierung neuer Applikationen zur Unterstützung von Aufgaben,
- Einbindung alter Applikationen
  - Altsysteme (Legacy Systems),
  - Emulation,
  - Screen Scraping,
  - ...
- Testen und Validierung,
- Motivation und Training der Angestellten.

## 7. Konzeptuelle Alternativen

Es gibt verschiedene Philosophien und Möglichkeiten bei der Umsetzung der Workflow-Produkte hinsichtlich der fundamentalen Konzepte:

1. Prozess-Logik,
2. Zuordnung von Ressourcen zu Tasks,
3. Bereitstellung von Anwendungen/Informationen für Tasks,
4. Prozess-Management.

# 1. Prozess-Logik

## WFMS

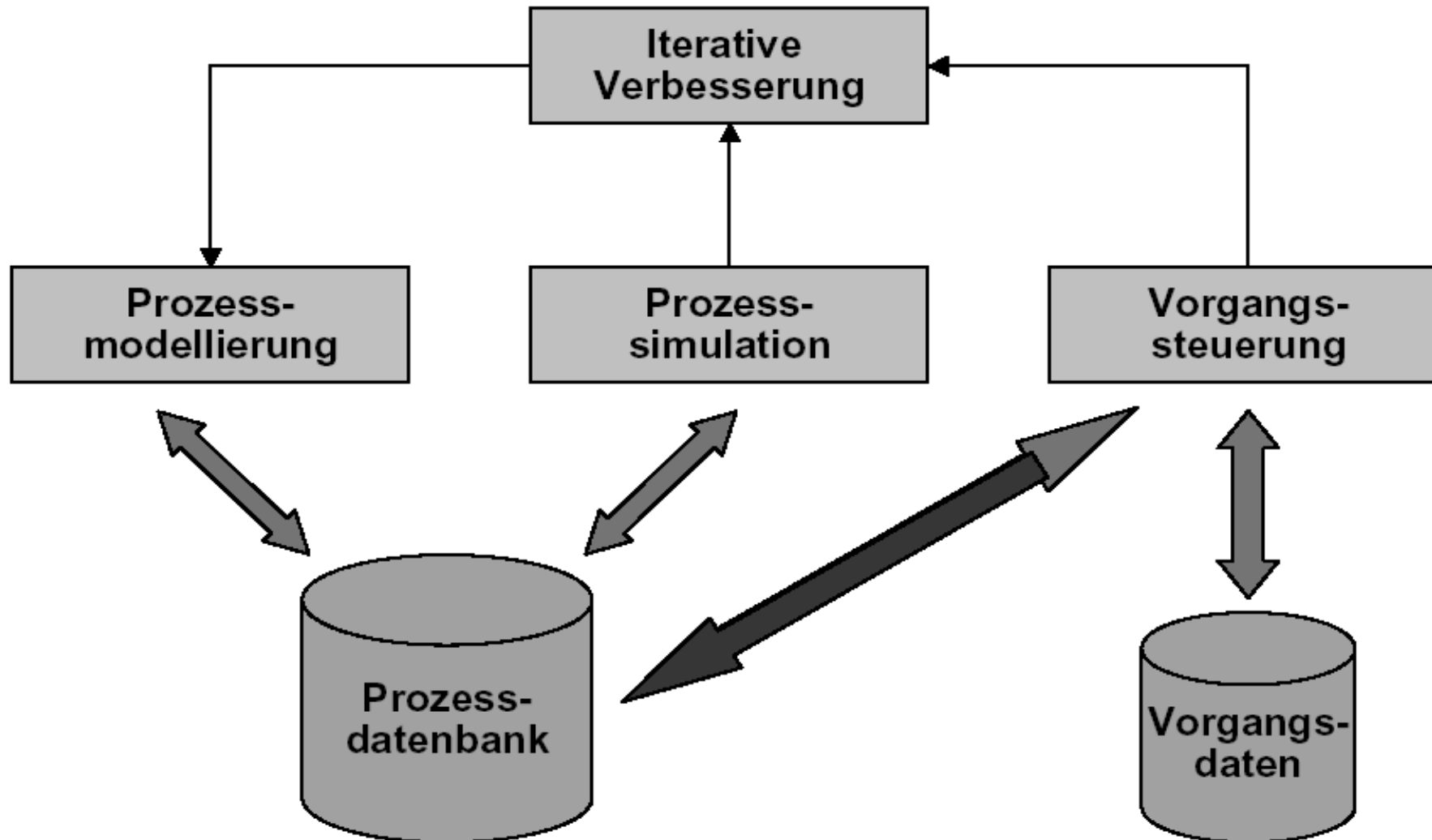
- liefert Rechnerunterstützung für die Prozess-Logik,
- es kann die Prozess-Abläufe sowohl erzwingen als auch dokumentieren,
- gibt die Verantwortlichkeit gemäß der Prozess-Logik von einer Person zur Nächsten, bzw. von einer Task zur Nächsten weiter,
- dokumentiert die einzelnen Prozess-Phasen.

### 1. Prozess-Logik

- Möglichkeit der Workflow-Definition durch den End-User  
↔ Workflow-Design nur Spezialisten erlaubt
- einsatzbereite Systeme, nach Definition der Prozess-Logik (keine aufwendige Integration von Informationen notwendig)  
↔ Prozess-Design ist nur Teil der spezialisierten Systemintegration (Verknüpfung von Prozess-Definition und Anwendungen)
- dynamische Ansätze:  
Möglichkeit zur Anpassung der Prozess-Logik  
↔ vollständige Prozess-Definition vor Prozess-Ausführung (Änderung erst bei erneuter Ausführung wirksam).



### Iterative Weiterentwicklung des Workflow Systems:



## 2. Zuordnung von Ressourcen und Tasks

- System wählt die Zuordnung aus (verantwortlich)
  - Rotierend,
  - anhand von gesammelten Performance-Daten,
  - anhand aktueller Auslastung, Verfügbarkeit, etc.
- Alternative: System bietet Tasks an, die nach Zuständigkeit entnommen werden können (Ressource ist selbst verantwortlich),
- einige Systeme erlauben eine Neuzuordnung von Ressourcen und Task.

### 3. Bereitstellung von Anwendungen / Dokumenten / Daten für Tasks

- Möglichkeit eigene Task-Applikationen zu entwickeln,
- Möglichkeit der Integration externer Applikationen,
- freie Zuordnung zu verschiedenen Tasks  
↔ konkrete Zuordnung (weniger flexibel),
- Unterstützung der Bearbeitung von Dokumenten/Daten (Bilder, Formulare, Tabellen, etc.).

## 4. Prozess-Management:

- durch die Separation der Prozess-Logik vom Prozess-Design sind Veränderungen und ein Prozess-Management überhaupt erst möglich,
- Anpassungsmöglichkeiten der Prozessdefinition (z.B. für Verbesserungen bei evtl. ‚bottlenecks‘, als Antwort auf ‚changing needs‘),
  - evtl. Regeln zur dynamischen Anpassung,
  - Loadbalancing,
  - automatische Anpassung des Folgeprozesses,
- Verbindungen zu Zeit-, Kosten- und Qualitätsmanagement,
- Prüf- und Analysefähigkeiten,
- Ziel: bessere Ressourcenauslastung.

## 8. WfMC-Referenzmodell



### Workflow Management Coalition

- 1993 gegründet,
- internationale non-profit-Organisation,
- mehr als 130 Mitglieder: Hersteller, Anwender, Systemanalytiker, Forscher,
- Ziele:
  - Verbreitung des Einsatzes von „Workflow“,
  - Entwicklung und Vermarktung von Standards in der Software-Terminologie,
  - Erreichen einer Interoperabilität zwischen verschiedenen Workflow-Produkten.

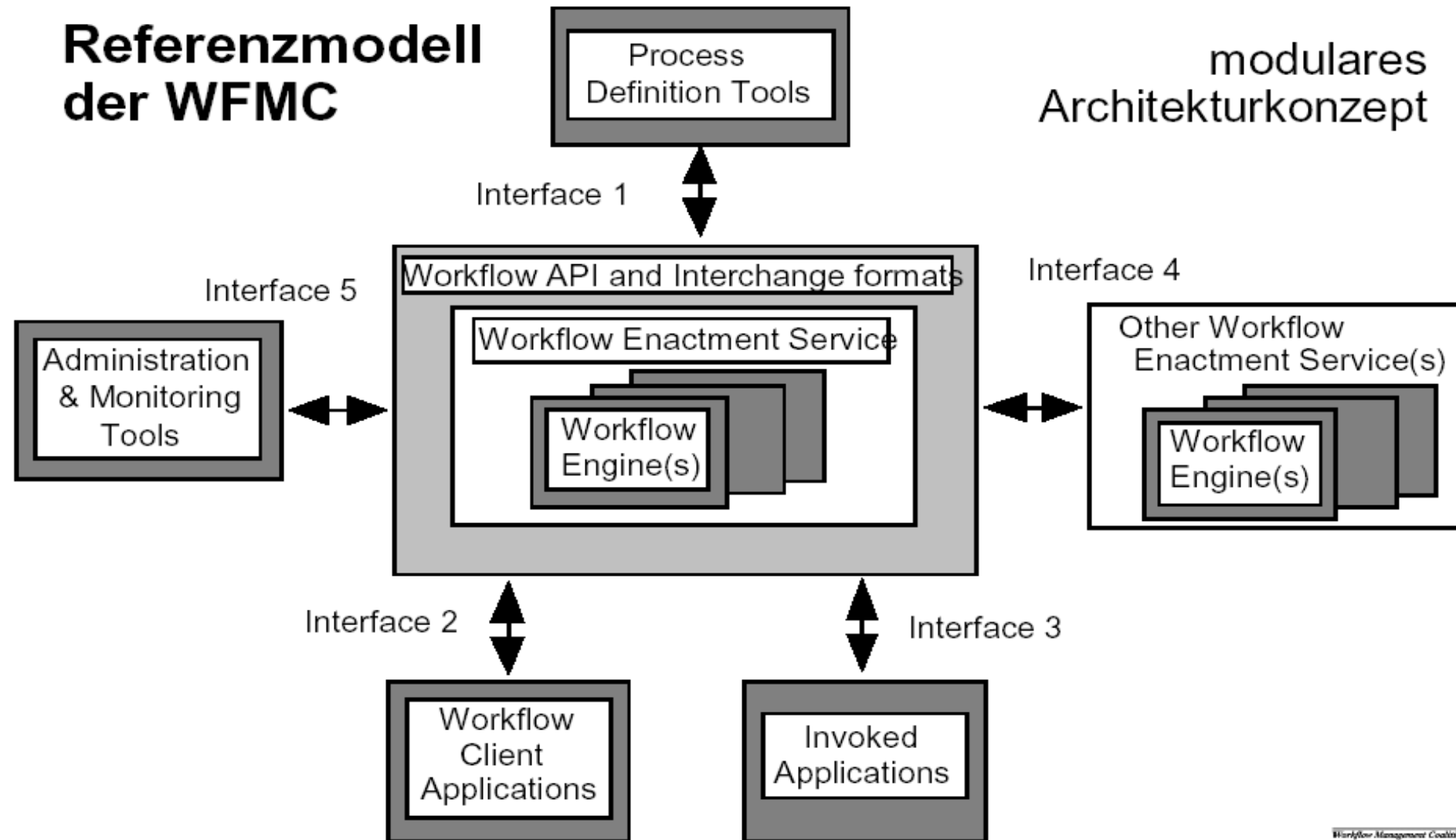
### Das Referenzmodell – Überblick

- entwickelt aus einer generischen Struktur eines Workflow-Management-Systems,
- die einzelnen Interfaces innerhalb dieser Struktur sollen den Produkten die Interoperation auf verschiedenen Niveaus ermöglichen,
- jedes Workflow Management System enthält mehrere generische Komponenten, die auf eine vorgegebene Weise interagieren sollen,
- Notwendigkeit von standardisierten Interfaces und ‚data interchange formats‘.
- Quelle: <http://www.wfmc.org/reference-model.html>

### Überblick

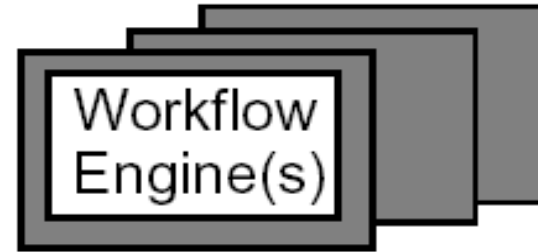
- Spezifikation einer Struktur für WFMS (Charakteristika, Funktionen, Schnittstellen),
- Einführung von Standards in dieser Struktur,
- Ermöglichung der Zusammenarbeit verschiedener Systeme,
- fünf Kategorien der Schnittstellen- u. Kommunikationsstandards.

## 8. WfMC-Referenzmodell





## 8. WfMC-Referenzmodell

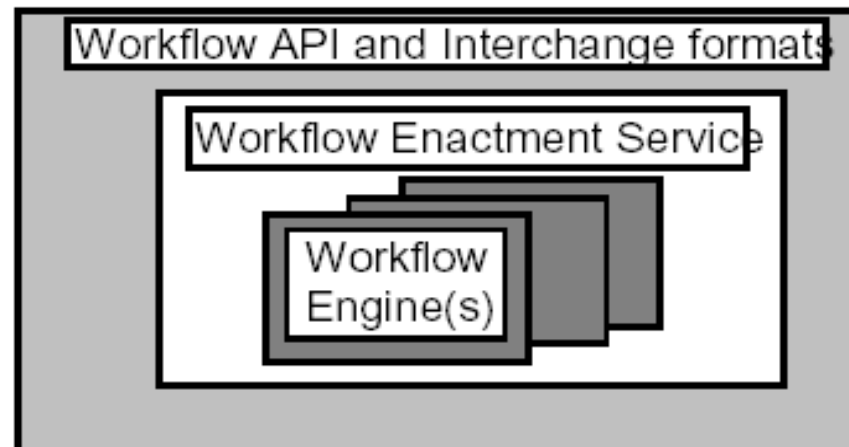


### Workflow Engine

- ist Softwarekomponente, die Laufzeitunterstützung für die Ausführung zur Verfügung stellt,
- generiert aus den Prozessdefinitionen Instanzen und arbeitet diese ab (Kreation, Aktivierung, Anhalten, Beenden, etc.),
- Navigation zwischen Prozessaktivitäten,
- Interface, um User-Interaktionen zu unterstützen,
- Austausch workflow-relevanter Daten zwischen User und Anwendung,
- Supervisor-Aktionen zur Steuerung, Administration und Überprüfung.

### Workflow Enactment Service

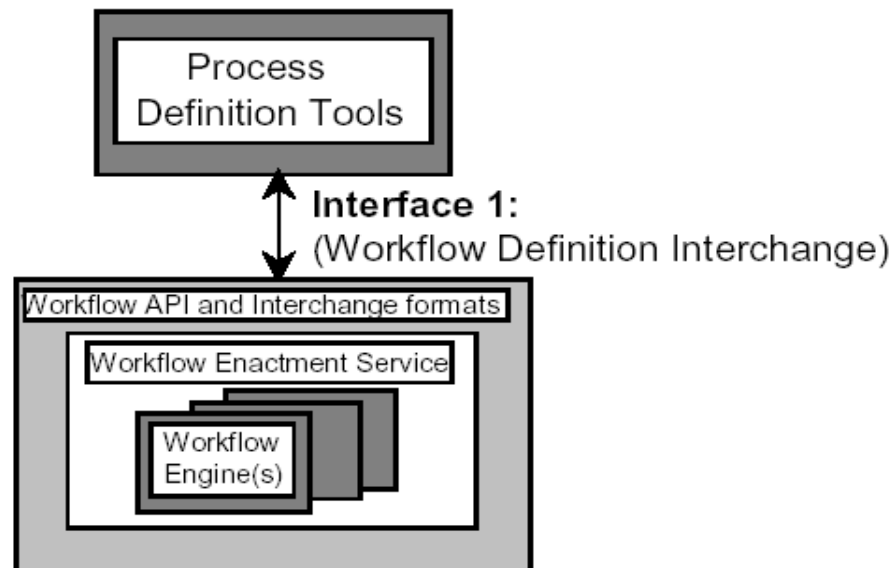
- besteht aus einer oder mehreren Workflow-Engines



- WAPI and Interchange formats dienen
  - standardisiertem Funktionsaufruf zwischen Systemkomponenten (Anwendung ohne Anwenderaktion)
  - Formatanpassung.

### Process Definition Tools

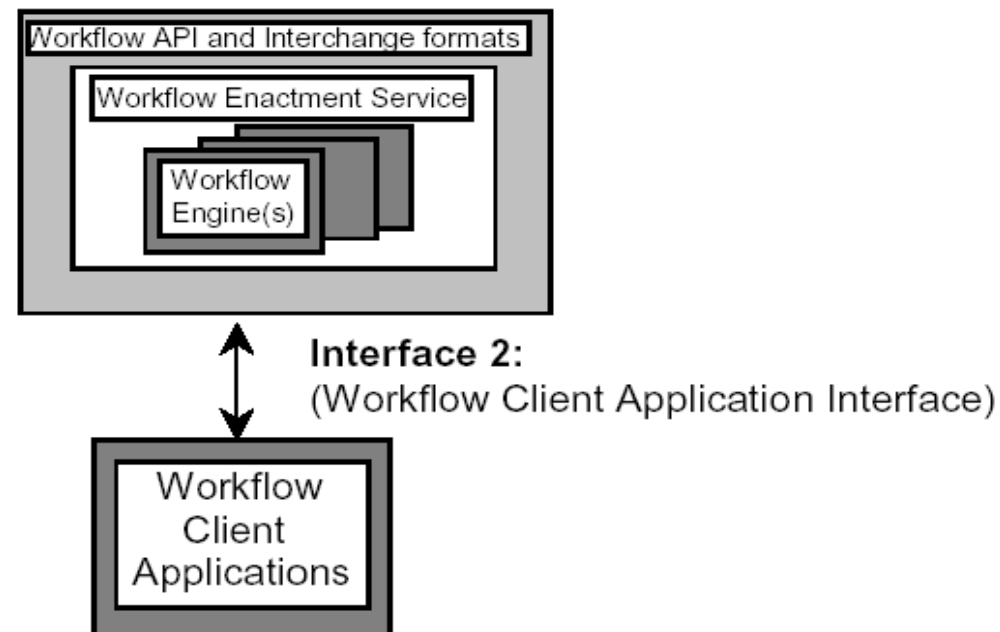
- Werkzeuge zur Erstellung der Prozessdefinition,
- Analyse, Modellierung, Beschreibung, Dokumentation des Prozesses
  - können vom Workflow-Management-Produkt selbst angeboten werden oder sind eigenständige Tools,
  - Übergabe einer Prozessdefinition, die zur Laufzeit von der Workflow-Engine interpretiert werden kann



**Ziel:** durch z.B. Metamodell sollen Prozessbeschreibungen aus Tools div. Hersteller eingebunden werden können

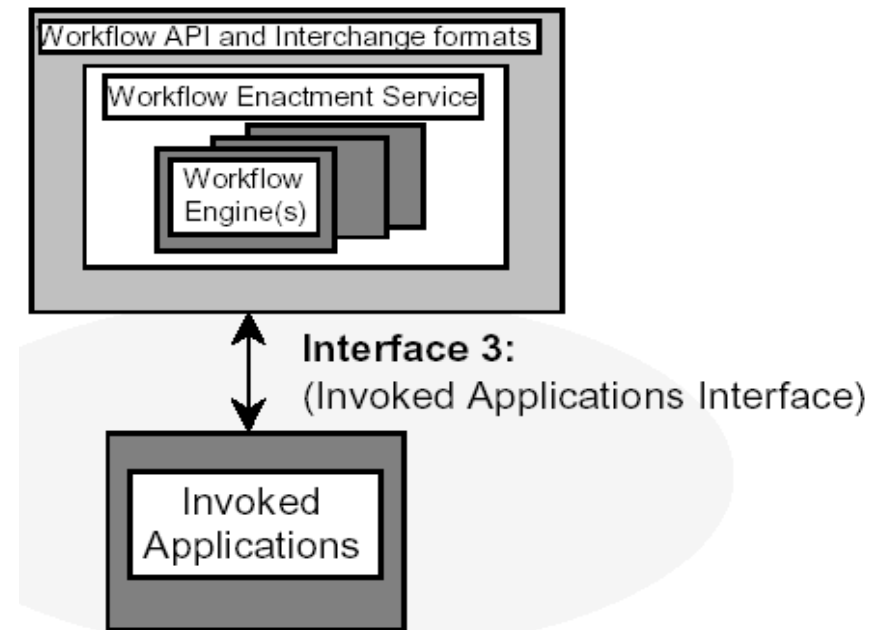
### Workflow Client Applications

- Zusammenarbeit mit den Anwendern,
- Aktivierung spezieller Applikationen,
- können ebenfalls zum WFMS gehören,
- Konzept: Worklist – Liste aller Arbeitsschritte und der zugehörigen User



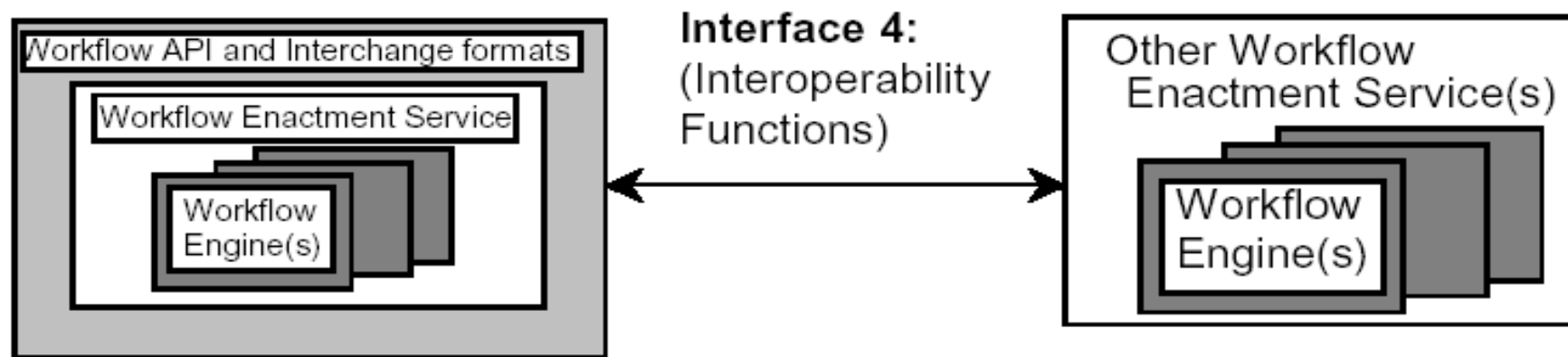
### Invoked Applications

- auch: ‚workflow-enabled‘ Tools (Standard API's),
- aus allen möglichen Plattformen und Netzwerkumgebungen,
- z.B. Übersetzungsprogramme, Fakturaprogramme, etc...
- z.Z. eher eingeschränkte Auswahl an einbindbaren Applikationen
- Konzept: Application Agent



### Workflow-Interoperabilität

- aufgrund verschiedener WFS-Hersteller existiert eine Diversität (z.B. ad-hoc – Systeme ↔ Produktionsworkflow),
- Ziel: systemübergreifende Zusammenarbeit,
- unwahrscheinlich in der nahen Zukunft, da alle Workflow-Engines eine gemeinsame Prozess-Definition interpretieren können müssen, Daten teilen müssen,...
- nahes Ziel: die Möglichkeit, Teile eines Prozesses an einen weiteren Enactment-Service zu schicken



### Administration und Monitoring

- Tools für Monitoring: Zustand/Auslastung der laufenden Instanzen (Rechen- und Speicherlast des Systems)
- Protokollierung zur nachträglichen Fehlererkennung und Auswertung
- Tools zur Benutzerverwaltung
- Tuning-Tools: Anpassung technischer Parameter.

