

Vorlesung
***Methodische Grundlagen des
Software-Engineering***
im Sommersemester 2014

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Teil 1.3: Workflow-Management-Systeme

v. 24.04.2014

1.3 Workflow-Management-Systeme

[inkl. Beiträge von Jutta Mülle und Dr. Silvia von Stackelberg,
(LS Prof. Böhm, Karlsruher Institut für Technologie) und
Professor Dr. Frank Leyman (Universität Stuttgart)]

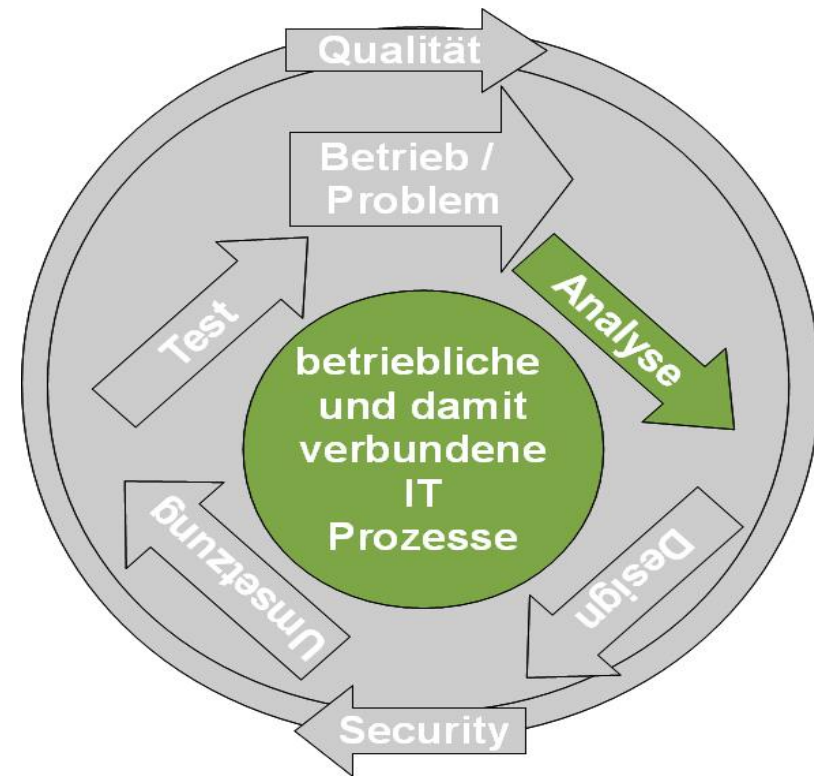
Literatur:

A. Gadatsch: Grundkurs Geschäftsprozess-Management. Springer Vieweg Verlag,
2013, 7. Auflage. Unibibliothek **E-Book**:

<http://www.ub.tu-dortmund.de/katalog/titel/1360251>

- Kapitel 4

- **Geschäftsprozessmodellierung**
 - Grundlagen Geschäftsprozesse
 - Ereignisgesteuerte Prozessketten (EPKs)
 - Einführung in die BPMN 2.0
 - **Workflow-Management-Systeme**
 - Workflow-Automatisierung
- Process Mining
- Modellbasierte Entwicklung sicherer Software



Vorige Abschnitte: Modellierung und Analyse von Geschäftsprozessen / Workflows.

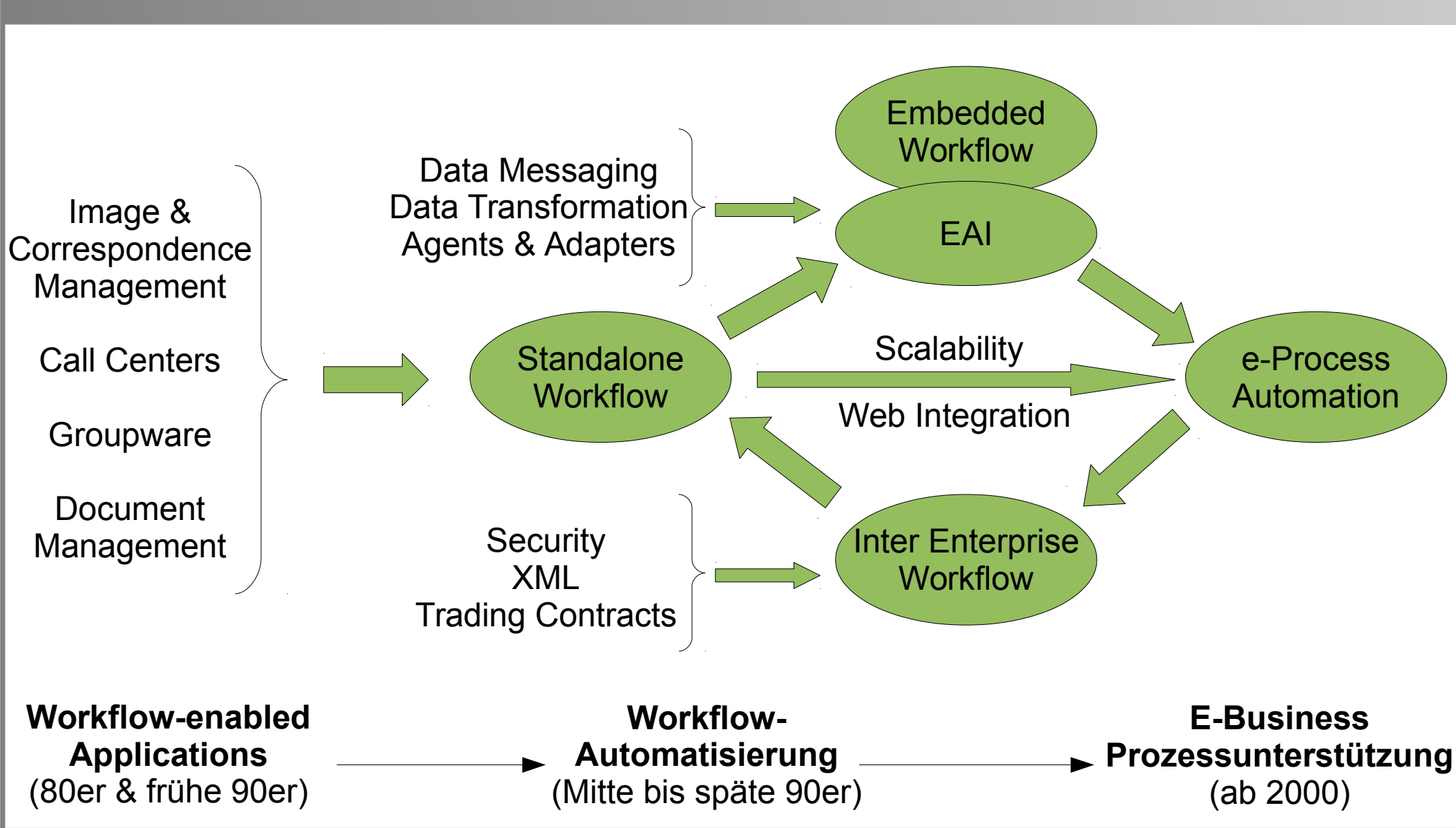
Hier: Wie kann man solche Modelle nicht nur zur Dokumentation und Analyse benutzen, sondern im Kontext der tatsächlichen **Ausführung der Workflows („Workflow Management“)** ?

Was sind praktische Systeme, die dies unterstützen (**Workflow-Management-Systeme**) ?

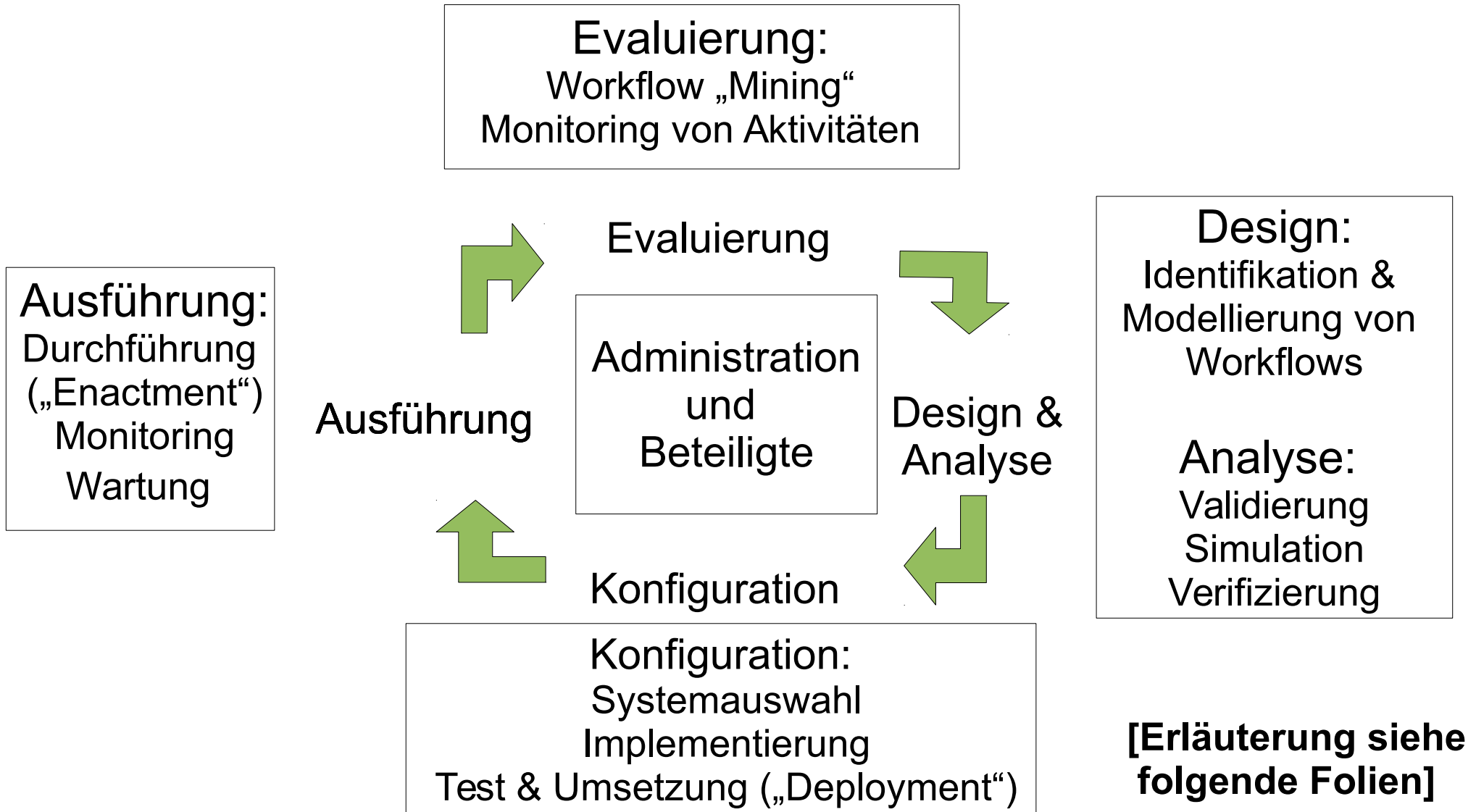
Darauf aufbauend (im nächsten Abschnitt):
Workflow-Automatisierung (im Kontext der Business Process Execution Language (BPEL)).

- **Lebenszyklus** von Workflows;
Überblick Modellierung und Ausführung.
- **Workflow-Aspekte** (Begriffe, Architekturansätze);
Integration von **Workflow-Management-Systemen** (WfMS) in
Anwendungssysteme.
- Referenzmodell der **Workflow Management Coalition** (WfMC):
 - Überblick
 - Komponenten und Schnittstellen
 - Anmerkungen

Historische Entwicklung von WfMS



Lebenszyklus von Workflows: Überblick



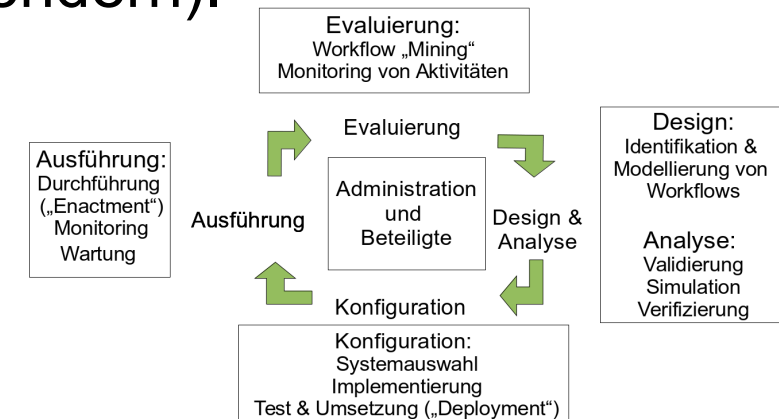
Quelle: M. Weske 2007

Design:

- Identifikation, Abgrenzung, Spezifikation von Zielen.
- Modellierung der Workflow-Schemata typischerweise in Graph-Notation (Kommunikationsbasis mit Anwendern).

Analyse:

- Validierung (Feedback der Anwender).
- Simulation (Prozessverhalten, z.B. Dauer, Lastverteilung).
- Verifizierung (z.B. Erkennung von Deadlocks).

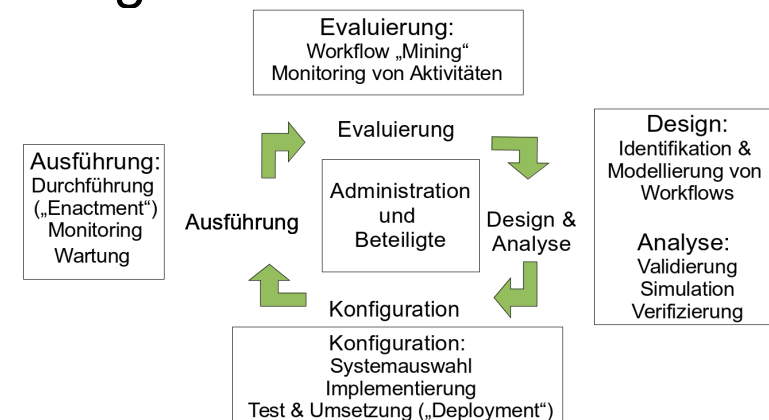


Konfiguration:

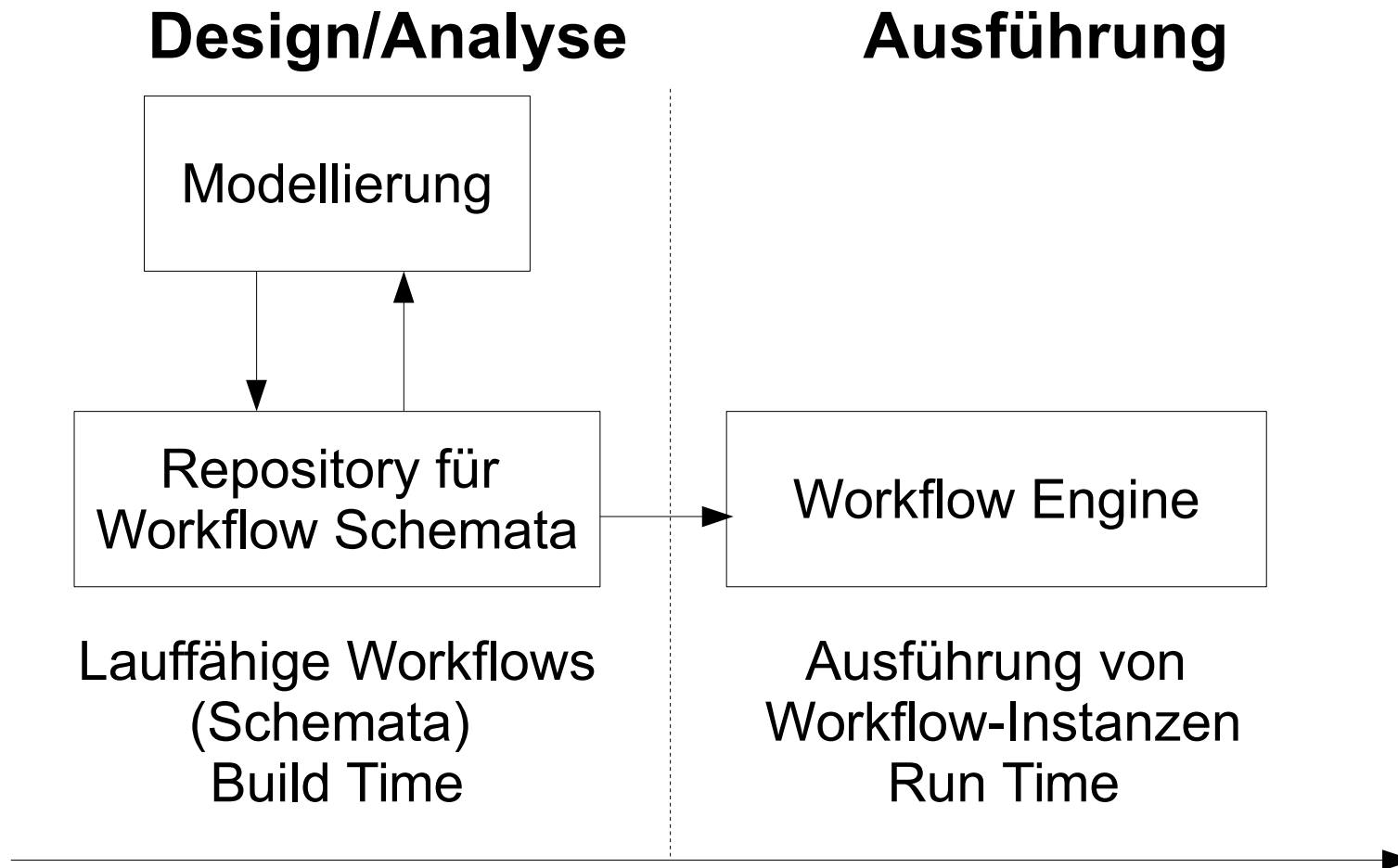
- Systemauswahl (z.B. WfMS), Spezifikation von **technischen Informationen** und **Benutzerinteraktionen**, Integration existierender Systeme.
- Implementierung.
- Test und Umsetzung („**Deployment**“).

Ausführung:

- Durchführung („**Enactment**“): korrekte Ausführung („**Orchestration**“) von Workflowinstanzen gemäß spezifizierter Bedingungen (zeitl. Ablauf etc.).
- **Monitoring** (z.B. Statusinformationen von aktiven Instanzen, Events in Log-Files).
- Betrieb und Wartung.

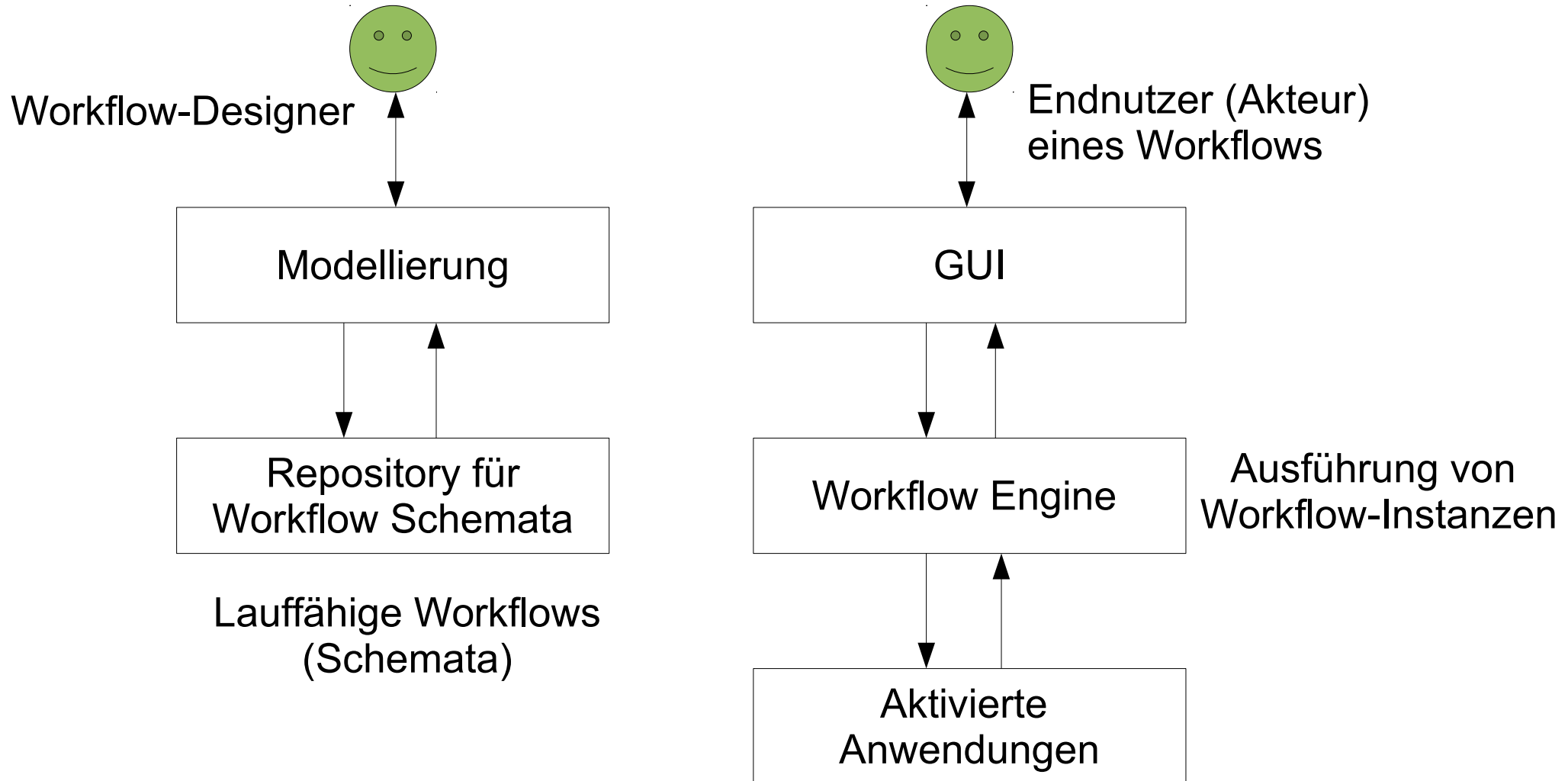


Modellierung vs. Ausführung: Technische Infrastruktur



Quelle: M. Weske, 2007

Modellierung vs. Ausführung: Workflow-Designer vs. Akteur



Quelle: M. Weske

- **Lebenszyklus** von Workflows;
Überblick Modellierung und Ausführung.
- **Workflow-Aspekte** (Begriffe, Architekturansätze);
Integration von **Workflow-Management-Systemen** (WfMS) in
Anwendungssysteme.
- Referenzmodell der **Workflow Management Coalition** (WfMC):
 - Überblick
 - Komponenten und Schnittstellen
 - Anmerkungen

Fundament eines **WfMS**:

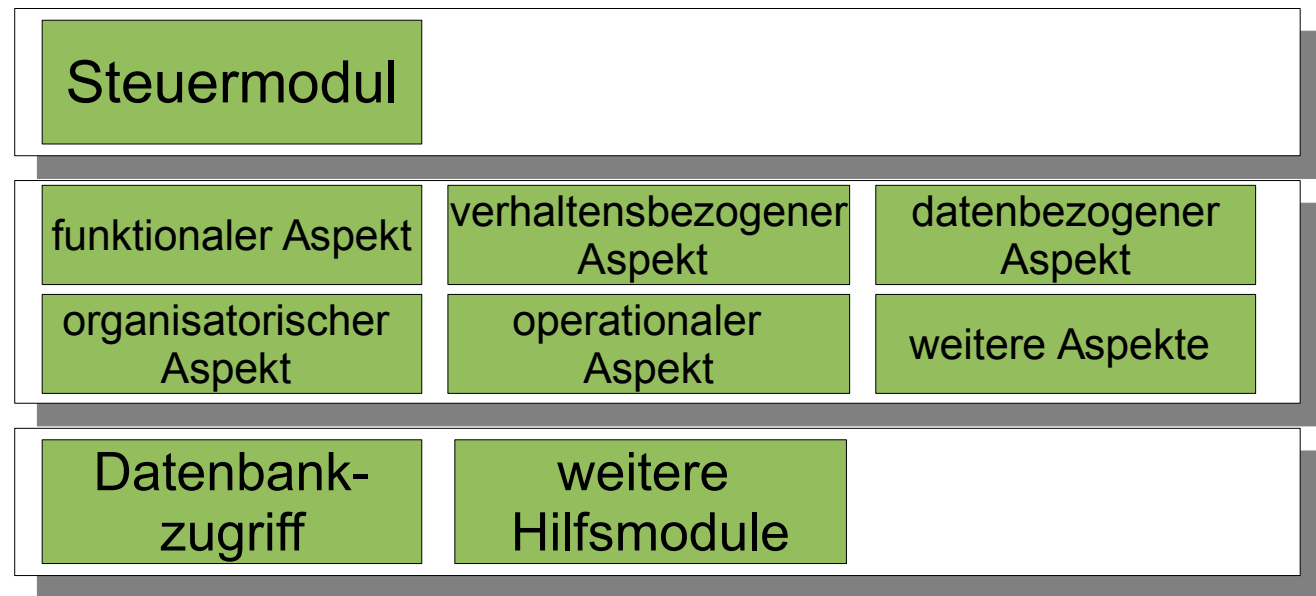
- **Funktionaler Aspekt:** funktionale Einheiten, d.h. Workflow-Struktur mit elementaren und zusammengesetzten Arbeitsschritten (Unter-Workflows).
- **Verhaltensbezogener Aspekt:** Kontrollfluss zwi. elementaren und zusammengesetzten Arbeitsschritten eines Workflows unter Berücksichtigung von kausalen und temporalen Zusammenhängen.
- **Datenbezogener Aspekt:** Datenfluss eines Workflows.
- **Organisations-Aspekt:** organisations-bezogene Inhalte mit Definition von Organisationsstrukturen und deren Population, sowie Festlegung, wer Operationen eines Workflows ausführen kann / darf.
- **Operationaler Aspekt:** Einbindung von Programmen in elementare Arbeitsschritte.

Erweiterungen oft notwendig (z.B. Sicherheitsaspekt).

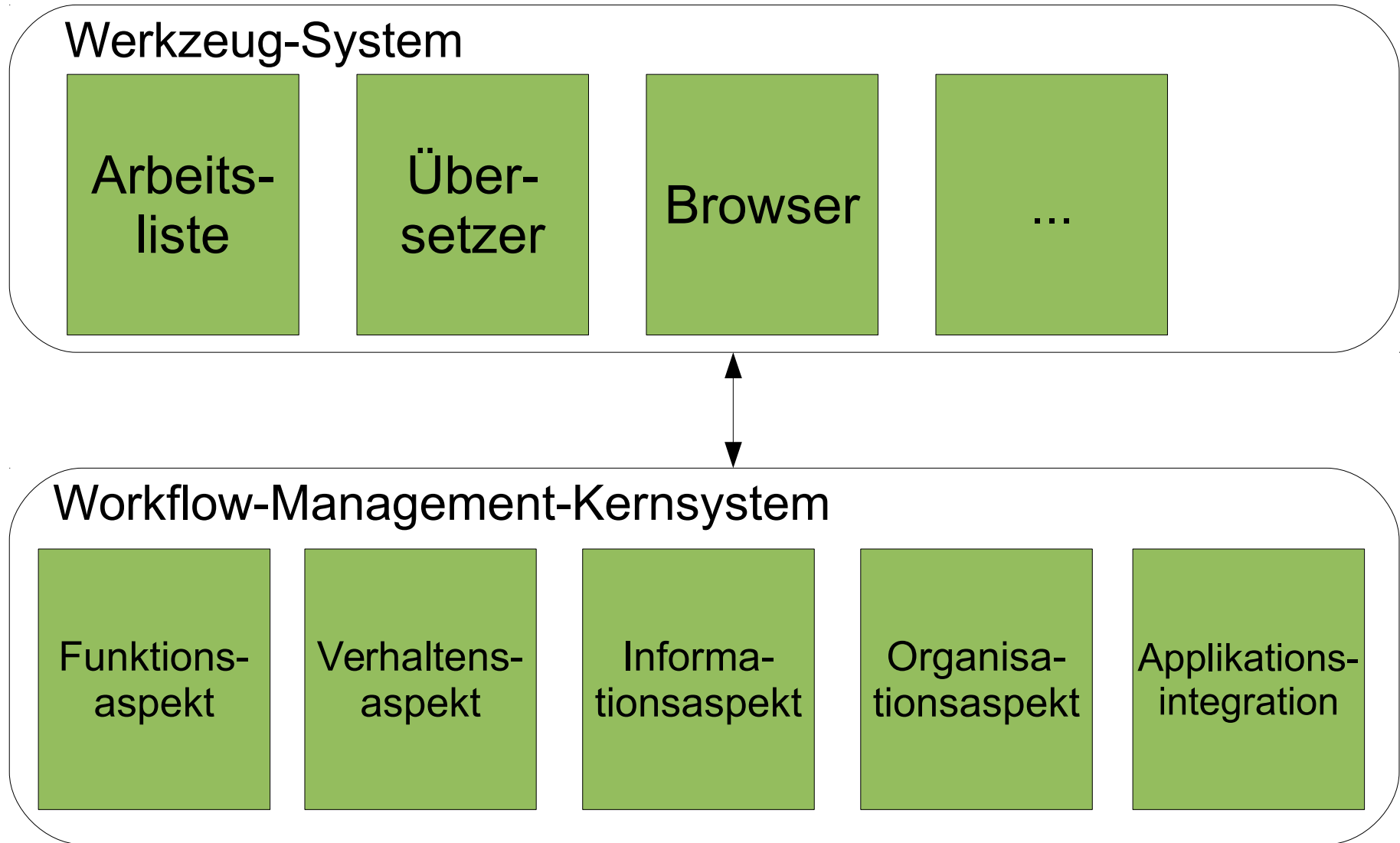
Ziel: Beschreibung des **implementierungs-unabhängigen** Teils der Architektur eines WfMSs.

Spezifikation wesentlicher Module oder Komponenten eines WfMSs.
Insbesondere:

- **Module** für o.g. Aspekte eines WfMSs.
- **Steuermodul**
- Hilfsmodule für **allgemeine Dienste**



Funktionale Komponenten eines WfMSs



- **Offenheit** der Systemimplementierung auf verschiedenen Ebenen:
 - Funktionale **Erweiterbarkeit** und **Konfigurierbarkeit**.
 - Unterstützung von **verteilten heterogenen Einsatzumgebungen**.
- **Zuverlässigkeit** des Systems:
 - **Verfügbarkeitsgarantien** für Workflow-Instanzen.
 - **Transparenz** von Systemfehlern.
- **Sicherheitsanforderungen**.
- **Analysierbarkeit** von Workflow-Management-Systemen.
- **Skalierbarkeit**.
- Berücksichtigung **organisatorischer** Gegebenheiten.

Diskussion: Workflow-Aspekt Sicherheit

Sicherheit als **Workflow-Aspekt** wird **orthogonal** zu klassischen Workflow-Aspekten gesehen.

Was könnte damit gemeint sein? Welche **Konsequenzen** hat das?

Diskussion: Workflow-Aspekt Sicherheit

Sicherheit als **Workflow-Aspekt** wird **orthogonal** zu klassischen Workflow-Aspekten gesehen.

Was könnte damit gemeint sein? Welche **Konsequenzen** hat das?

Antwort:

Sicherheit ist orthogonal: durchdringt andere Aspekte. Sicherheit nicht getrennt betrachtbar.

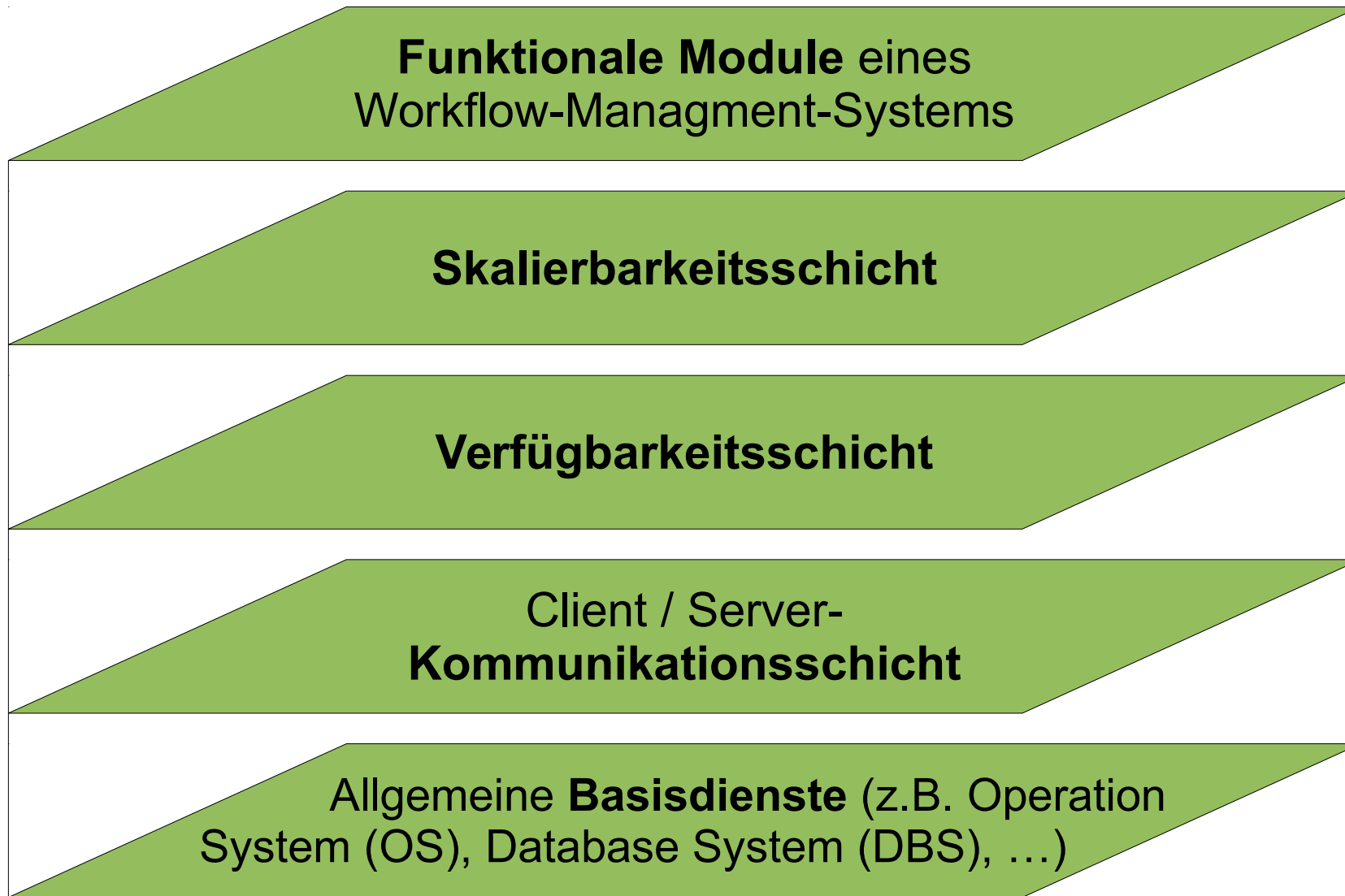
Konsequenz: Einführen von Sicherheitsaspekten in Workflow-Betrachtung **komplex** und wird im Kontext von Workflows leider oft vernachlässigt.

Schichtenarchitektur eines WfMS

Methodische Grundlagen
des Software-Engineering
SS 2014

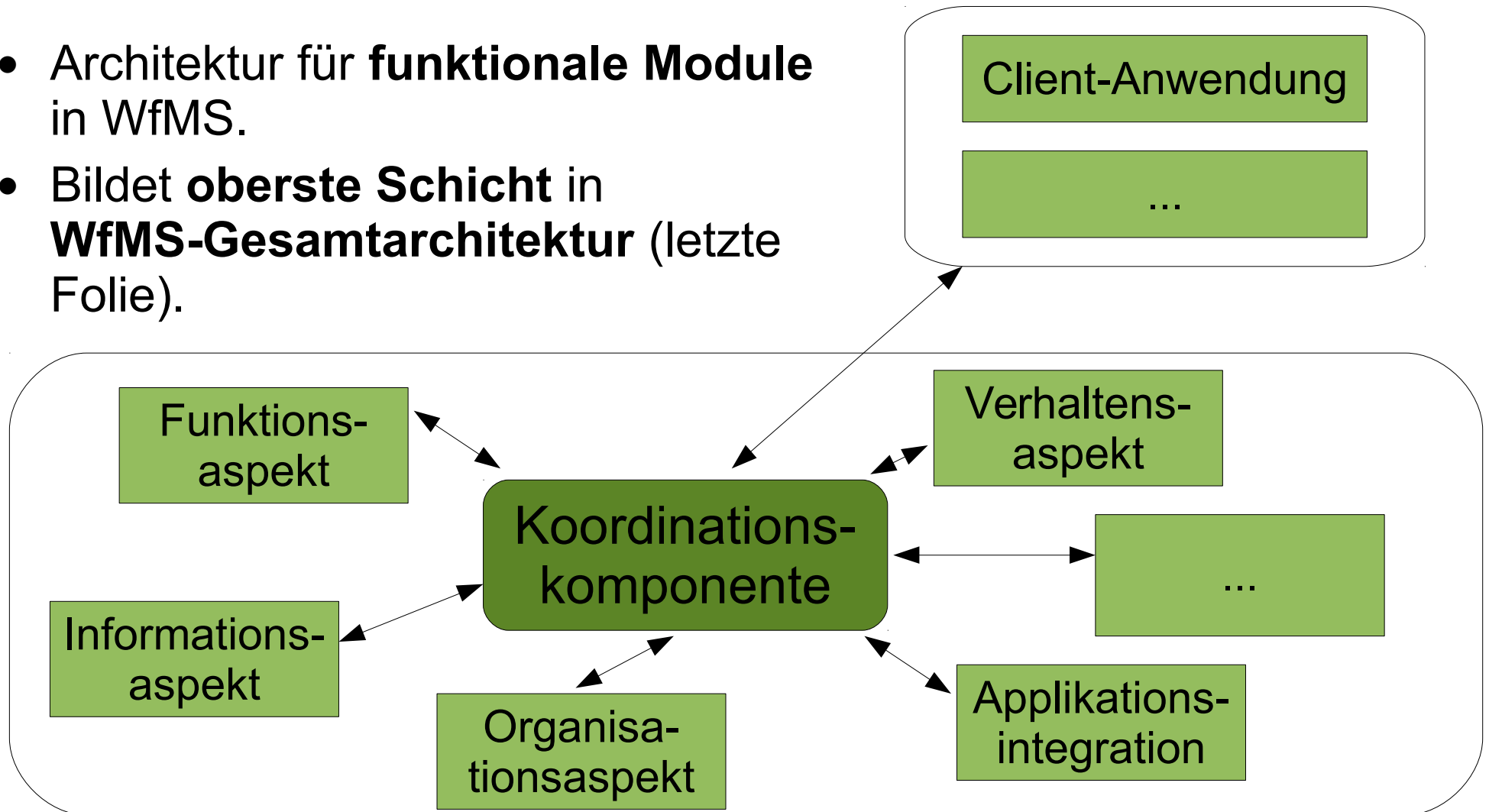


LEHRSTUHL 14
SOFTWARE ENGINEERING

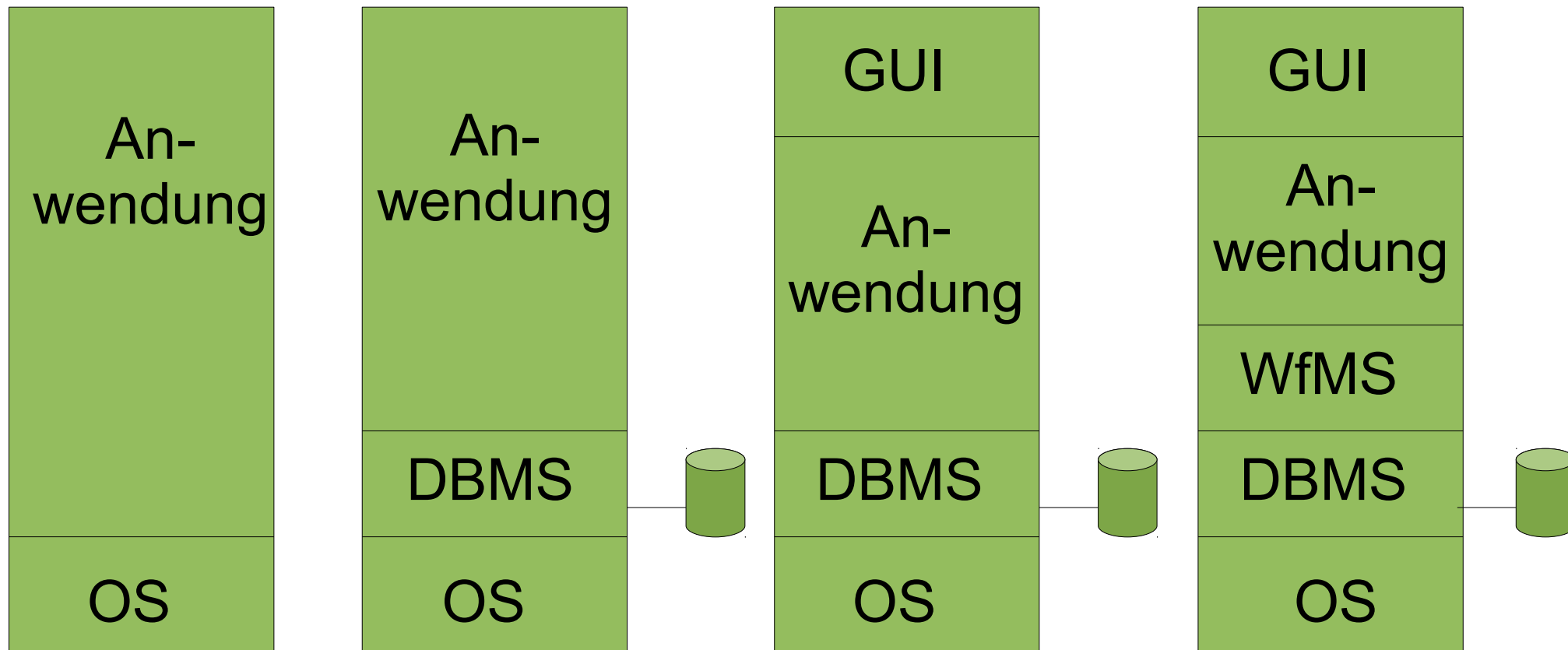


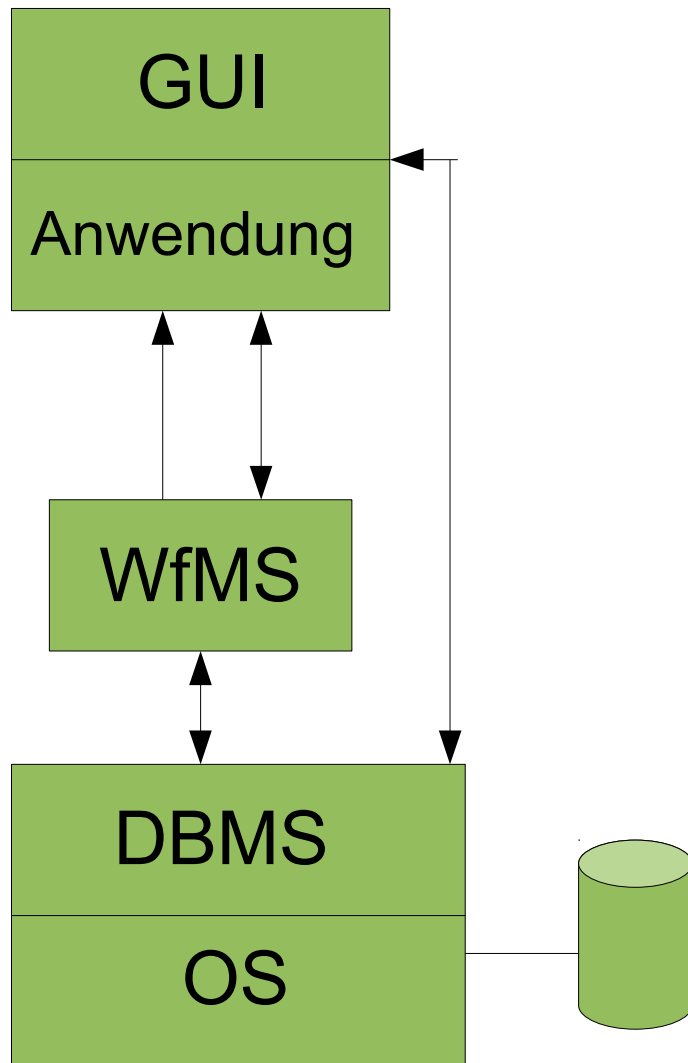
Schichtenarchitektur eines WfMS: Funktionale Module

- Architektur für **funktionale Module** in WfMS.
- Bildet **oberste Schicht** in **WfMS-Gesamtarchitektur** (letzte Folie).



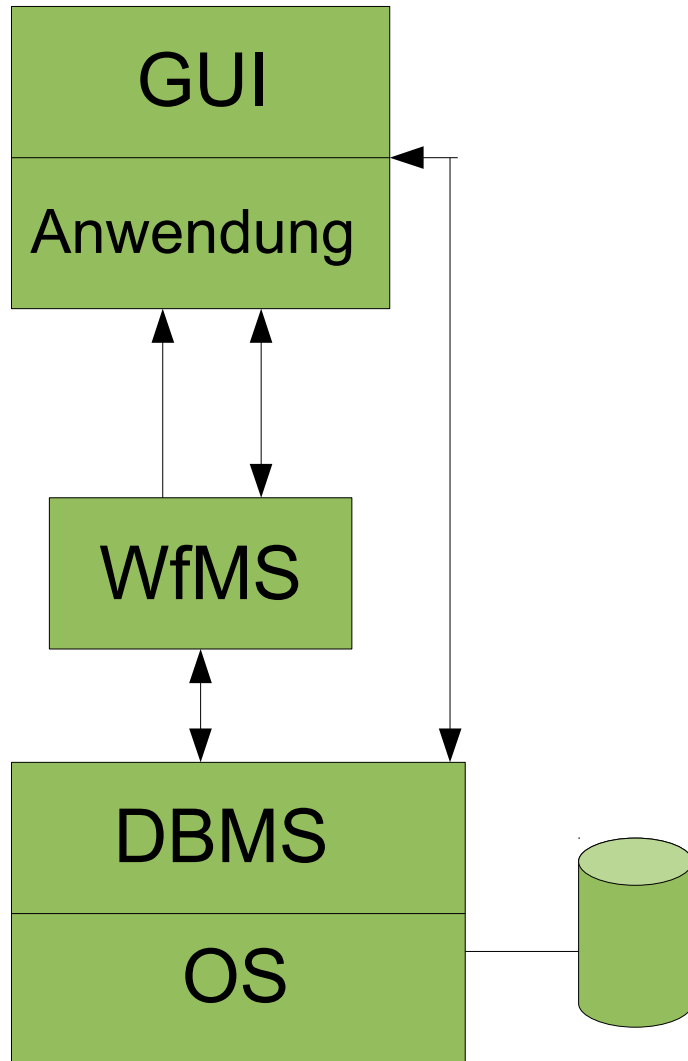
Evolution von Anwendungssystemen





Explizite Trennung von Workflow-Management und Anwendung:

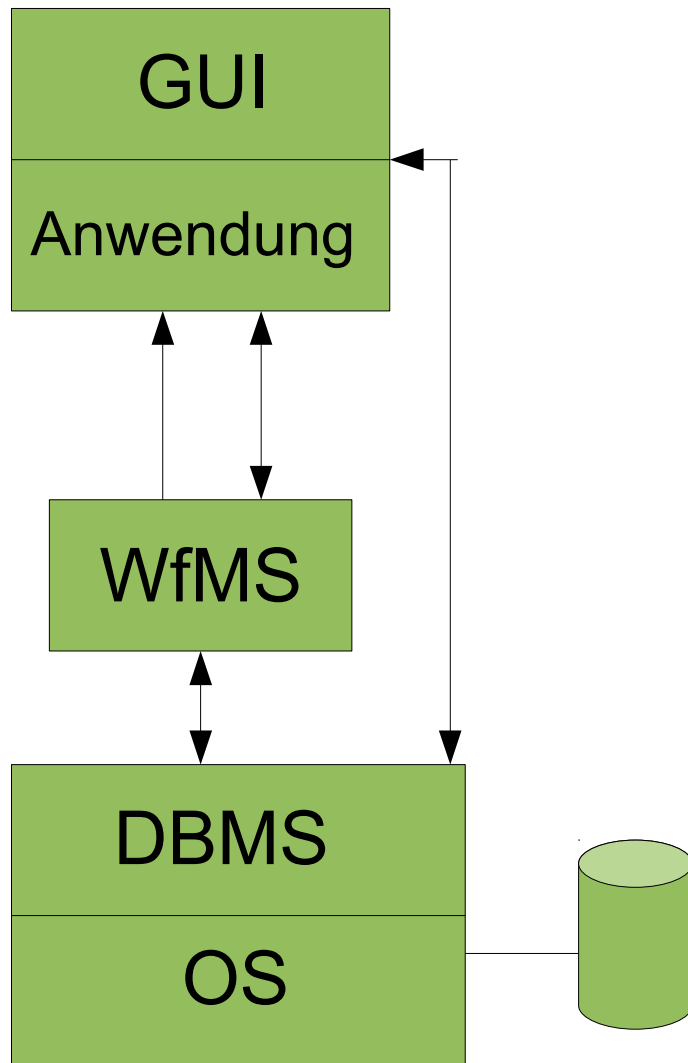
- **Workflow-Management:**
Zuständig für „Logistik“ des Workflows
 - Zeitliche Steuerung
 - Aufruf von Anwendungen
 - Berücksichtigung von „Constraints“
 - Zuordnung von Endbenutzern (Anwender, Akteure)
- **Anwendung:**
Ausführen von Arbeitsvorgängen
 - automatisiert – semi-automatisiert



Zur Diskussion:

Was könnten **Vorteile** der **Trennung** von
Workflow-Management und
Anwendungslogik sein ?

Antwort:



Zur Diskussion:

Was könnten **Vorteile** der **Trennung** von
Workflow-Management und
Anwendungslogik sein ?

Antwort:

- **Komplexitätsreduktion.**
- **Wiederverwendung** generischer
Prozessfunktionalitäten.
- Einfache **Prozessanalyse.**
- Einfache, modulare
Anwendungsentwicklung.

- **Lebenszyklus** von Workflows;
Überblick Modellierung und Ausführung.
- **Workflow-Aspekte** (Begriffe, Architekturansätze);
Integration von **Workflow-Management-Systemen** (WfMS) in
Anwendungssysteme.
- Referenzmodell der **Workflow Management Coalition** (WfMC):
 - Überblick
 - Komponenten und Schnittstellen
 - Anmerkungen

WfMC (**Workflow Management Coalition**):

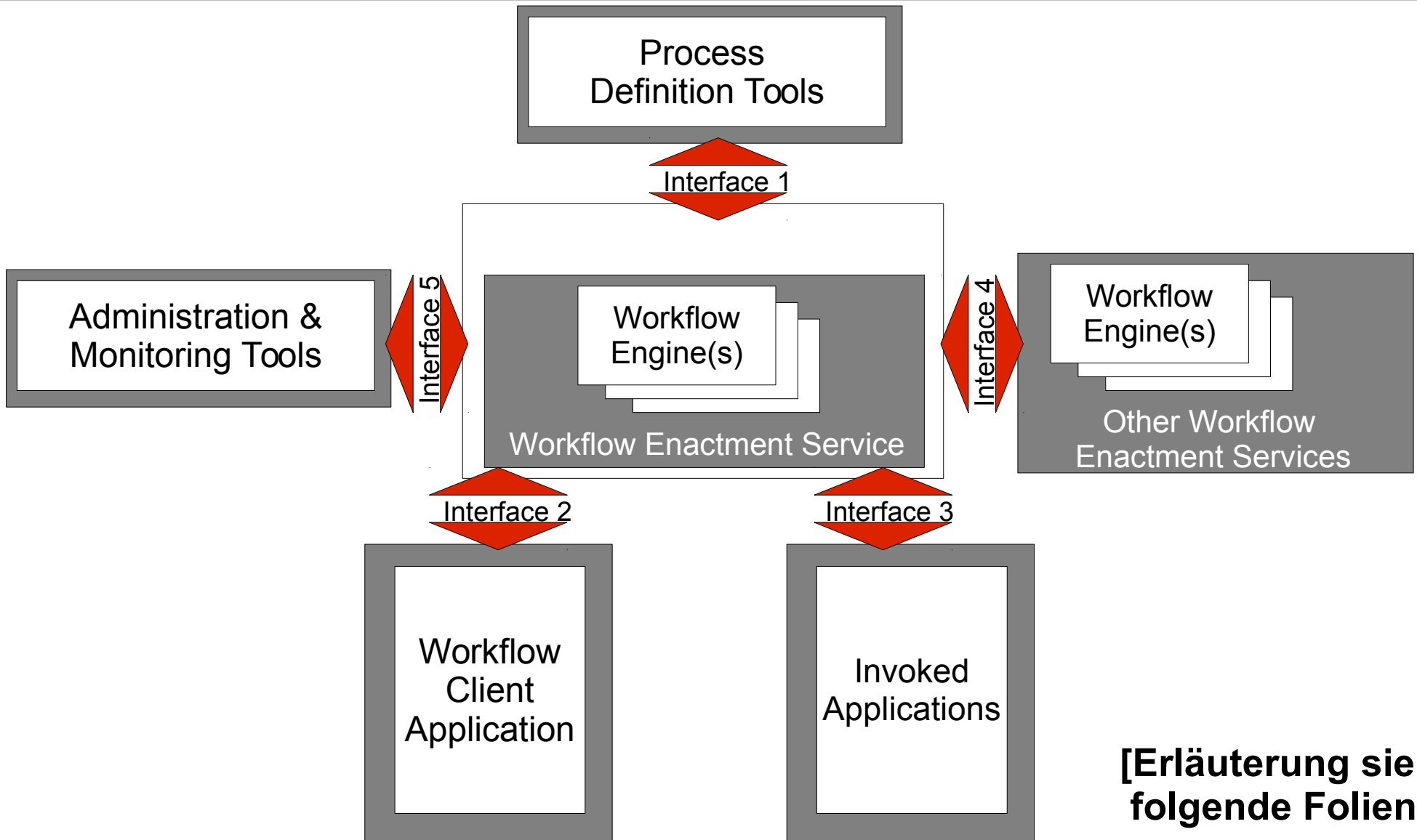
- Non-Profit-Organisation zur Standardisierung.
- davon ca. 50 Prozent WfMS Anbieter (Forschung ca. 25 Prozent).

Referenzmodell von 1995.

Ziele:

- Festlegung von **standardisierten Komponenten** und **Schnittstellen** für **Basisfunktionalität** eines **WfMS**.
- Generisches Ziel: **Interoperabilität**:
 - Standards zum Datenaustausch.
 - Verknüpfung von Ausführungsmaschinen unterschiedlicher Hersteller.
 - Unabhängigkeit der Anwendungen von konkreten WfMS.

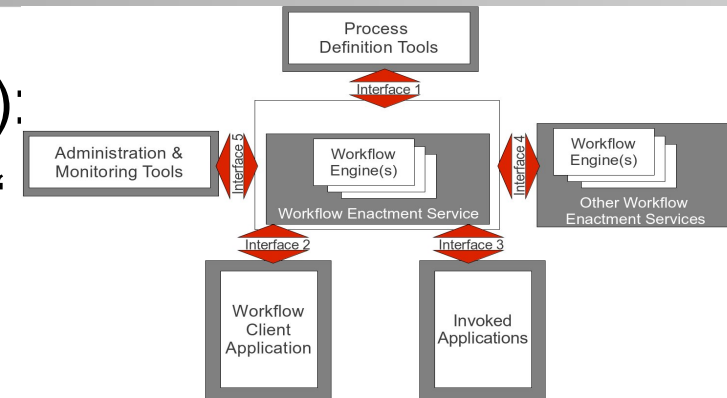
Referenzmodell der WfMC - Überblick



**[Erläuterung siehe
folgende Folien]**

„Workflow Enactment Service“ (Ausführung):

- Workflow Engine(s) als „Herz des Systems“
 - Aus Performanzgründen ggf. mehrere Engines, nicht relevant für Endnutzer.
- Ziel der Steuerung des Prozessablaufes:
„right activities at right order to right people“



Aufgaben des **Workflow Enactment Service**:

- Initialisieren und Beenden von Workflowinstanzen.
- Planung des Ablaufs von Instanzen.
- Zuordnung von Tasks zu Ressourcen.
- Starten von Anwendungen zur Ausführung von Aktivitäten.

„Process Definition Tools“:

Prozessdefinitionen im engeren Sinn:

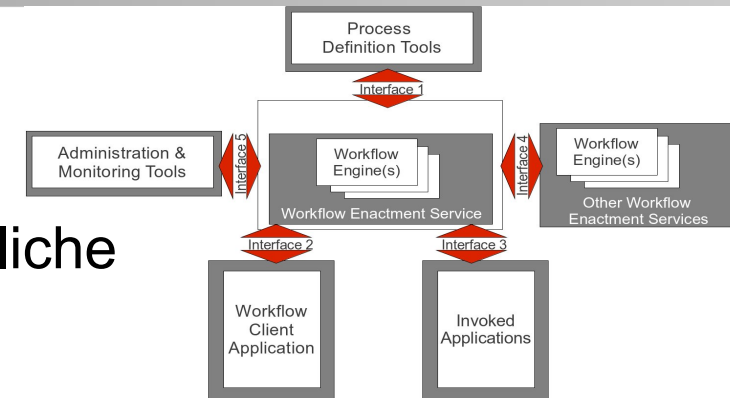
- Erstellung / **Spezifikation** von Prozessen: zeitliche und konditionale Abfolge von Komponenten, Constraints, etc.
- **Sprachen**: z.B Petri-Netze, BPMN.
- Spezifikation von Aktivitäten (z.B. Rollen zur Ausführung).
- Überprüfung auf **syntaktische Korrektheit** und **Konsistenz**.

Ressourcenklassifikation:

- z.B. Rollen, organisatorische Einheiten

Workflow-Analyse:

- Semantische Korrektheit, Simulation zur Analyse des Prozessverhaltens.

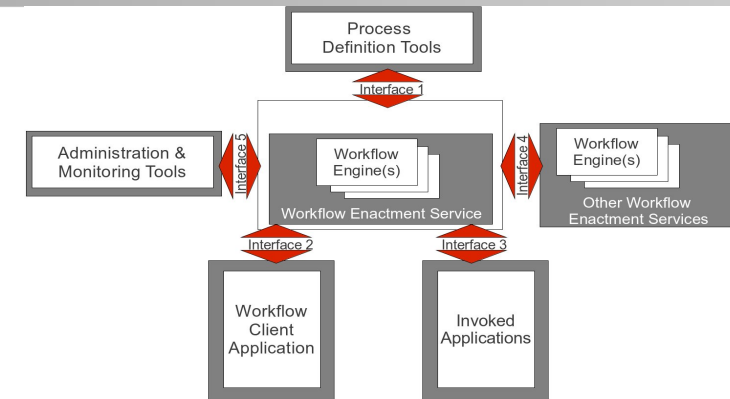


„Workflow Client Application“:

- Präsentation der Aufgaben („work items“) an Endnutzer als Arbeitsliste („worklists“).
- Informationen zu „work items“.
- Initiieren und Beenden von Aufgaben.
- Übergabe von Zustandsparametern (Aufruf **Workflow Engine**).

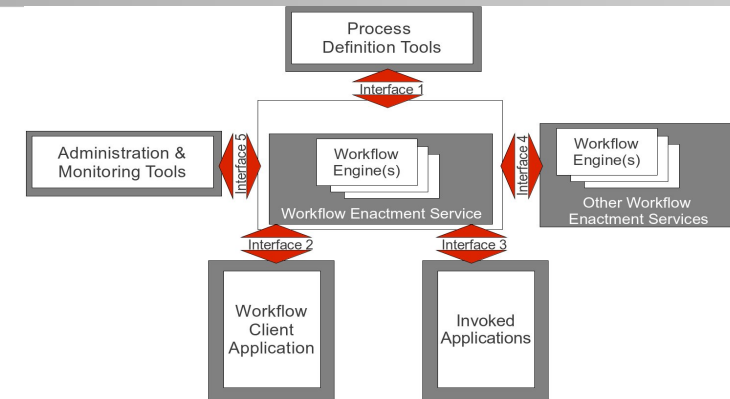
„Invoked Applications“:

- Anwendungen (Start durch Aufruf **Workflow Engine**) sowie Anwendungs-Parameter (Austausch mit **Workflow Engine**);
- Konfigurationsdateien.
- Datenzugriff.
- Automatisierte sowie interaktive Anwendungen.



„Other Workflow Enactment Service(s)“

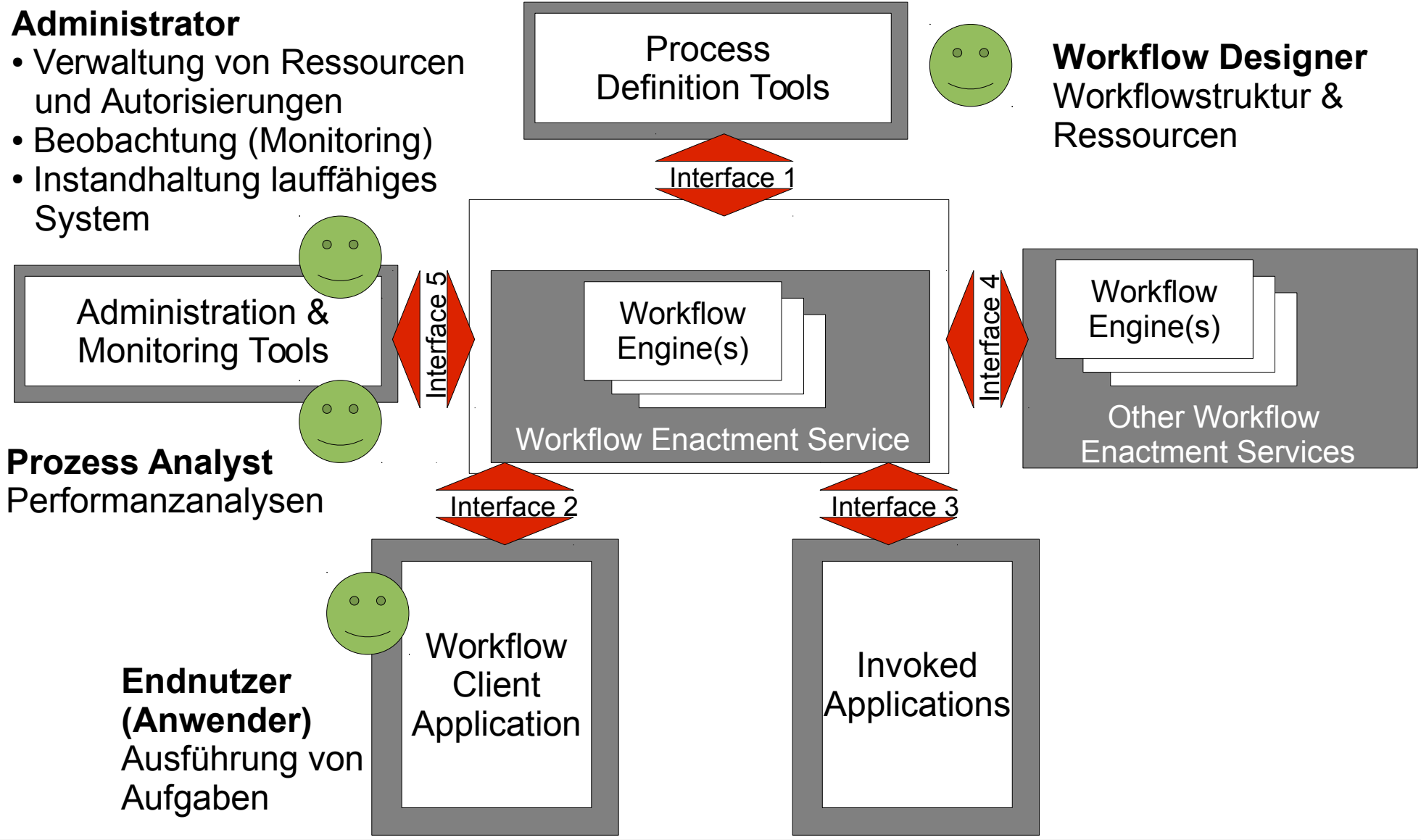
- Workflow Interoperabilität: Verknüpfung von autonomen Workflow-Systemen.
- Sequentielle und hierarchische Ausführung von Teil-Workflows.
- Beispiele: bereichs- bzw. unternehmensübergreifende Kopplung.



„Administration & Monitoring Tools“

- Operationales Management:
 - Verwaltung von Endnutzern.
 - Zustandsinformationen einer Prozessinstanz.
- Beobachtungs- und Report-Tools:
 - Verarbeitung vergangenheitsbezogener „Rohdaten“ zur Berechnung von Performanz-Indikatoren.

Einbindung von Rollen am Beispiel des WfMC-Referenzmodells



Schnittstelle 1: Workflow Process Definition

Read/Write-Interface: Verbindung von Workflow-Modellierungswerkzeugen mit zentraler Komponente - Austausch von Workflow-Schemata

Schnittstelle 2: Notifikationsdienst auf Clientseite

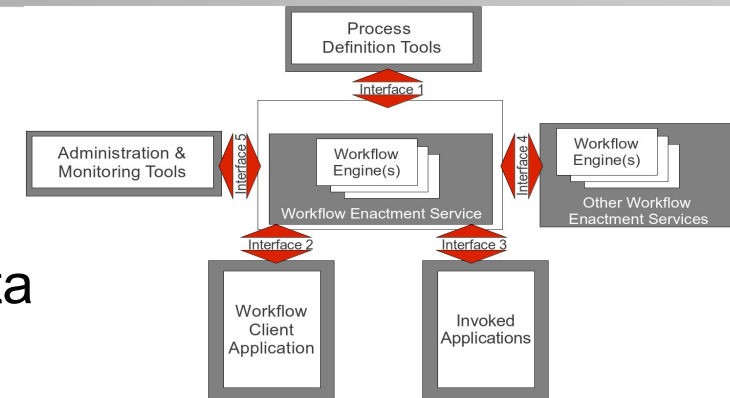
(Workflow Client Application Programming Interface)

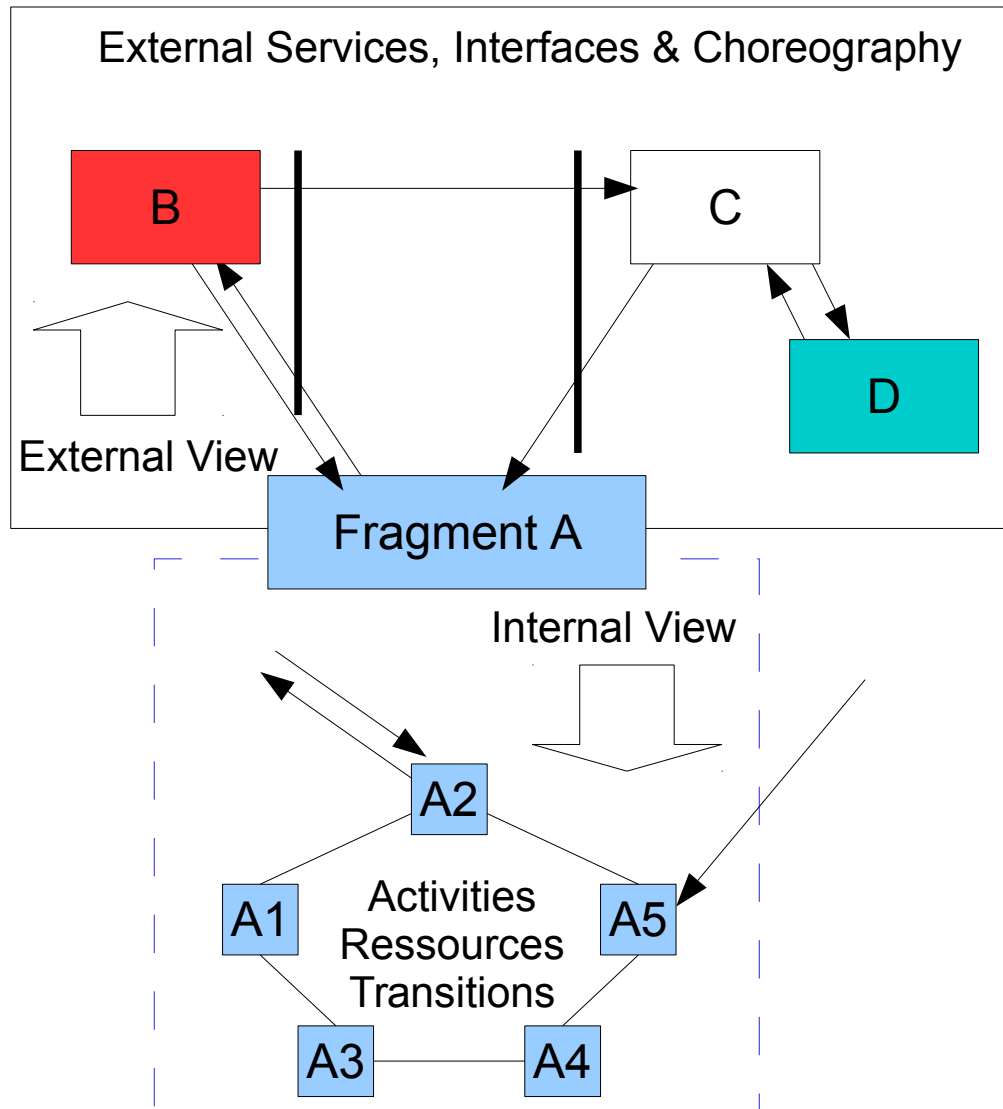
- Problem: ein Eingangskorb für mehrere Systeme

Schnittstelle 3: Programmierschnittstelle für den Applikationsaufruf

Schnittstelle 4: Austausch von workflow-relevanten Daten zwischen verschiedenen, autonomen **Workflow-Engines**. Ziel: Übergabe von Teil-Workflow zur Ausführung an anderes WfMS.

Schnittstelle 5: Administrations- und Monitoring-Werkzeuge anderer Hersteller. Information über aktuellen Ausführungszustand von Workflows und Administration laufender Workflows.





Externe Sicht:

- Nur Schnittstellen, ansonsten Black Box.
- Choreographie.

Interne Sicht:

- Spezifikation von Aktivitäten, Ressourcen, Schnittstellen.

Die Verknüpfung von **Workflow Enactment Services** findet man in der Praxis selten. Woran könnte das liegen?

Die Verknüpfung von **Workflow Enactment Services** findet man in der Praxis selten. Woran könnte das liegen?

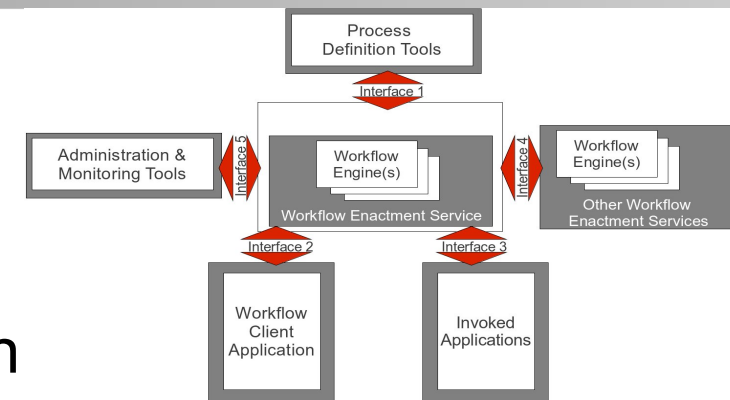
Antwort:

Prozesse müssen über alle Workflow Enactment Services modelliert werden. → In einer Notation.

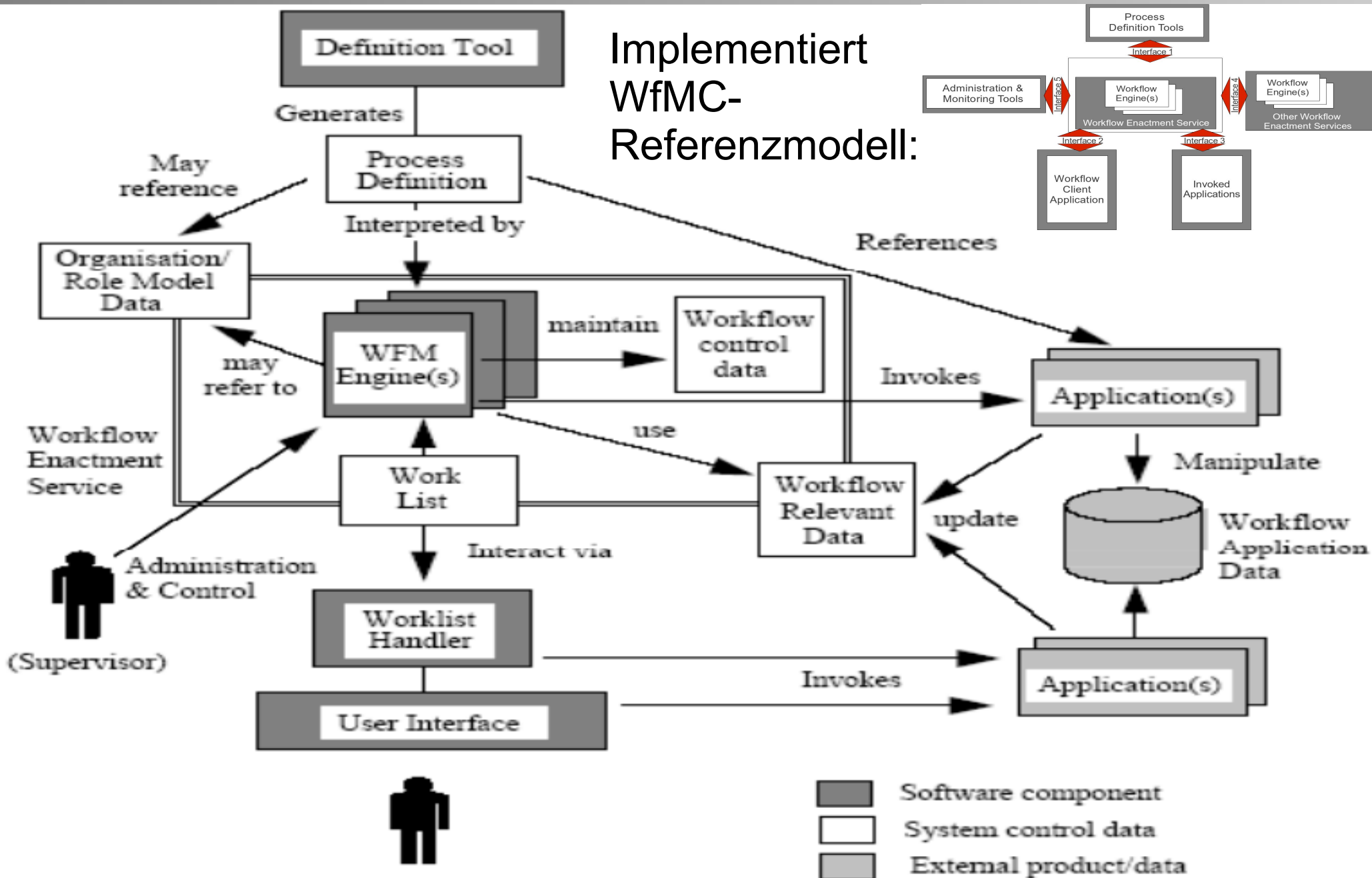
→ Komplette **Prozessmodelle** allein in dem passenden WES laufen lassen, anstatt sie über mehrere hinweg zu verteilen und damit **Heterogenität** zu fördern.

Gefördert wird dieses Vorgehen oft durch die fehlende **Interoperabilität** einzelner WES.

- **Datensicht** vernachlässigt.
- Spezifikation der **Schnittstellen** „nach außen“:
 - Schnittstelle 3: Fehlende Synchronisation von Anwendungen mit Arbeitslisten und Engine.
 - Schnittstelle 4: lose Kopplung (z.B. unternehmens-übergreifend) nicht adäquat unterstützt, keine Datenstruktur-Spezifikation.
- Zu geringe **Flexibilität**:
 - **Adaptive Workflows** → Anpassungen zur Laufzeit.
 - **Ad-hoc-Entscheidungen** (Problem lokale Autonomie).
- Berücksichtigung **neuer Workflow-Paradigmen**:
 - Z.B. Daten-zentrierte Workflows



Generisches WF-Management System (WfMC)



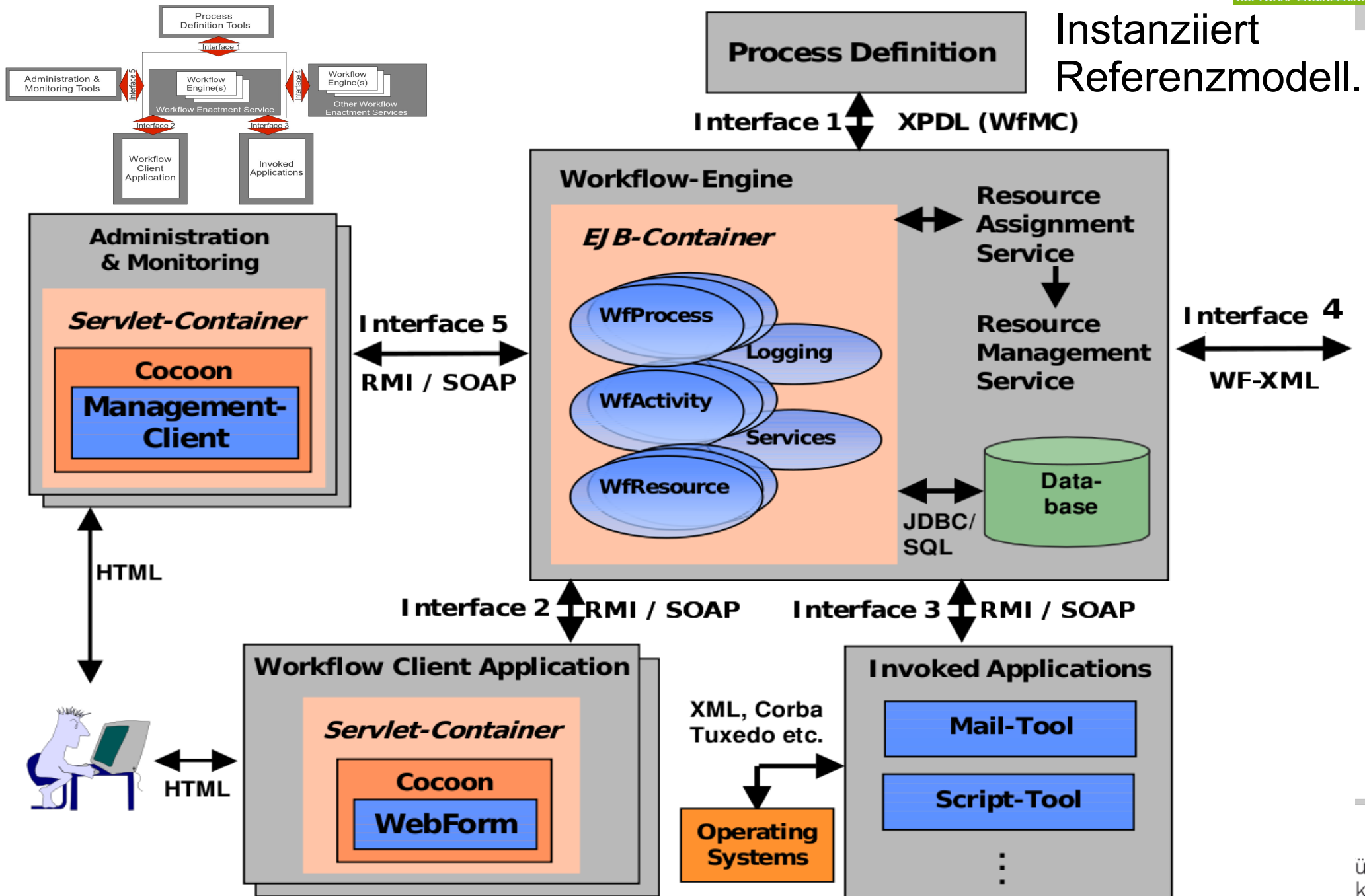
- Open-Source: <http://wfmopen.sourceforge.net/>
- J2EE-basiert (EJB Container, JBoss)
- Beherrscht nativ **XML Process Definition Language (XPDL)**
 - WfMC-Standard zum Austausch von Prozessdefinition zwischen Werkzeugen.
 - Aktuell nicht separat weiterentwickelt, an **BPMN 2.0** (Business Process Model and Notation) ausgerichtet.
- **Modulare** Umsetzung von WfMC Referenzarchitektur.

Im Einsatz z.B. in den Unternehmen:

- IniT: Im Bereich öffentliche Verkehrssysteme.
- T Mobile: Unterbau für T-Traffic (digitale Karten für Navis).

WfMC-Implementierungen

Beispiel WfMOpen



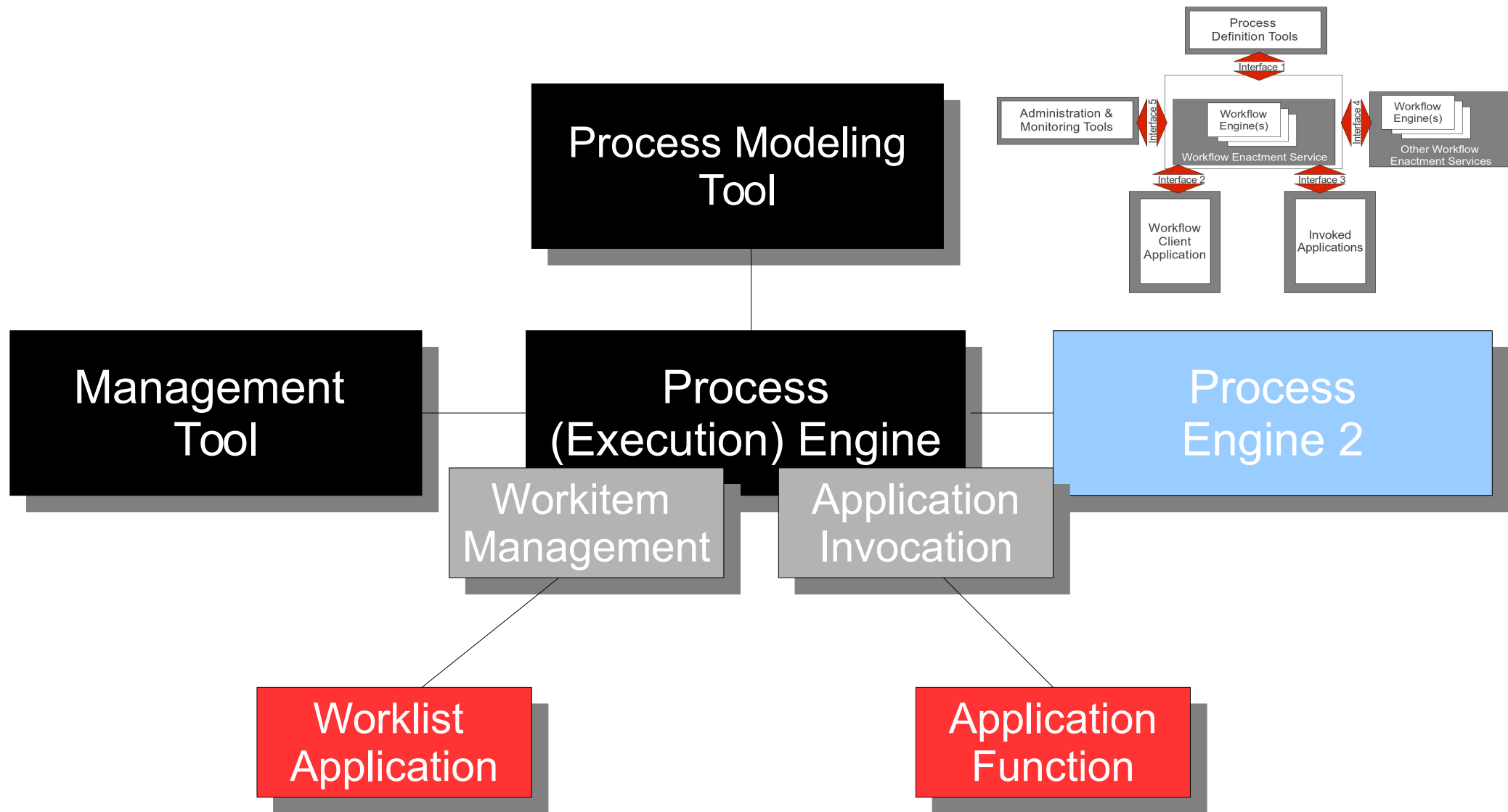
Weitere Open Source Implementierungen, z.B.:

- **Enhydra Shark** (s. Anhang)
- **Open Business Engine** (<http://obe.sourceforge.net/>)
- **PL/Flow** (<http://plflow.sourceforge.net/>)

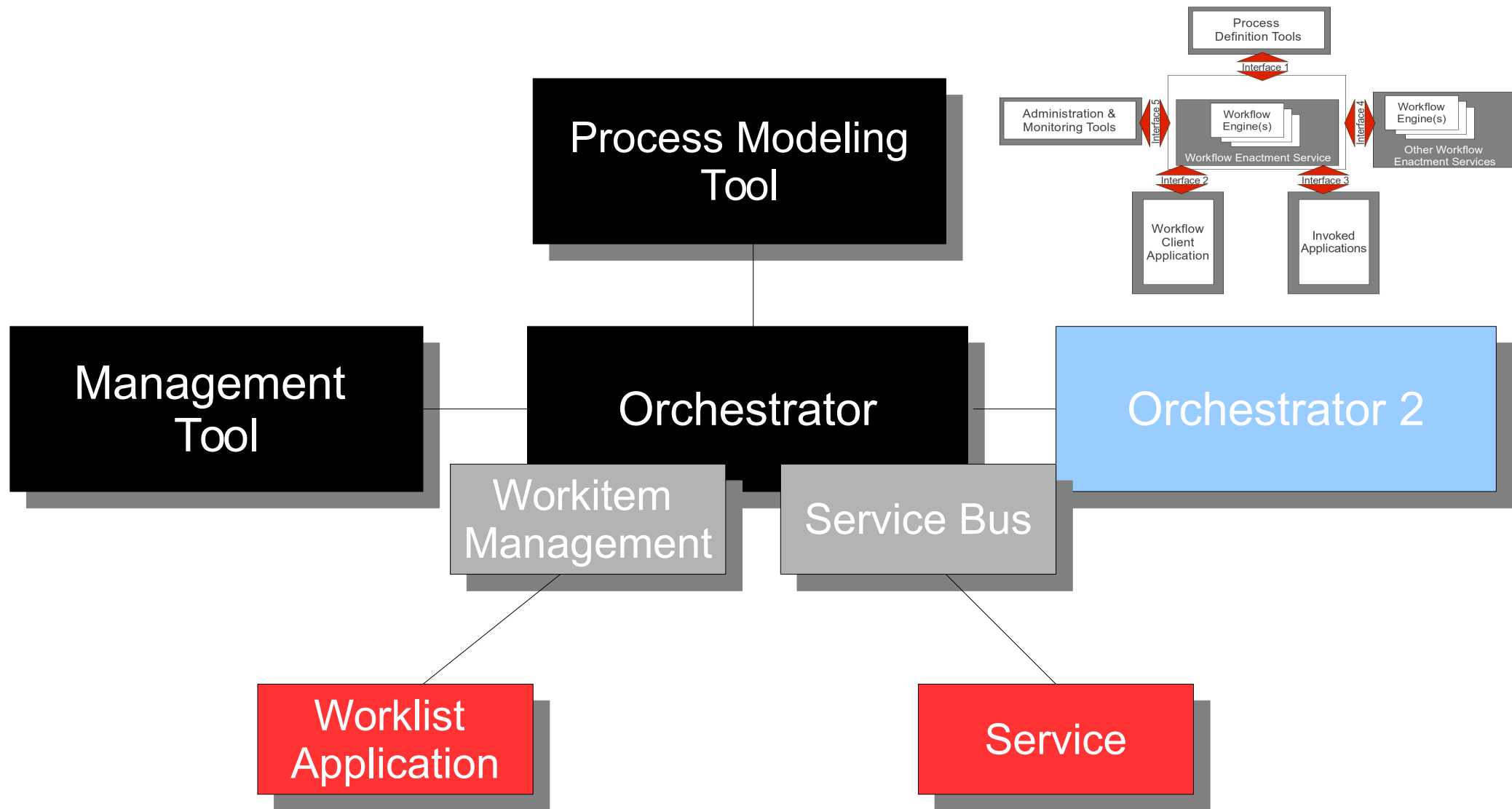
Viele kommerzielle Anbieter haben **Conformance-Test** der WfMC durchlaufen.

- Testen nur Existenz der Schnittstellen, nicht die eigentliche Architektur.

WfMC Referenzmodell: Moderne Terminologie

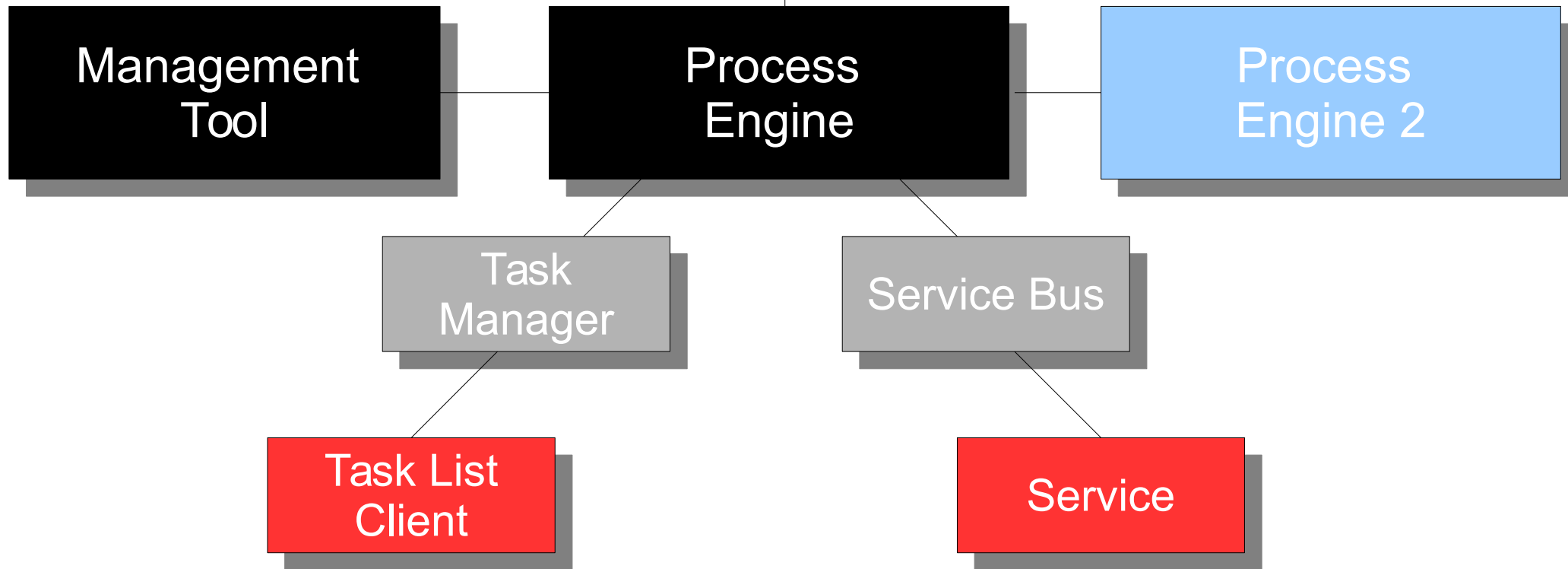
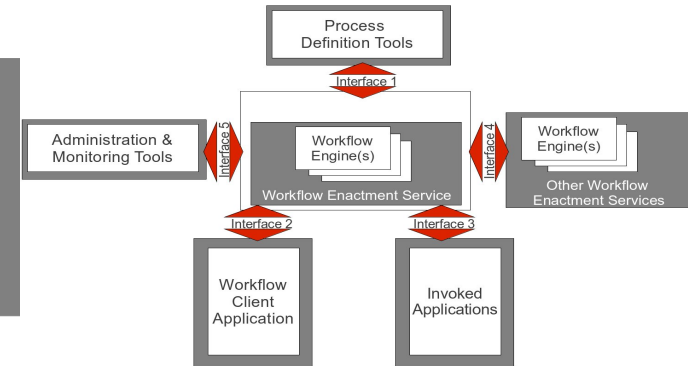
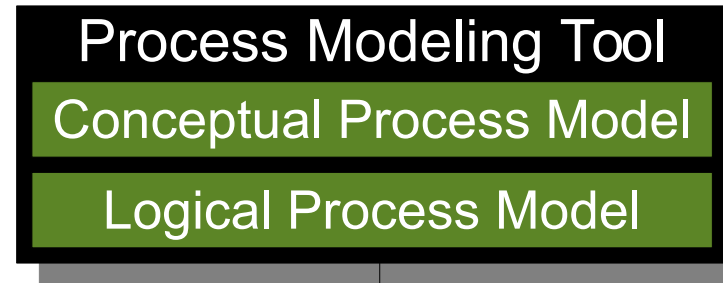


WfMC Referenzmodell: Noch modernere Terminologie



Terminologie im Kontext von BPEL4People

[BPEL4People erweitert BPEL um Modellierung menschlicher Aktivitäten.]



In diesem Abschnitt:

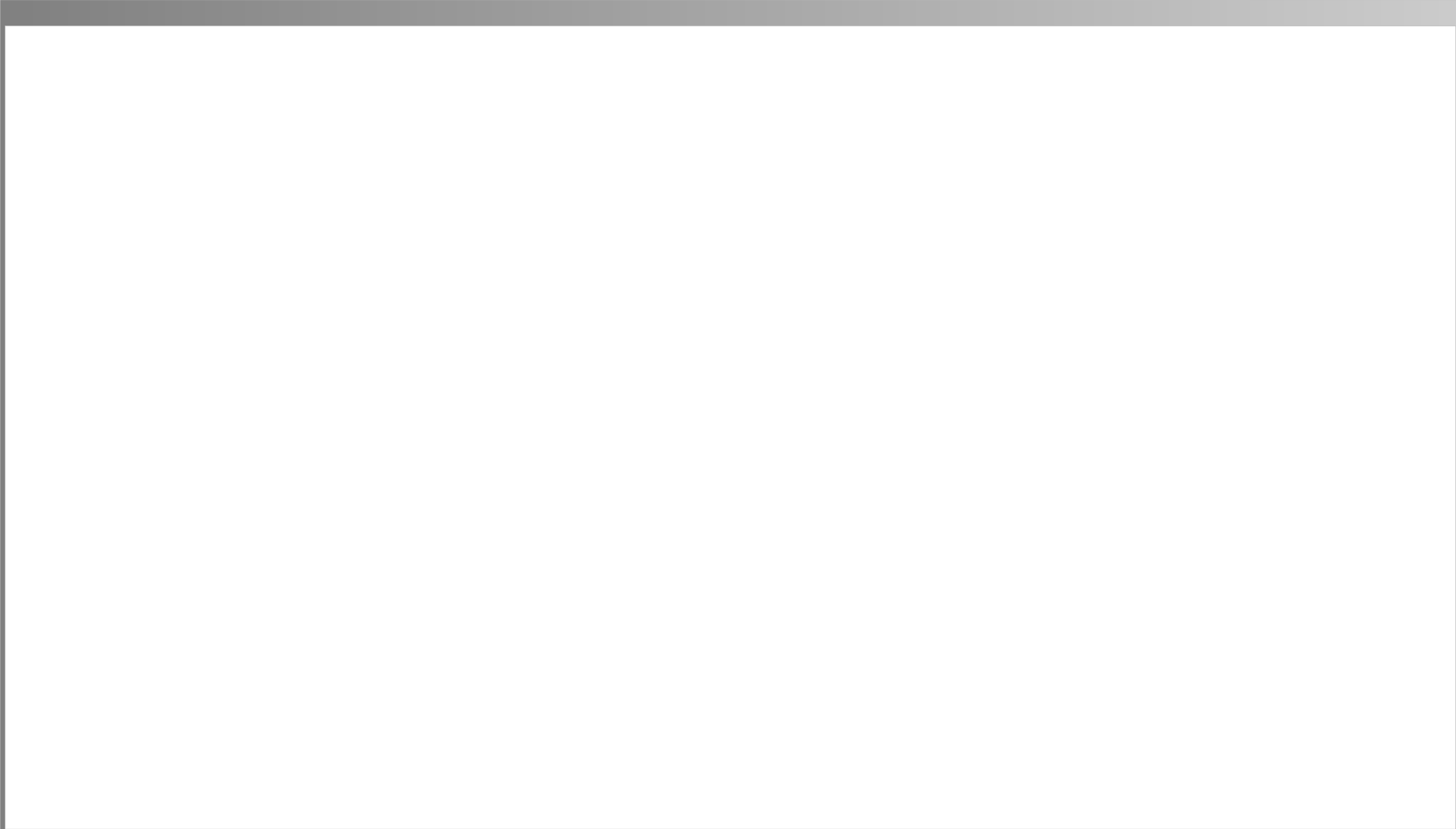
- **Lebenszyklus** von Workflows
- Zusammenhang **Modellierung** und **Ausführung**
- **Anforderungen** und **Architekturansätze** für **WfMS**
- **Integration** von WfMS in **Anwendungssysteme**
- **Referenzmodell** der **WfMC**

Im nächsten Abschnitt:

Workflow-Automatisierung (im Kontext der Business Process Execution Language (BPEL)).

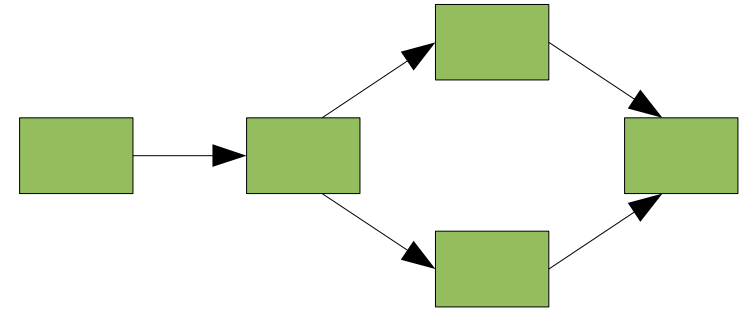
Anhang (zusätzliche Informationen zum selbständigen Nacharbeiten)

Methodische Grundlagen
des Software-Engineering
SS 2014



Grobes Vorgehen:

1. **Definition** des Workflowmodells
(Schema) und anderer Daten mit
einem Modellierungstool.



2. Spezifikation von **Ressourcen** (z.B. Personen).

3. Implementierung / Integration der von den einzelnen
Arbeitsschritten ausgeführten Programme.

1. **Initialisieren** und Starten einer neuen **Workflow-Instanz**.
2. Solange **Arbeitsschritte** auszuführen sind und kein „Terminate“-Arbeitsschritt erreicht wird:
 - **Ausführung** eines Arbeitsschrittes (automatisiert, semi-automatisiert).
 - Weiteres Vorgehen nach Beendigung des Arbeitsschrittes erfolgt nach Vorgaben des **Workflow-Schemas** (Bedingungen, Verzweigungen, ...).
3. **Terminierung** der Workflow-Instanz.

Entscheidungen beim Übergang vom **Implementierungsmodell** zur **Implementierungsarchitektur**:

- Werden Module durch **Basissysteme** (z.B. Betriebssystem, Middleware) unterstützt ?
- Wie werden (persistente) Daten einer **Komponente** verwaltet ?
- Wie wird **Kommunikation** zwischen Modulen realisiert ?

Architekturansätze unterscheiden sich hinsichtlich dieser Fragen.

→ **Keine allgemeine Implementierungsarchitektur.**

Konkretisierung der Implementierungskonzepte: in **Implementierungsphase.**



Funktionale Anforderungen:

- Werden durch **Workflow-Sprache** und ihre Ausdrucksfähigkeit, sowie durch Vorgaben bezüglich Benutzerschnittstellen bestimmt.

Nicht-Funktionale Anforderungen:

- Werden durch beabsichtigtes Einsatzgebiet eines **Workflow-Management-Systems** beeinflusst.

- **Implementierung des Workflow-Meta-Schemas:**
 - Funktionsaspekt
 - Verhaltensaspekt
 - Informationsaspekt
 - Organisationsaspekt
- **Applikationsintegration (Operationsaspekt):**
Nutzung der Eigenschaften externer Programme,
z.B. hinsichtlich transaktionaler Ausführung.
- **Bereitstellung von Benutzerschnittstellen:**
 - Endbenutzer (Anwender), Entwickler, Administrator, etc.

Abstraktion:

- Beschreibung der wesentlichen Komponenten und Schnittstellen, unabhängig von technischer Umsetzung.
- **Spezifikation** der Interfaces (abstrakte Terme und konkrete „Bindings“).
- Konkrete **Bindings** konnten im Lauf der Zeit den Entwicklungen angepasst werden (API in C → Middleware → Web Services).

Lebenszyklus eines Prozesses:

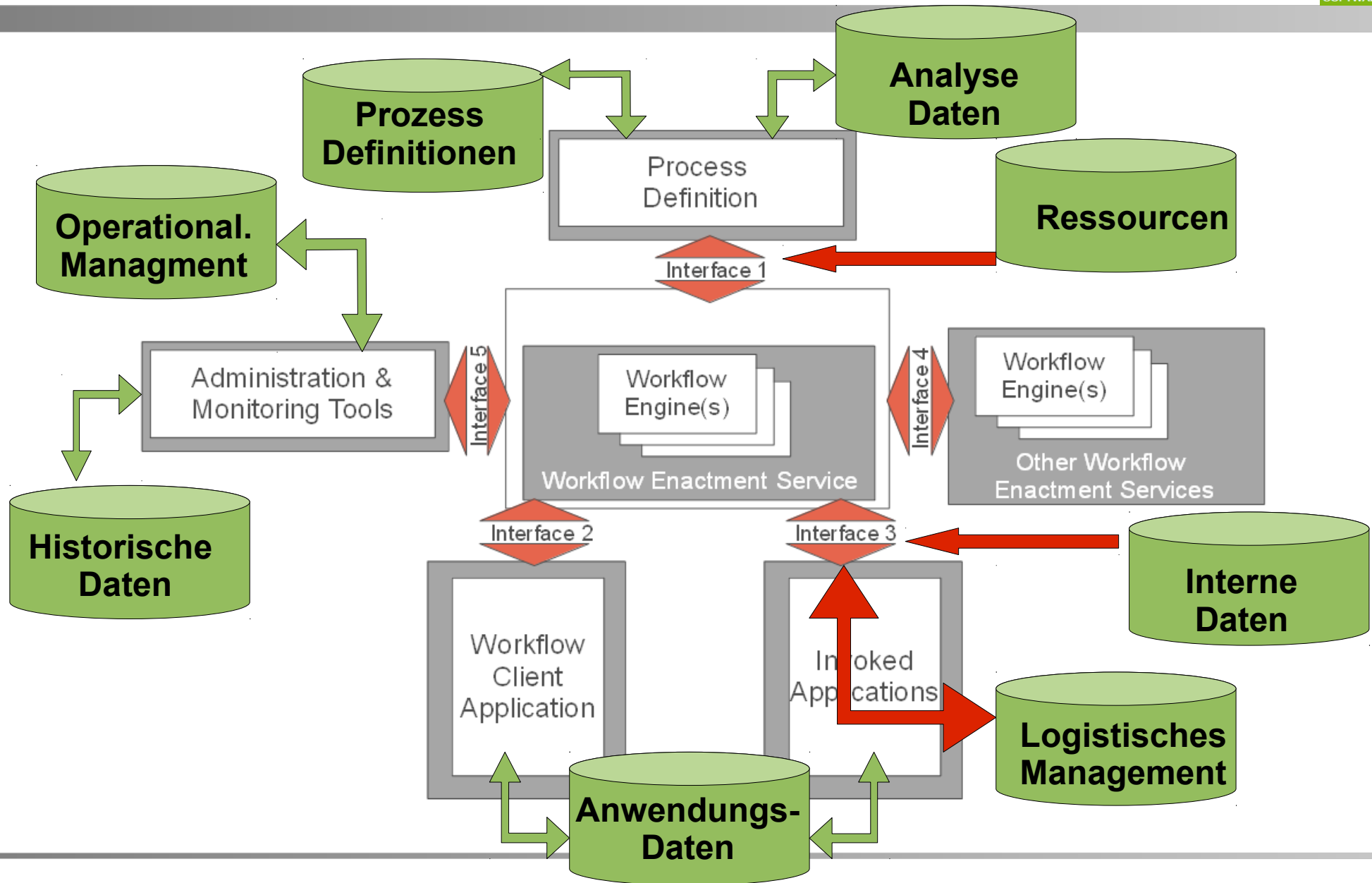
- Phasen des Lebenszykluses tragen zum Gesamtbild bei.

Integrierte Sicht von Daten, Prozess und Organisation.

- **Prozess-Definitionen**, inkl. Definitionen der Aktivitäten.
- **Ressourcen** (Rollen, Organisationen).
- **Analysedaten**.
- **Operationales Management** (technische Parameter).
- **Historische Daten**: Nachverfolgung einer Prozessinstanz.
- **Anwendungsdaten**:
 - Zugriff über Applikation, nicht über Engine.
 - Masterdaten (generisch) und Prozess-Instanzdaten (spezifisch) .
- **Interne Daten**: technische Daten.
- **Logistisches Management**.

[Zuordnung in Referenzmodell siehe nächste Folie.]

Daten am Beispiel des WfMC Referenzmodells



- Open-Source:
 - <http://sourceforge.net/projects/sharkwf/>
 - Auch bekannt als „Together Workflow Server“:
<http://www.together.at/prod/workflow/tws>
- Java-basiert:
 - Als **Library** implementiert (Plain Old Java Object (POJO)).
 - Verschiedene **Wrapper** für unterschiedliche Szenarien:
 - EJB, CORBA, RMI, WebService.
- **Adaptiert** unter anderem WfMC:
 - Aufbau WfMC-konform, implementiert Interfaces.
 - Bietet Schnittstellen konform zu anderen Standards.

WfMC-Implementierungen

Beispiel Enhydra Shark

Methodische Grundlagen
des Software-Engineering
SS 2014



LEHRSTUHL 14
SOFTWARE ENGINEERING

