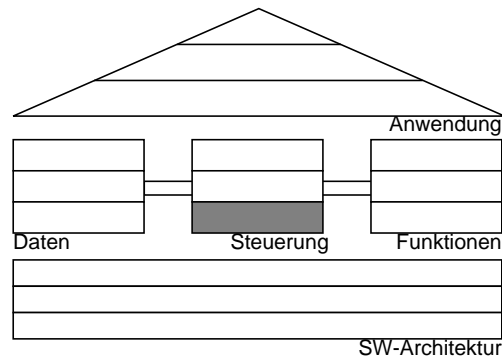


# 8. Workflow-Management

- **GBIS-Rahmen: Einordnung**



- **Aspekte von Workflows**

- Was, Wer, Womit, Wann
- Aspektorientierte Modellierung

- **Zusammenspiel der Komponenten**

- Workflow-System
- Workflow-Referenzmodell der WfMC
- Ausführung von Workflows

- **Anforderungen an ein WfMS**

- **Funktionale Architektur eines WfMS**

- Aspekte von Workflows
- Komponenten der WfMS-Ausführungskomponente

- **Transaktionen und Workflows**

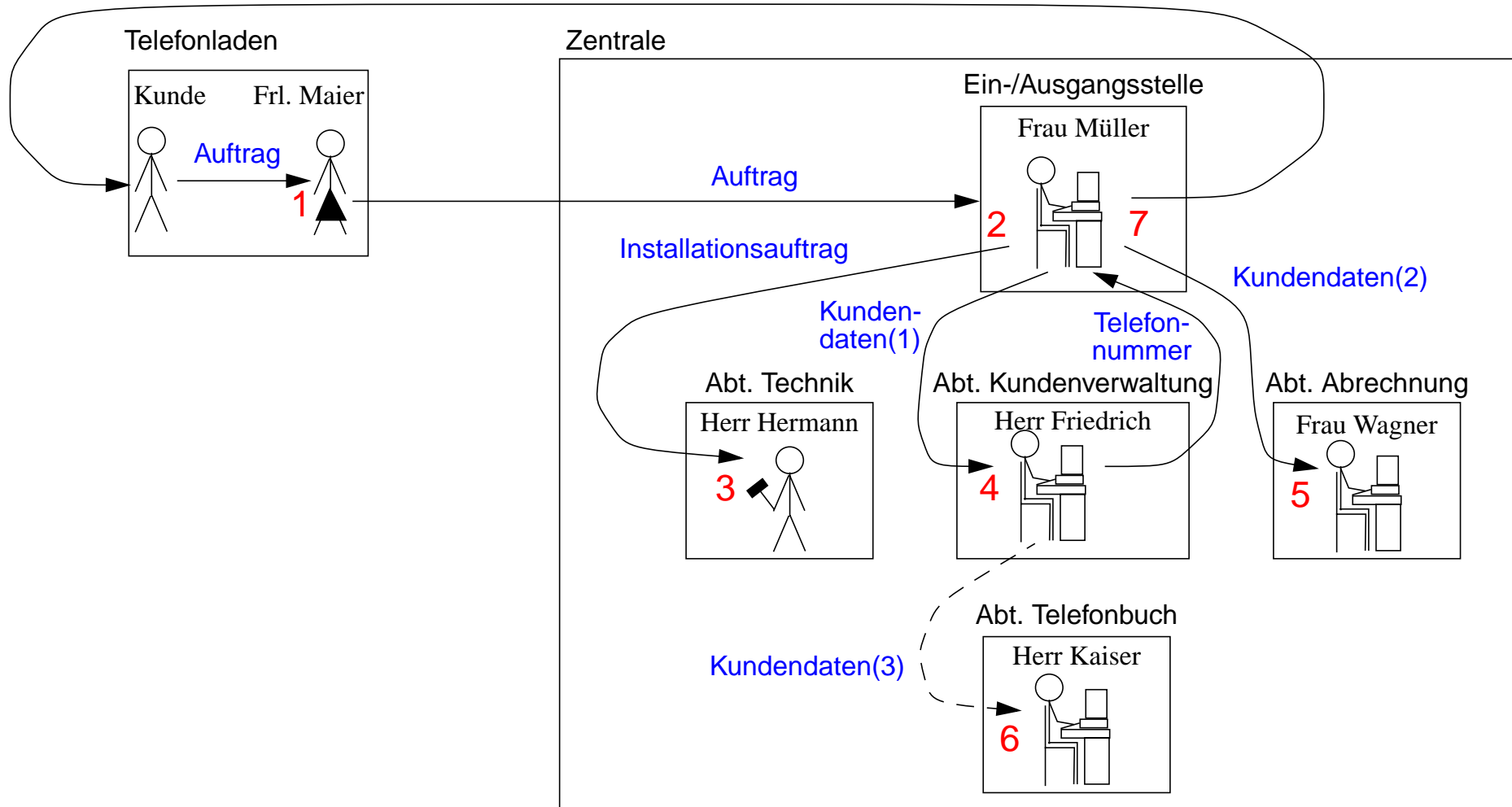
- Einsatz persistenter Warteschlangen
- Transaktionsgestützte Wf-Ausführung

- **Beziehungen zwischen den grundlegenden Begriffen**

# Workflows

- Beispiel: Telefonanschluß

Durchführungsbeleg



## Workflows (2)

### • WAS ?

- Kunden betreuen (1)
- eingehende Aufträge bearbeiten (2)
- Installation durchführen (3)
- Telefonnummer zuteilen / Kundendaten aufnehmen (4)
- Abrechnungsdaten erfassen (5)
- Kundendaten für Telefonbuch aufnehmen (6)
- Kundenmitteilung erstellen (7)

### • WER ?

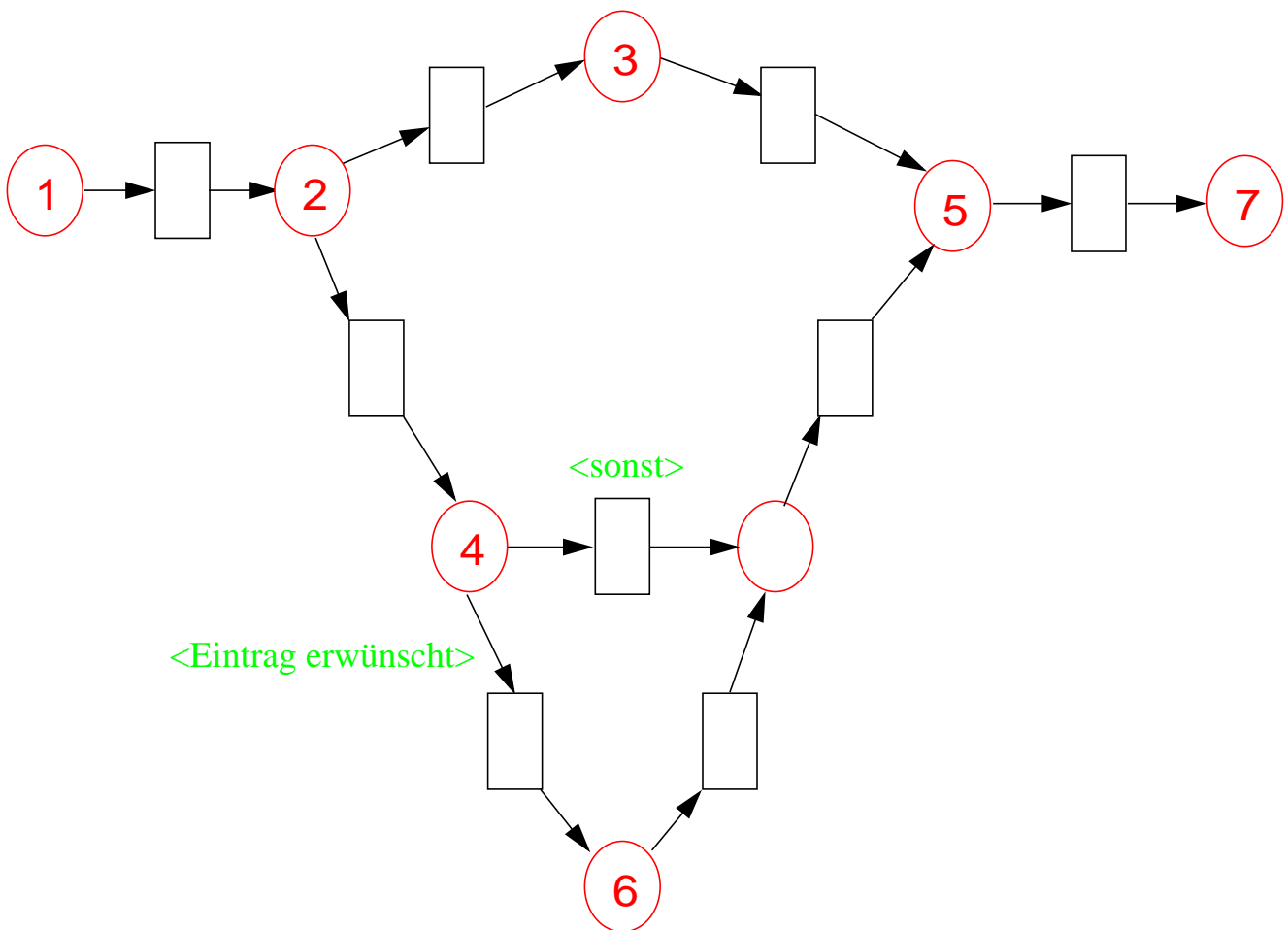
- Kundenbetreuerin (Frl. Maier) (1)
- Beschäftigter der Ein-/Ausgangsstelle (Frau Müller) (2)
- Techniker (Herr Hermann) (3)
- Sachbearbeiter "Neukunden" (Herr Friedrich) (4)
- Sachbearbeiter "Abrechnung" (Frau Wagner) (5)
- Sachbearbeiter "Telefonbuch" (Herr Kaiser) (6)
- Beschäftigter der Ein-/Ausgangsstelle (Frau Müller) (7)

### • WOMIT ?

- (1) → (2) Auftrag
- (2) → (3) Installationsauftrag
- (2) → (4) Kundendaten1
- (2) → (5) Kundendaten2
- (4) → (6) Kundendaten3

## Workflows (3)

- WANN ?



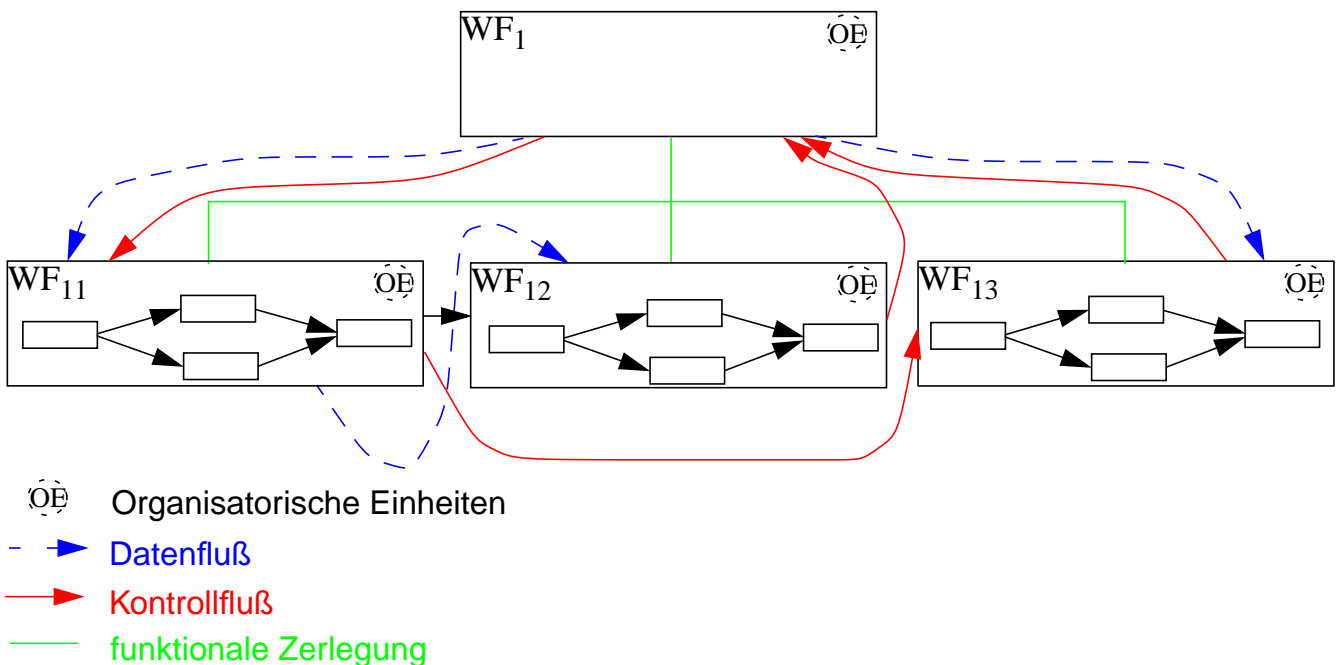
- Graph beschreibt zeitliche Abhängigkeiten zwischen einzelnen Tätigkeiten
- Parallelausführung verschiedener Tätigkeiten möglich
- optionale Ausführung von Tätigkeiten möglich
- Datenfluß i. allg. verschieden von Kontrollfluß

# Aspektorientierte Modellierung

- **Betrachtung des Prozesses aus verschiedenen Perspektiven (Aspekte nach Jablonski)**

- funktional was wird ausgeführt?
- verhaltensbezogen wann wird ausgeführt?
- datenbezogen was wird wo benötigt?
- organisatorisch wer führt aus?

- **Beziehung zwischen Aspekten**



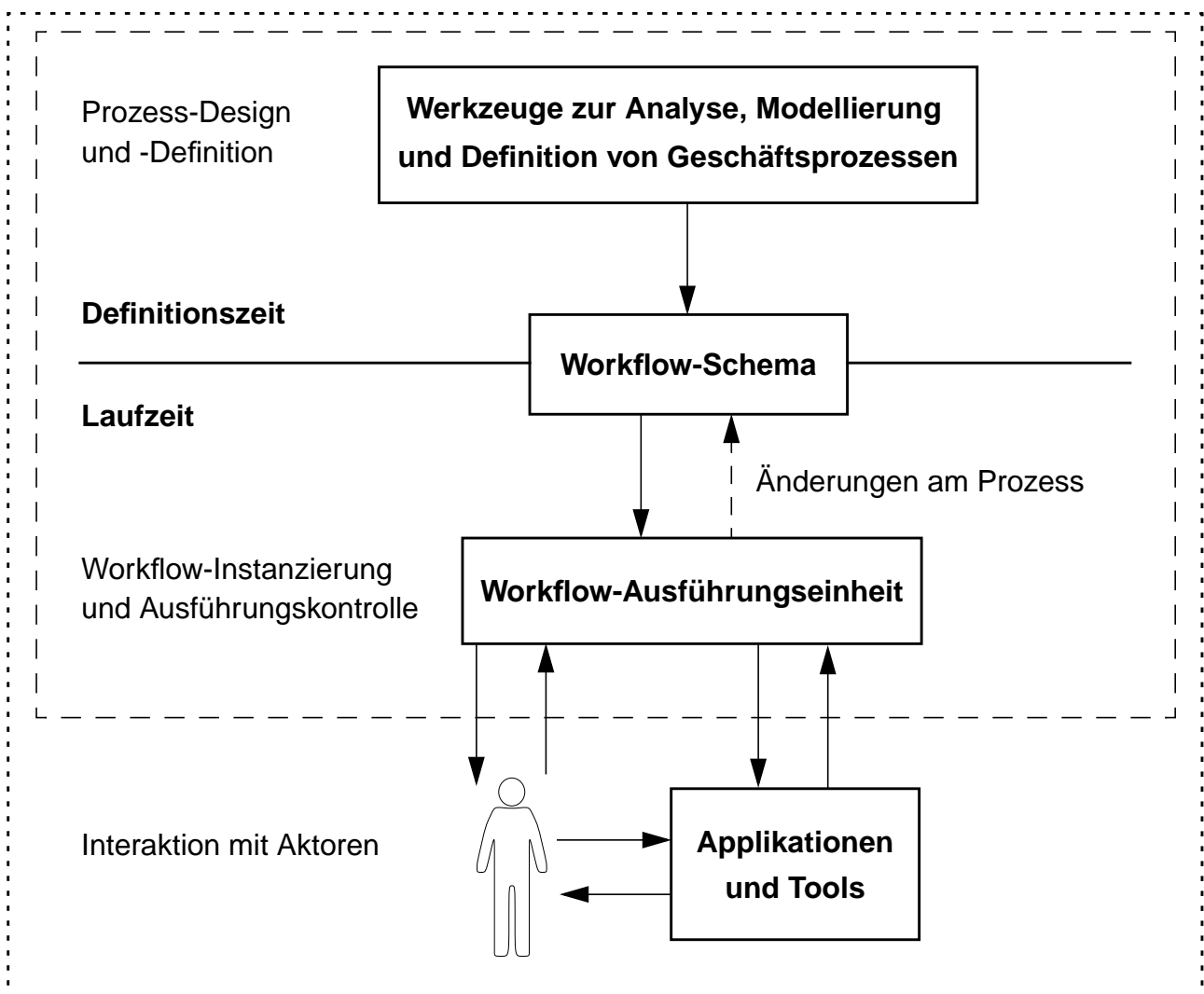
- Funktionaler Aspekt gibt grundlegende Struktur vor (Zerlegung in Sub-Workflows)
- Daten- und Kontrollfluß weitestgehend unabhängig voneinander modellierbar
- Elementare Sub-Workflows erfordern Zusammenführen der Aspekte

# Workflow-System als Gesamtsystem

- **Workflow-System (WfS)**

- besteht aus WfMS, Aktoren und allen zur Aktorenanbindung benötigten Komponenten
- WfMS besitzt Komponenten für die **Definitionszeit** und für die **Laufzeit**
- Wf-Ausführungseinheit wird auch als **Workflow-Engine** bezeichnet; sie kann durch mehrere Workflow-Engines realisiert sein

- **Funktionsbereiche eines WfS**



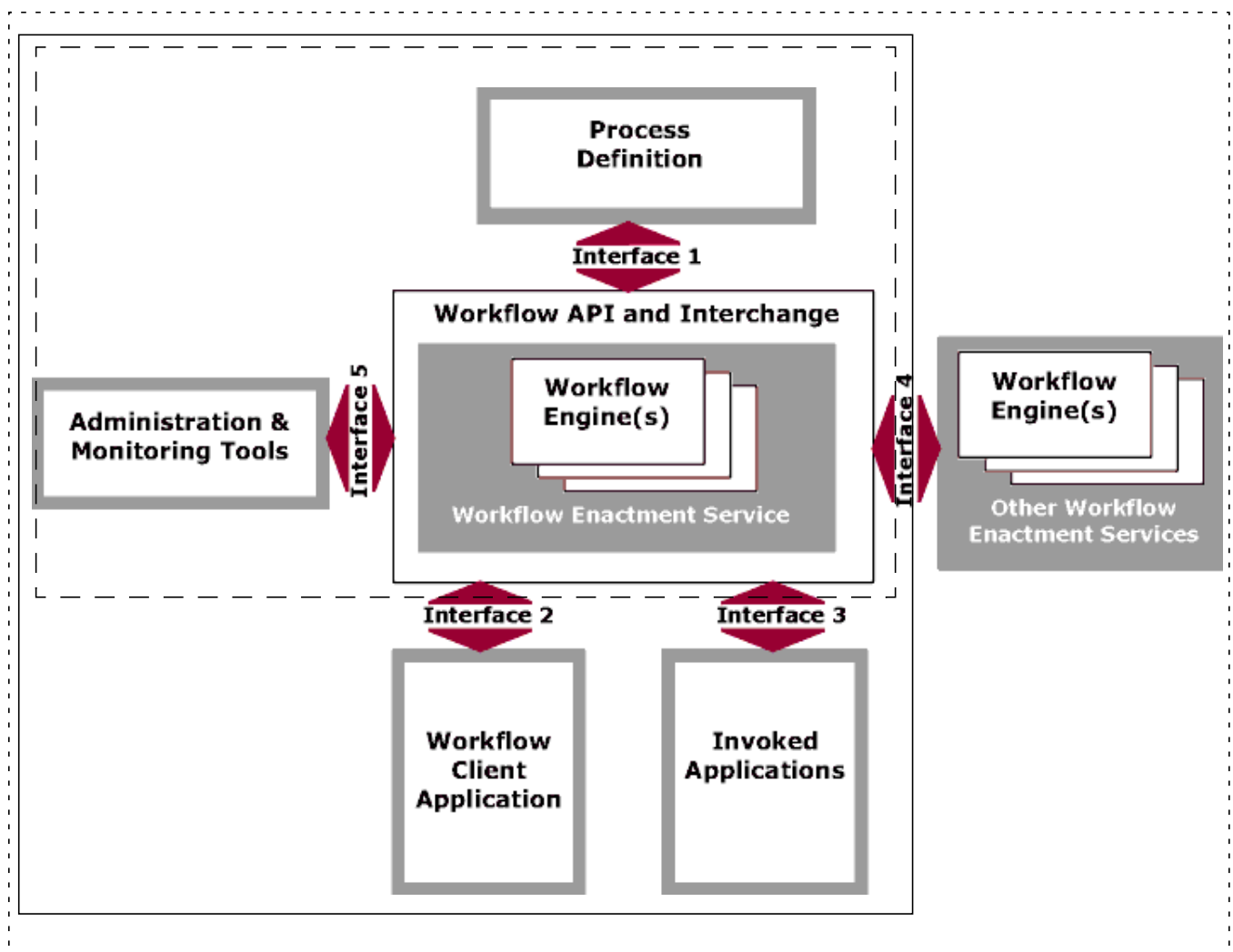
[ ] : **Workflow-Management-System (WfMS)**

[ ] : **Workflow-System (WfS)**

# Workflow-Referenzmodell der WfMC

- **Workflow Management Coalition (WfMC)**

- beschreibt eine generische WfS-Architektur
- definiert die Schnittstellen und Komponenten
- Kern des Systems: Wf-Ausführungseinheit (Wf Enactment Service)
- Schnittstellen 1 und 5 trennen unterschiedliche Funktionsschichten im WfMS
- Schnittstellen 2, 3 und 4 beschreiben die Grenzen des WfMS-Kontrollbereichs



□ : WfMS

□ : Verteiltes WfS, bei dem Sub-Workflows auf eigenständigen WfMS ausgeführt werden

□ : WfS

# Transaktionen - Erinnerung

- **Transaktionskonzept**

- führt ein neues Verarbeitungsparadigma ein
- ist Voraussetzung für die Abwicklung betrieblicher Anwendungen (*mission-critical applications*)
- erlaubt „Vertragsrecht“ in rechnergestützten IS zu implementieren

- **Welche Eigenschaften von Transaktionen sind zu garantieren? (ACID-Paradigma)**

- **Atomicity (Atomarität)**

- TA ist kleinste, nicht mehr weiter zerlegbare Einheit
- Entweder werden alle Änderungen der TA festgeschrieben oder gar keine („alles-oder-nichts“-Prinzip)

- **Consistency**

- TA hinterläßt einen konsistenten DB-Zustand, sonst wird sie komplett (siehe Atomarität) zurückgesetzt
- Zwischenzustände während der TA-Bearbeitung dürfen inkonsistent sein
- Endzustand muß die Integritätsbedingungen des DB-Modells erfüllen

- **Isolation**

- Nebenläufig (parallel, gleichzeitig) ausgeführte TA dürfen sich nicht gegenseitig beeinflussen
- Alle anderen parallel ausgeführten TA bzw. deren Effekte dürfen nicht sichtbar sein

- **Durability (Dauerhaftigkeit)**

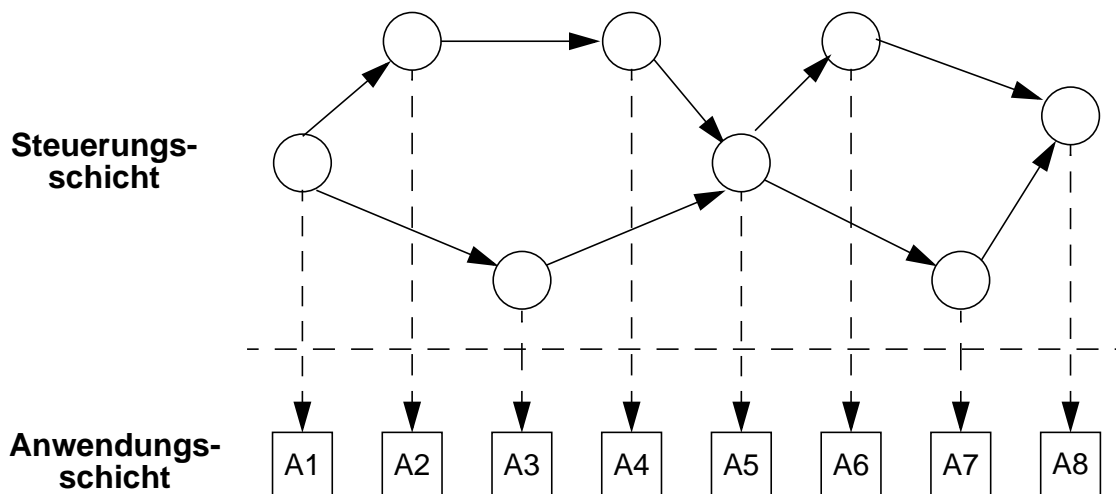
- Wirkung einer erfolgreich abgeschlossenen TA bleibt dauerhaft in der DB erhalten
- TA-Verwaltung muß sicherstellen, daß dies auch nach einem Systemfehler (HW- oder System-SW) gewährleistet ist
- Wirkungen einer erfolgreich abgeschlossenen TA kann nur durch eine sog. kompensierende TA aufgehoben werden



# Ausführung von Workflows

- **Eigenschaften von Workflows**

- verteilt, parallel, heterogen, hierarchisch organisiert, langlebig
- Aktionen auf getrennten und gemeinsamen Datenbereichen (Wf- und AW-spezifisch)
- Kooperation zwischen unabhängigen Komponenten (z. B. als Ressourcen-Mgr (RM) realisiert)



- **Sicherung durch TA-Konzept?**

- Workflows als globale Transaktionen? - nicht erreichbar, aber auch nicht wünschenswert
- ACID zumindest selektiv erforderlich
- kritische Kooperationen erfordern Transaktionsschutz durch spezielle RM-Protokolle

- **Konsequenzen für die WF-Ausführung**

- frühzeitige, aber trotzdem kontrollierte Freigabe von Änderungen auf gemeinsamen Daten
- Verwaltung der Ausführungshistorie und der Kontextdaten erlauben Unterbrechbarkeit sowie oft eine Wiederholbarkeit von Aktivitäten
- persistente Zwischenergebnisse und Nachrichten

# Anforderungen an ein WfMS

- **AW-bezogene Anforderungen**

- Einsatz als Middleware-Komponente
- Unterstützung verschiedenster Anwendungsgebiete (Flexibilität)
- Erweiterbarkeit des Wf-Modells
- Offenheit des Wf-Modells  
(neue Einsatzbereiche, Integration von Altsystemen)
- dynamische Anpaßbarkeit von Wfs an neue AW-Situationen  
(Versions- und Konfigurationsverwaltung)

- **Systembezogene Anforderungen**

- allgemeine Systemeigenschaften:  
Zuverlässigkeit, Wartbarkeit, Korrektheit, . . .
- Integration verteilter und heterogener Plattformen
- Portabilität der WfMS-Implementierung
- . . .

- **Zwei Arten von Datenhaltung erforderlich**

- Kontrolldaten (durch WfMS verwaltet)
- Produktionsdaten (durch DBMS oder Anwendung verwaltet)

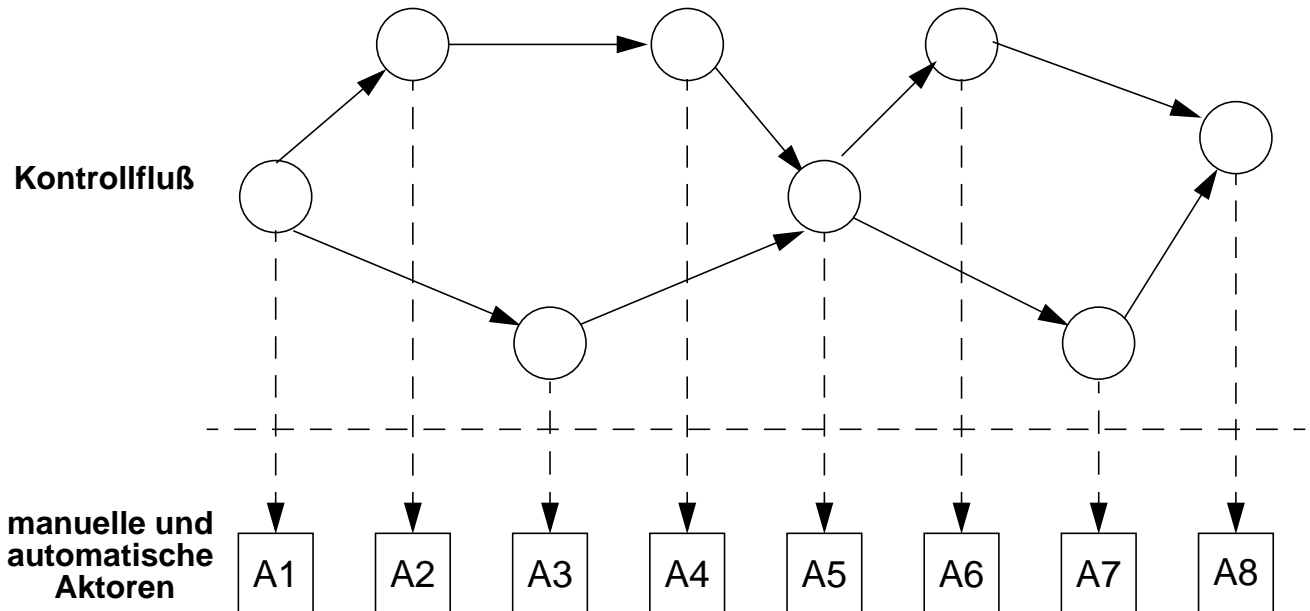
- **Systemdienste für die TA-orientierten Verarbeitung**

Gewährleistung der ACID-Eigenschaften für DB-Daten und auch für andere Betriebsmittel (BM wie persistente Warteschlangen, Nachrichten)

- ➔ Weitere Funktionalität erforderlich neben den beiden Arten der Datenverwaltung, geschützt durch TA-Mgr

## Anforderungen an ein WfMS (2)

- Zwei Arten von Datenhaltung erforderlich



- Anforderungen an die Wf-Ausführung

- Kosteneffektivität, Verlustminimierung im Fehlerfall
- einfache ACID-Transaktionssemantik ist nicht anwendbar
  - ➔ semantisch reichhaltigere Fehlerbehandlungsmodelle zwingend erforderlich
- frühzeitige Freigabe von Betriebsmitteln (v.a. Daten), Recovery im Fehlerfall
  - ➔ Semantik der Wf-Ausführung?

- Einsatz eines TP-Monitors (siehe Kap. 9)

- Scheduling in Wfs - Ausführung von AW-Funktionen:  
**TA-orientiertes BM-Scheduling**  
(kurzfristige Aspekte, Allokation von Ressourcen für Anforderungen)
- Verantwortung für Korrektheit der verteilten TA-Verarbeitung und für einheitliche Fehlersemantik (zusammen mit TA-Mgr)
  - ➔ alle TP-Monitor-Aufgaben fallen bei jeder Anforderung an:  
Authentifikation, Autorisierung, Prozeß-, Speicher-Allokation

# Funktionale Architektur von WfMS

- **WfMS benötigt persistente Daten**

- die die Struktur der Workflows spezifizieren (Workflow-Schemata)
- die den aktuellen Zustand der instanziierten Workflows verkörpern und deren Ausführungshistorie aufzeichnen (Laufzeitdaten)
- zur Haltung von Kontexten für AW-Funktionen (Aktoren)
- zur Speicherung von Beschreibungs- und Zustandsinformationen für die weiteren WfMS-Aspekte
  - ➔ Diese Daten erfordern den Einsatz eines DBS
- Aktualisierung der Kontrolldaten muß transaktionsgestützt erfolgen
  - ➔ Kontrolle durch einen WfMS-TA-Mgr

- **AW-Funktionen benötigen persistente Daten**

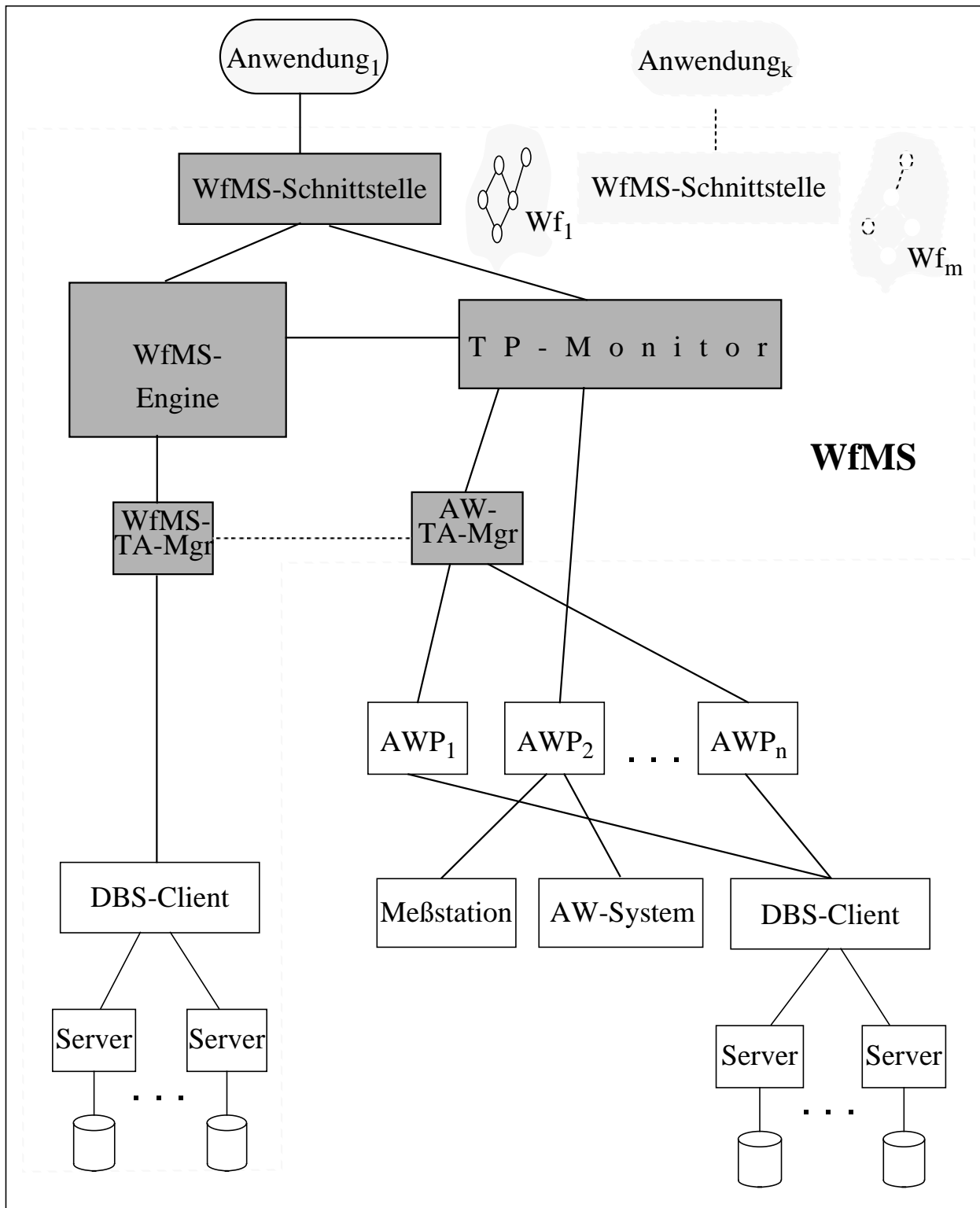
- alle Daten der AW-Miniwelt sind persistent und konsistent zu halten, was den Einsatz eines DBS impliziert (Schemakonsistenz)
- ihre Aktualisierung muß transaktionsgestützt erfolgen
  - ➔ Kontrolle durch einen AW-TA-Mgr

- **Synchronisation der Wf-Ausführung**

- die Zustände von WfMS- und AW-DBS müssen konsistent gehalten werden und dem Wf-Ablauf entsprechen
- jede "relevante" Datenaktualisierung erfordert spezielle Abstimmung (Zwei-Phasen-Commit-Protokoll)
  - ➔ Kooperation der TA-Mgr von WfMS und AW

## Funktionale Architektur von WfMS (2)

- Funktionale Architektur im Überblick



# Aspekte von Workflows

- **Essentielle Strukturelemente eines Workflows**

- **funktionaler Aspekt:**

Wie sieht die Struktur des auszuführenden Wf aus?

Beschreibung der funktionalen Einheiten, Zusammenhänge, Abhängigkeiten

- **verhaltensbezogener Aspekt:**

Wann sollen die Aktivitäten ausgeführt werden?

Kontrollfluß zur Abwicklung und Steuerung des Wf

- **datenbezogener Aspekt:**

Wie sind die am Wf beteiligten Daten definiert und wo werden sie gebraucht?

Datenfluß von Kontroll- und Produktionsdaten (Bereitstellung und Sicherung)

- **organisatorischer Aspekt:**

Wer führt die Aktivitäten aus?

Zuordnung von organisatorischen Einheiten (z.B. Personen) zu Rollen

- **operationaler Aspekt:**

Wie werden die Akteure in einen Wf eingebunden?

- **Sicherheitsaspekt:**

Wer ist zur Wf-Ausführung berechtigt? (Authentifikation und Autorisierung)

➔ Sie konstituieren notwendige Aspekte einer allgemeinen WfMS-Architektur

➔ Realisierung durch WfMS-Engine

- **Technische Aspekte der Wf-Ausführung**

- **transaktionaler Aspekt:**

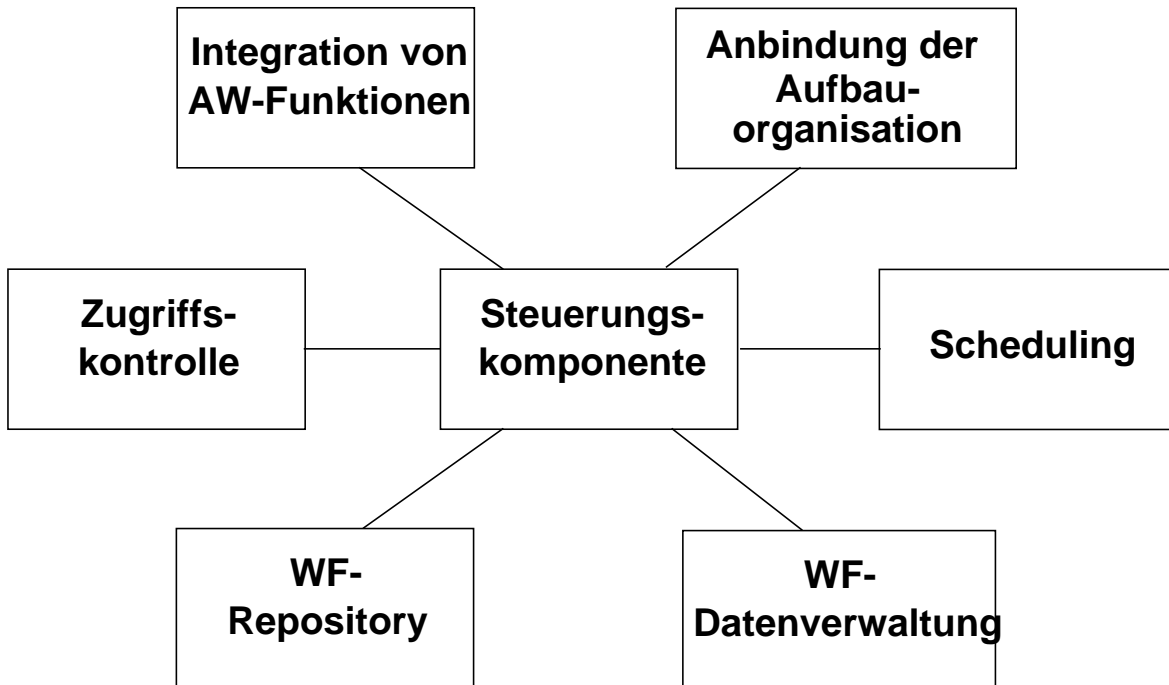
Wie wird erreicht, daß Aktivitäten und logisch zusammengehörige Folgen von Aktivitäten atomar und persistent ausgeführt werden?

- **historischer Aspekt:**

Welche Information existiert zu bereits abgeschlossenen Workflow-Ausführungen? (Nachweis, Optimierung)

## Funktionale Architektur von WfMS (3)

- **Funktionen der WfMS-Engine**



- **Integration von manuellen und automatischen Aktivitäten**

(operationaler Aspekt)

- Schnittstellendefinitionen zu externen Aktoren
- Fehlerüberwachung und -behandlung
- . . .

- **Anbindung der Aufbauorganisation**

(organisatorischer Aspekt)

- Beschreibung organisatorischer Zuordnungen
- dynamische Zuordnung von Rollen zu „organisatorischen Elementen“ (z.B. Personen)
- Einhaltung von „Business Rules“

# Funktionale Architektur von WfMS (4)

- **Scheduling von Wfs**

(verhaltensbezogener Aspekt)

- Auswahl der Wfs
- Aktivierung des Kontrollflusses zwischen Wfs
- Zeitüberwachung, Deaktivierung/Aktivierung

- **Wf-Datenverwaltung**

(datenbezogener Aspekt)

- Datenversorgung der Wf-Instanzen
- Verwaltung persistenter Nachrichten (Wf-Ergebnisdaten / -Parameter)
- Speicherung ggf. persistenter Kontexte
- Sicherung der Ablaufhistorie (historischer Aspekt)

- **Wf-Repository**

(funktionaler Aspekt)

- Verwaltung der Wf-Schemata und Integritätsbedingungen
- Definition der Abhängigkeiten zwischen Wf-Schritte
- wenn möglich/erforderlich: (transaktionaler Aspekt)
  - Definition atomarer Ablaufeinheiten
  - Definition der Kompensationssphären und -einheiten

- **Zugriffskontrolle**

(Sicherheitsaspekt)

- Authentifikation und Autorisierung
- spezifische Sicherheitsanforderungen der Wf-Ausführung
- Kontrolle des Rechtetransfers

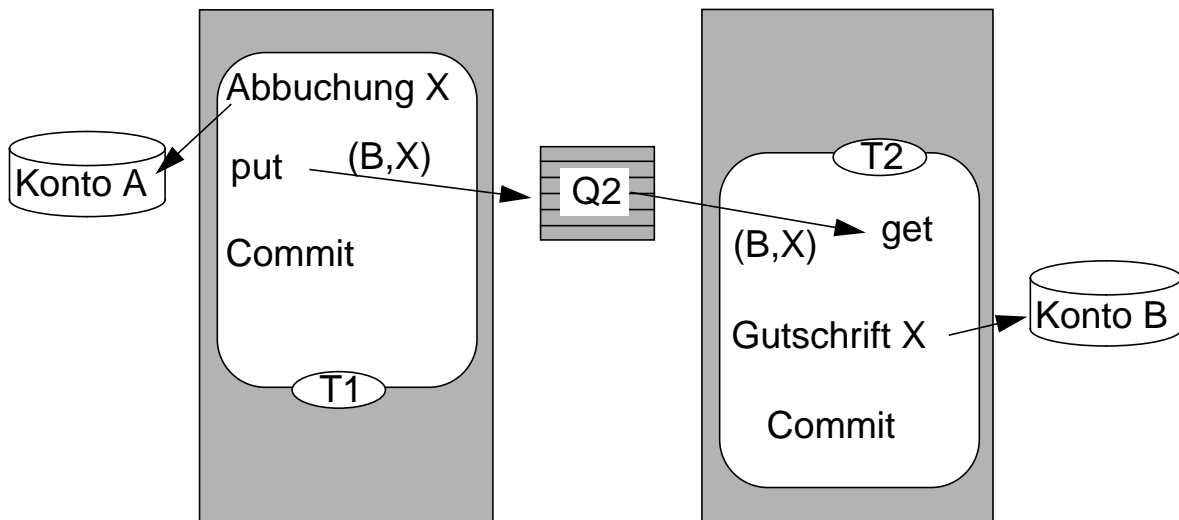


# Transaktionen und Workflows

- **Workflow als eine globale Transaktion T**

- Einfluß auf Durchsatz und Leistungsverhalten des Systems
  - ACID-Eigenschaft, daß alle Operationen zur gleichen Zeit erfolgreich beendet werden, aus AW-Sicht oft nicht erforderlich
- ➔ Garantie ausreichend, daß nach erfolgreicher Beendigung einiger Operationen die restlichen „irgendwann“ erfolgreich ausgeführt werden

- **Einsatz von „Persistenten Warteschlangen“ (recoverable messaging)**



Zerlegung der globalen Transaktion  
"Buchung von Konto zu Konto"

- **Semantisch reichhaltigere Fehlermodelle**

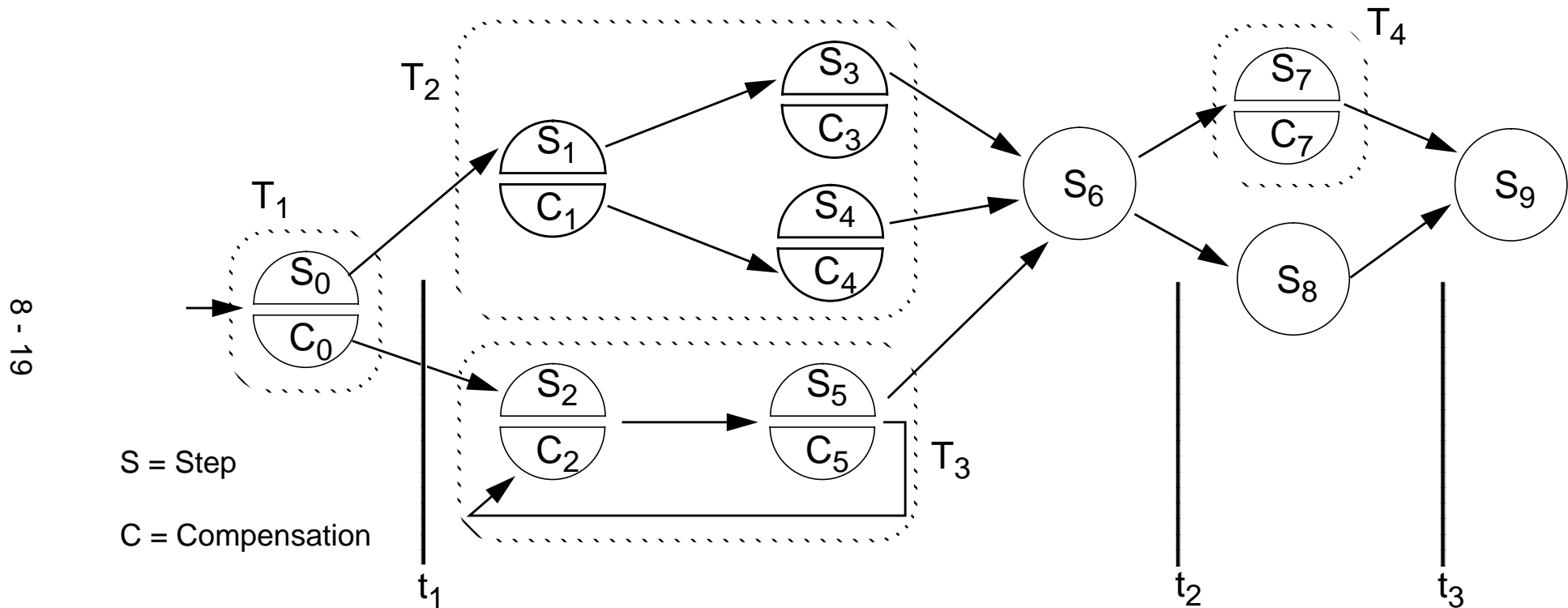
- Kombination von ungeschützten Aktionen und Transaktionen
- Wiederanlauf: persistente Warteschlangen erlauben Wiederholung von TA und Vorwärts-Recovery
- Scheitern des WF:  
Rücksetzen und Kompensation von TA-Ergebnissen

## Transaktionen und Workflows (2)

- **AW-orientierte Zerlegung von Transaktion T**
  - in  $T_1, \dots, T_n$
  - mit persistenten Warteschlangen  $Q_1, \dots, Q_n$
  - lineare Reihenfolge nicht zwingend
- **Einsatz von persistenten Warteschlangen**
  - zeitliche Unabhängigkeit der Funktionsausführung
    - ↳ höhere Unabhängigkeit der Komponenten, höhere Verfügbarkeit des Gesamtsystems
- **Ausführung der Transaktion T**
  - $T_i$  schreibt in  $Q_{i+1}$ ,  $T_{i+1}$  liest aus  $Q_{i+1}$
  - Verkettung der  $T_i$  durch spezielles Protokoll (XA-Protokoll von X/Open)
  - jede  $T_i$  macht unabhängig Commit
  - es soll sichergestellt werden, daß alle Transaktionen  $T_i$  in T erfolgreich beendet werden
    - ↳ Voraussetzung: alle aktualisierten Ressourcen (auch Nachrichten) sind wiederherstellbar
- **Scheitern der Transaktion T**
  - zunächst Vorwärts-Recovery bei Crash
  - Weiterführung scheitert auch bei wiederholten Versuchen
  - Rücksetzen offener  $T_i$
  - Kompensation erfolgreich beendeter  $T_i$

## Transaktionen und Workflows (3)

- Transaktionsgestütztes Workflow-Szenario

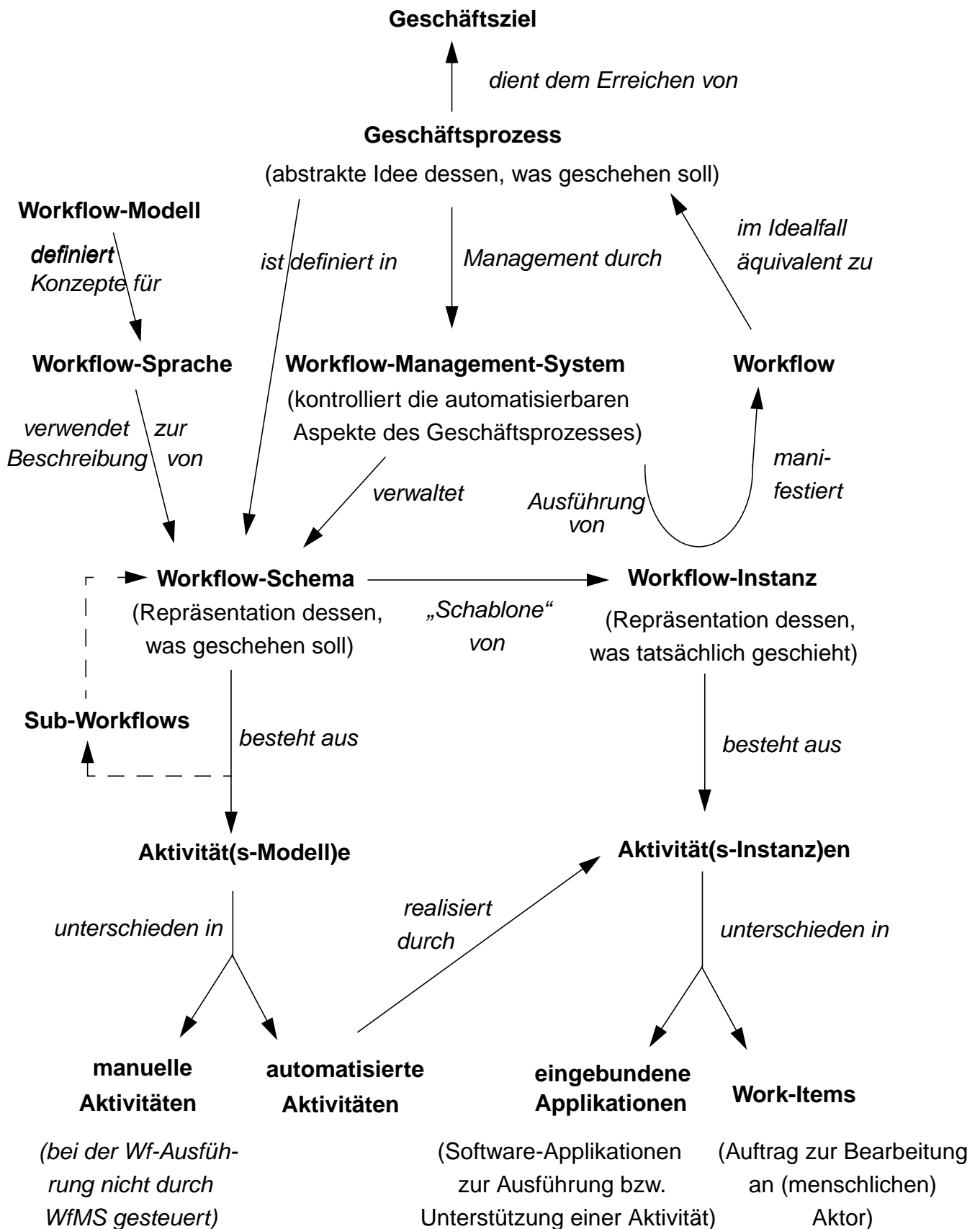


### ↳ Recovery-Aktionen bei

- Rücksetzen von  $T_2$
- Crash bei  $t_1, t_2, t_3$
- Einsatz von persistenten Warteschlangen

# Zusammenfassung - Workflow-System

- Beziehungen zwischen den grundlegenden Begriffen (nach WfMC)



## Zusammenfassung - Aspekte von Workflows

| <b>Aspekt</b>     | <b>Bedeutung</b>  |
|-------------------|---|
| funktional        | Wie sieht die Struktur des auszuführenden Workflows aus?  |
| verhaltensbezogen | Wann sollen die Aktivitäten ausgeführt werden?  |
| operational       | Wie werden die Aktoren in einen Wf eingebunden implementiert?   |
| datenbezogen      | Wie sind die am Workflow beteiligten Daten definiert und wo werden sie gebraucht?   |
| organisatorisch   | Wer führt die Aktivitäten aus?  |
| Sicherheit        | Wer darf die Aktivitäten ausführen?   |
| historisch        | Welche Information existiert zu bereits abgeschlossenen Workflow-Ausführungen   |
| transaktional     | Wie wird erreicht, daß Aktivitäten und logisch zusammengehörige Folgen von Aktivitäten atomar und persistent ausgeführt werden? |