

Willkommen zur Vorlesung
*Methodische Grundlagen
des Software-Engineering*
im Sommersemester 2012
Prof. Dr. Jan Jürjens

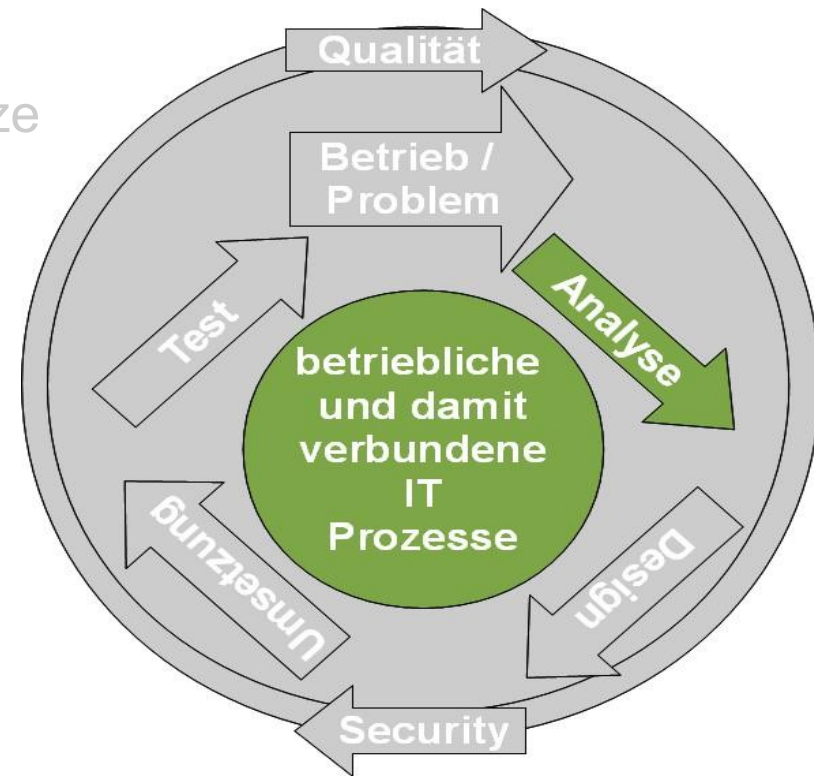
TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

2.5 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)]**

Einordnung Workflow-Management-Systeme

- Anwendungsbeispiel Finanz- und Versicherungsdomäne
- **Geschäfts-Prozesse**
 - Grundlagen Geschäfts-Prozesse
 - Einführung in die BPMN
 - Elektronische Prozessketten
 - Grundlagen der GP-Modellierung: Petri-Netze
 - **Workflow-Management-Systeme**
 - Workflow-Automatisierung
- Qualitätsmanagement
- Testen
- Sicherheit
- Sicheres Software Design



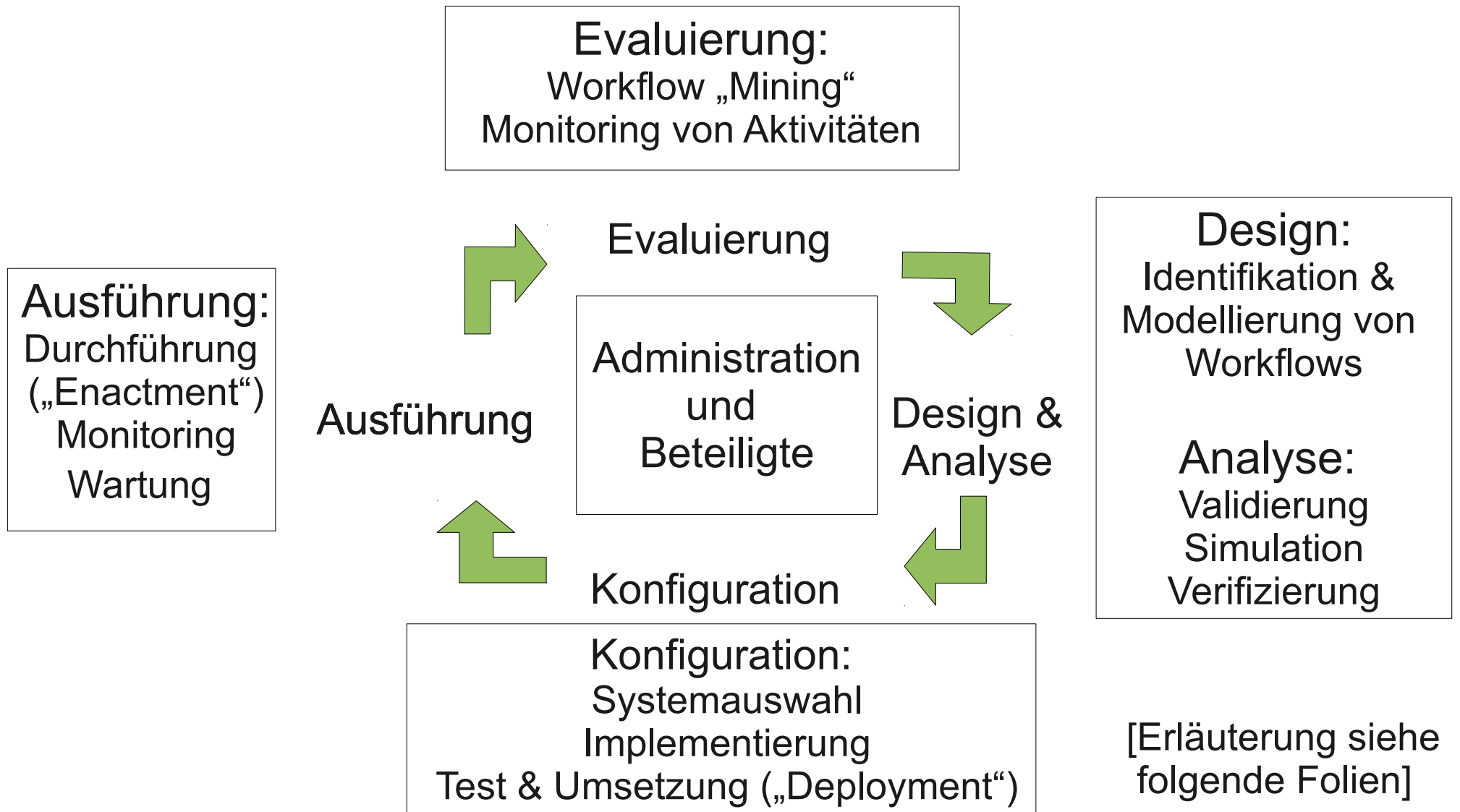


In den letzten Abschnitten haben wir uns mit der Modellierung und Analyse von Geschäftsprozessen / Workflows beschäftigt.

In diesem Abschnitt geht es darum, wie man solche Modelle nicht nur zur Dokumentation und Analyse benutzen kann, sondern im Kontext der tatsächlichen Ausführung der Workflows („Workflow Management“). Wir betrachten insbesondere praktische Systeme, die dies unterstützen (Workflow-Management-Systeme).

Das Thema Workflow-Automatisierung werden wir dann im nächsten Abschnitt betrachten (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



Quelle: M. Weske 2007

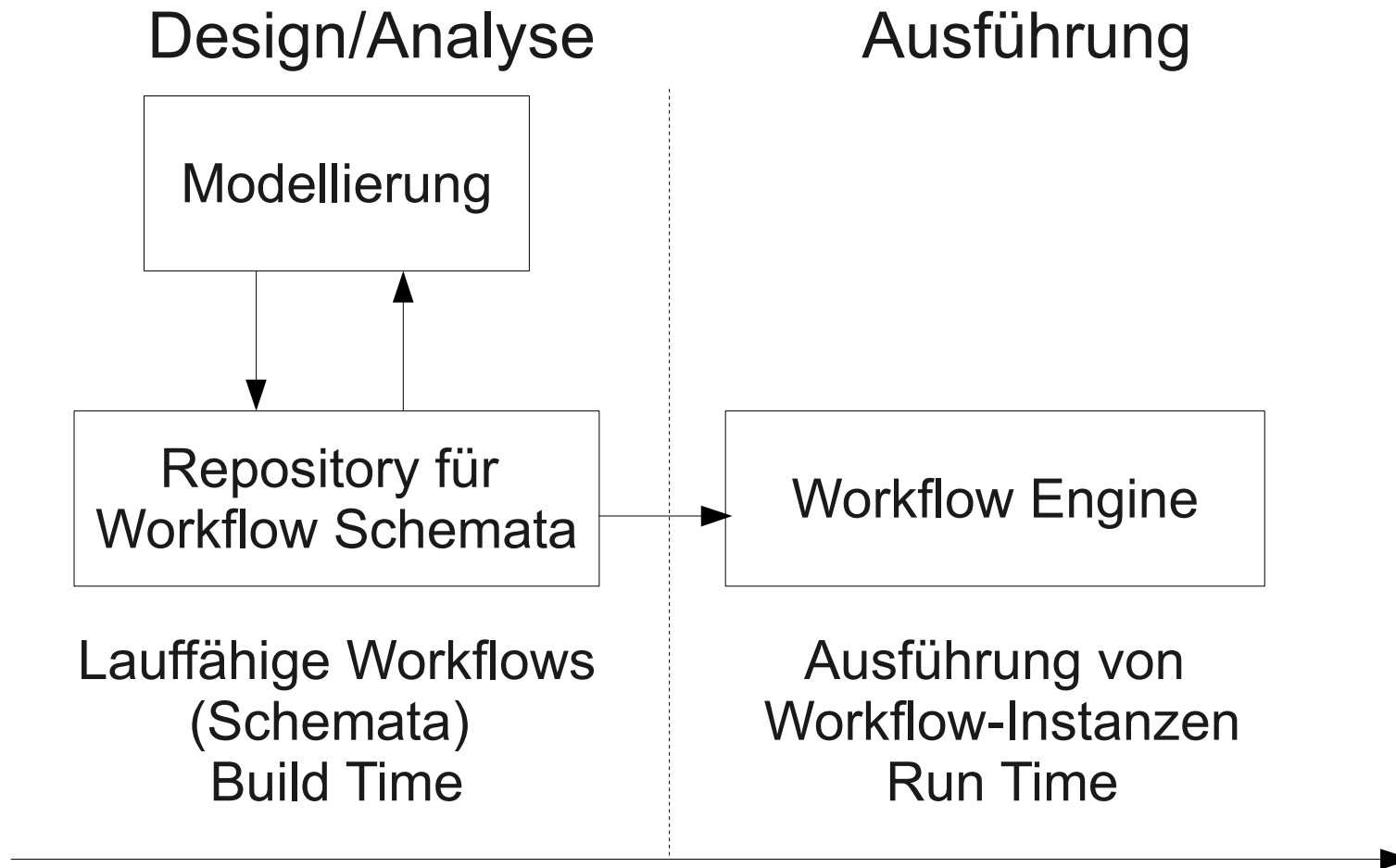
- Design und Analyse
 - Design:
 - Identifikation, Abgrenzung, Spezifikation von Zielen
 - Modellierung der Workflow Schemata typischerweise in graph. Notation (Kommunikationsbasis mit Anwendern)
 - Analyse:
 - Validierung (Feedback der Anwender)
 - Simulation (Prozessverhalten, z.B. Dauer, Lastverteilung)
 - Verifizierung (z.B. Erkennung von Deadlocks)



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

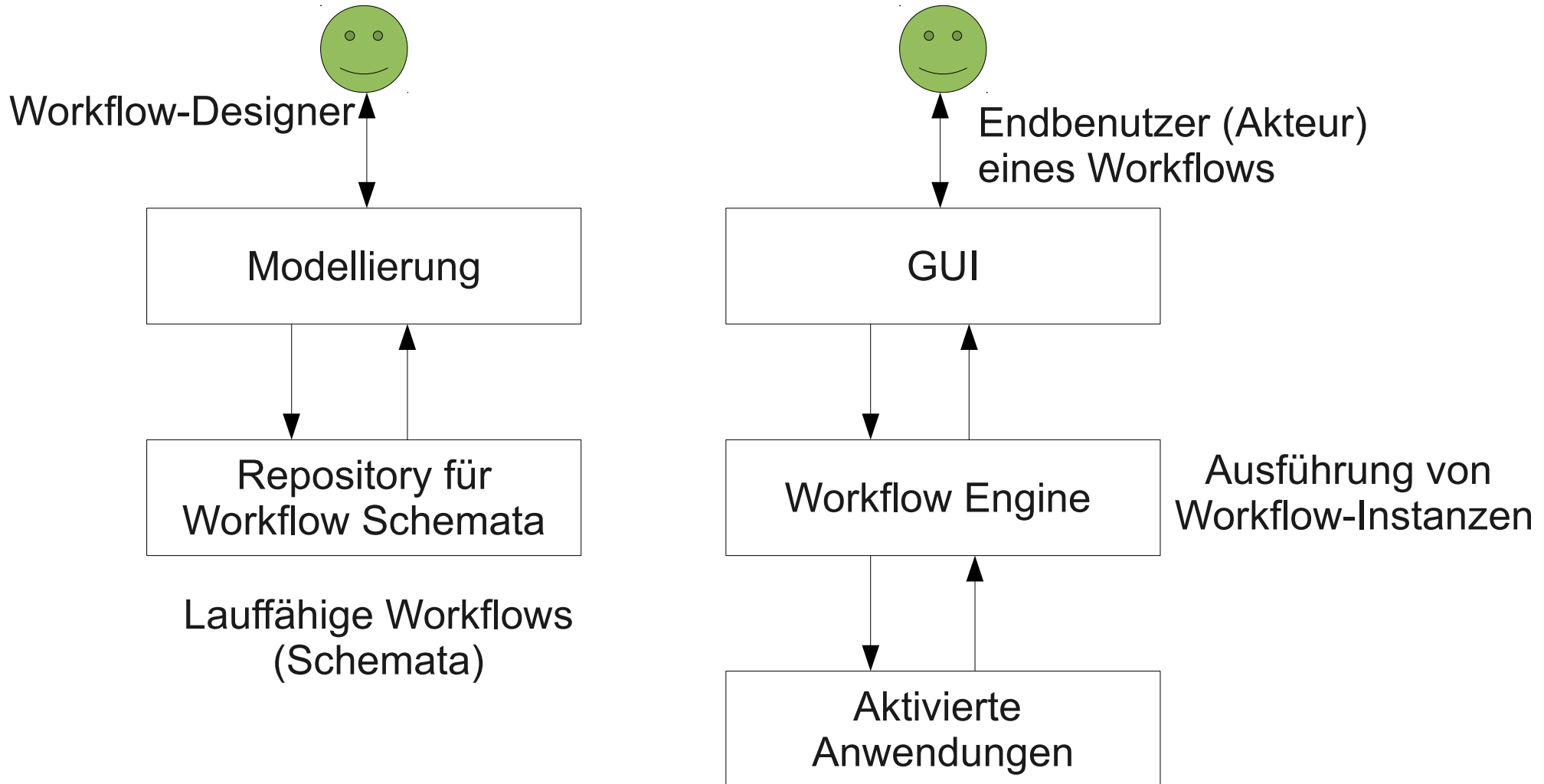
- Ausführung
 - Durchführung („Enactment“): korrekte Ausführung („Orchestration“) von Workflowinstanzen gemäß der spezifizierten Bedingungen (zeitl. Ablauf, Rechte für Endbenutzer etc.).
 - Monitoring (z.B. Statusinformationen von aktiven Instanzen, Events in Log-Files)
 - Betrieb und Wartung

Modellierung vs. Ausführung (1)



Quelle: M. Weske, 2007

Modellierung vs. Ausführung (2)

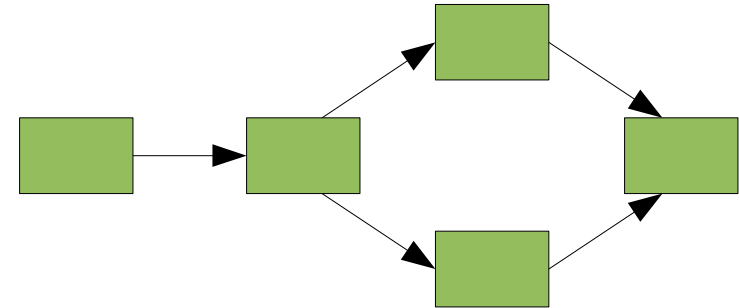


Quelle: M. Weske

Erstellung eines lauffähigen Workflows

Grobes Vorgehen:

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



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

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

1. Initialisieren und Starten einer neuen Workflow-Instanz
(gesteuert durch ein Ereignis, z.B. Eintreffen einer Dienstreisegenehmigung oder eines Auftrages)
2. Solange Arbeitsschritte auszuführen sind und kein „Terminate“-Arbeitsschritt erreicht wird:
 - Ausführung eines Arbeitsschrittes (automatisiert, semi-automatisiert)
 - Das weitere Vorgehen nach Beendigung des Arbeitsschrittes erfolgt nach den Vorgaben des Workflow-Schemas (Bedingungen, Verzweigungen, ...).
3. Terminierung der Workflow-Instanz

- 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**

- **Funktionaler Aspekt:** beschreibt die funktionalen Einheiten, d.h. die Struktur des Workflows mit elementaren und zusammengesetzten Arbeitsschritten (Unter-Workflows).
- **Verhaltensbezogener Aspekt:** konstituiert den Kontrollfluss zwischen den elementaren und zusammengesetzten Arbeitsschritten eines Workflows unter Berücksichtigung von kausalen und temporalen Zusammenhängen.
- **Datenbezogener Aspekt:** betrifft den Datenfluss eines Workflows.

- **Organisations-Aspekt:** beschreibt die organisationsbezogenen Inhalte mit der Definition von Organisationsstrukturen und deren Population, sowie die Festlegung, wer die verschiedenen Operationen eines Workflows ausführen kann / darf.
- **Operationaler Aspekt:** regelt die Einbindung von Applikationen in einen elementaren Arbeitsschritt. Applikationen sind Programme oder Programmsysteme, die für die Ausführung eines Workflows benötigt werden.

Die genannten Aspekte bilden das Fundament eines WfMSs.

Erweiterungen sind möglich und in vielen Anwendungsfällen notwendig, z.B. ist in vielen Umgebungen ein Sicherheitsaspekt erforderlich, um Zugriffskontrolle zu regeln.

- Ziel: Beschreibung des implementierungs-unabhängigen Teils der Architektur eines WfMSs.
- Spezifikation der wesentlichen Module oder Komponenten eines WfMSs.
- Die wesentlichen funktionalen Komponenten der Architektur eines WfMSs umfassen Module, welche die Aspekte eines WfMSs realisieren.
- Ergänzend ist ein Steuermodul (nicht unbedingt zentral !) erforderlich, sowie...
- ... Hilfsmodule, welche allgemeine Dienste zur Verfügung stellen.

Implementierungsmodell eines WfMS (2)

Steuermodul

funktionaler Aspekt

verhaltensbezogener
Aspekt

datenbezogener
Aspekt

organisatorischer
Aspekt

operationaler
Aspekt

weitere Aspekte

Datenbankzugriff

weitere
Hilfsmodule

- Beim Übergang vom Implementierungsmodell zur Implementierungsarchitektur sind folgende Entscheidungen zu treffen:
 - Werden Module durch Basissysteme (z.B. Betriebssystem, Middleware) unterstützt ?
 - Wie werden die (persistenten) Daten einer Komponente verwaltet ?
 - Wie wird die Kommunikation zwischen den Modulen realisiert ?
- Architekturansätze unterscheiden sich stark hinsichtlich dieser drei Fragen, daher keine allgemeine Implementierungsarchitektur.
- Die Konkretisierung der Implementierungskonzepte erfolgt in der eigentlichen Implementierungsphase.

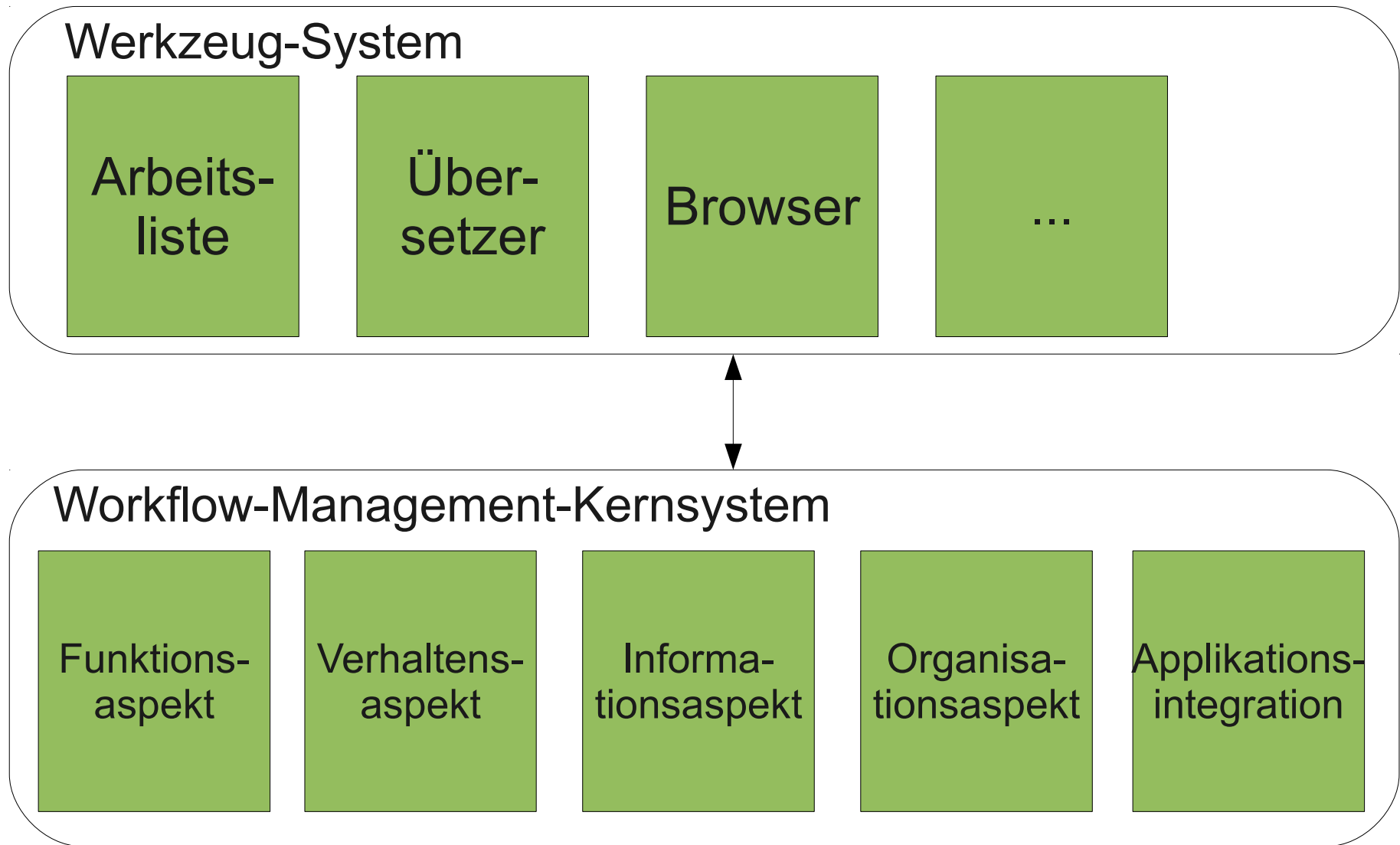


- **Funktionale** Anforderungen
 - Werden vor allem durch die Workflow-Sprache und ihre Ausdrucksfähigkeit und durch Vorgaben bezüglich der Benutzerschnittstellen bestimmt.
- **Nicht-Funktionale** Anforderungen
 - Werden vor allem durch das beabsichtigte Einsatzgebiet eines Workflow-Management-Systems 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.

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

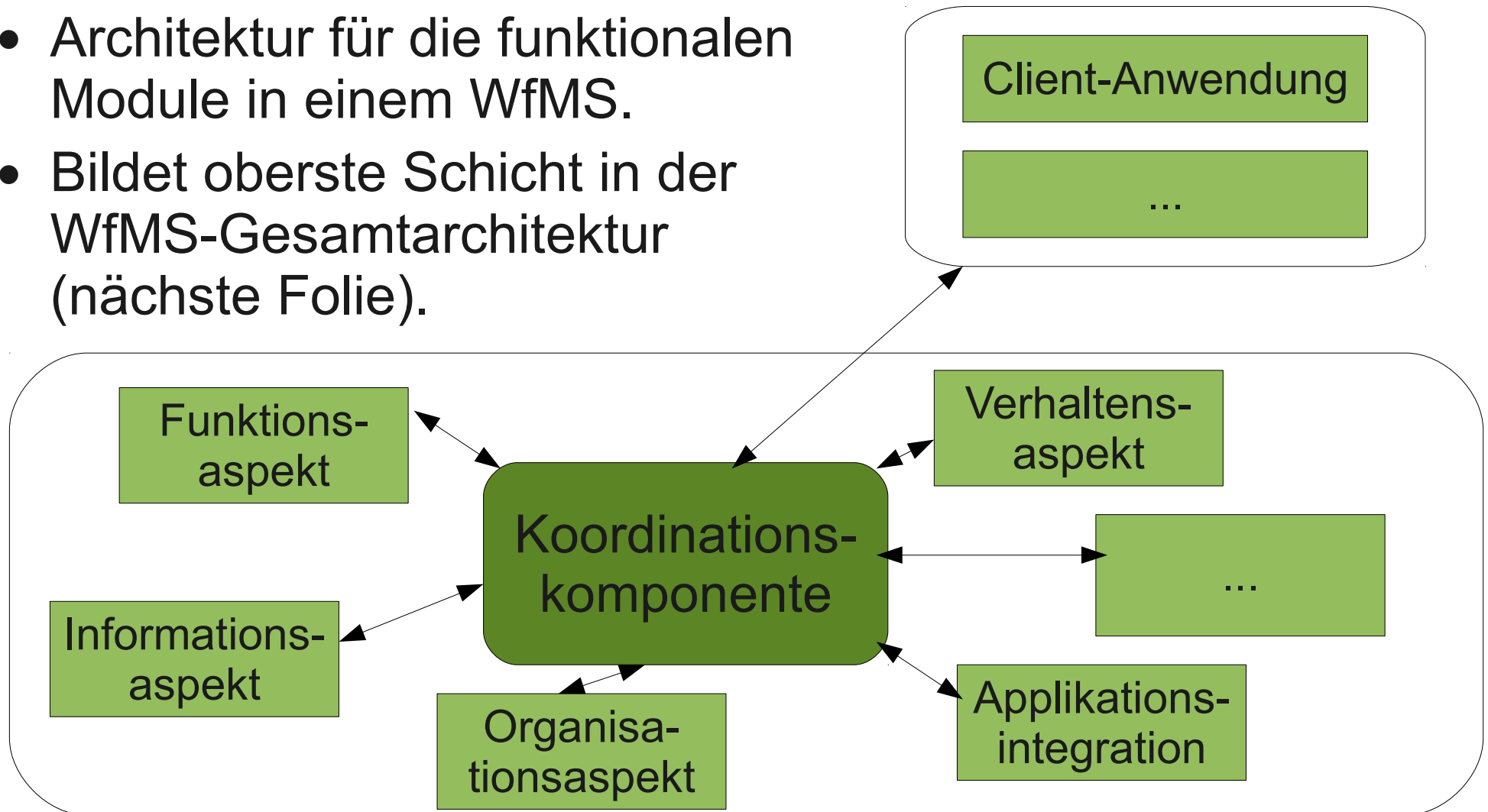
Sicherheit als Workflow-Aspekt wird eine Orthogonalität zu allen klassischen Workflow-Aspekten zugeschrieben. Was könnte damit gemeint sein ? Welche Konsequenzen hat das ?

Sicherheit als Workflow-Aspekt wird eine Orthogonalität zu allen klassischen Workflow-Aspekten zugeschrieben. Was könnte damit gemeint sein ? Welche Konsequenzen hat das ?

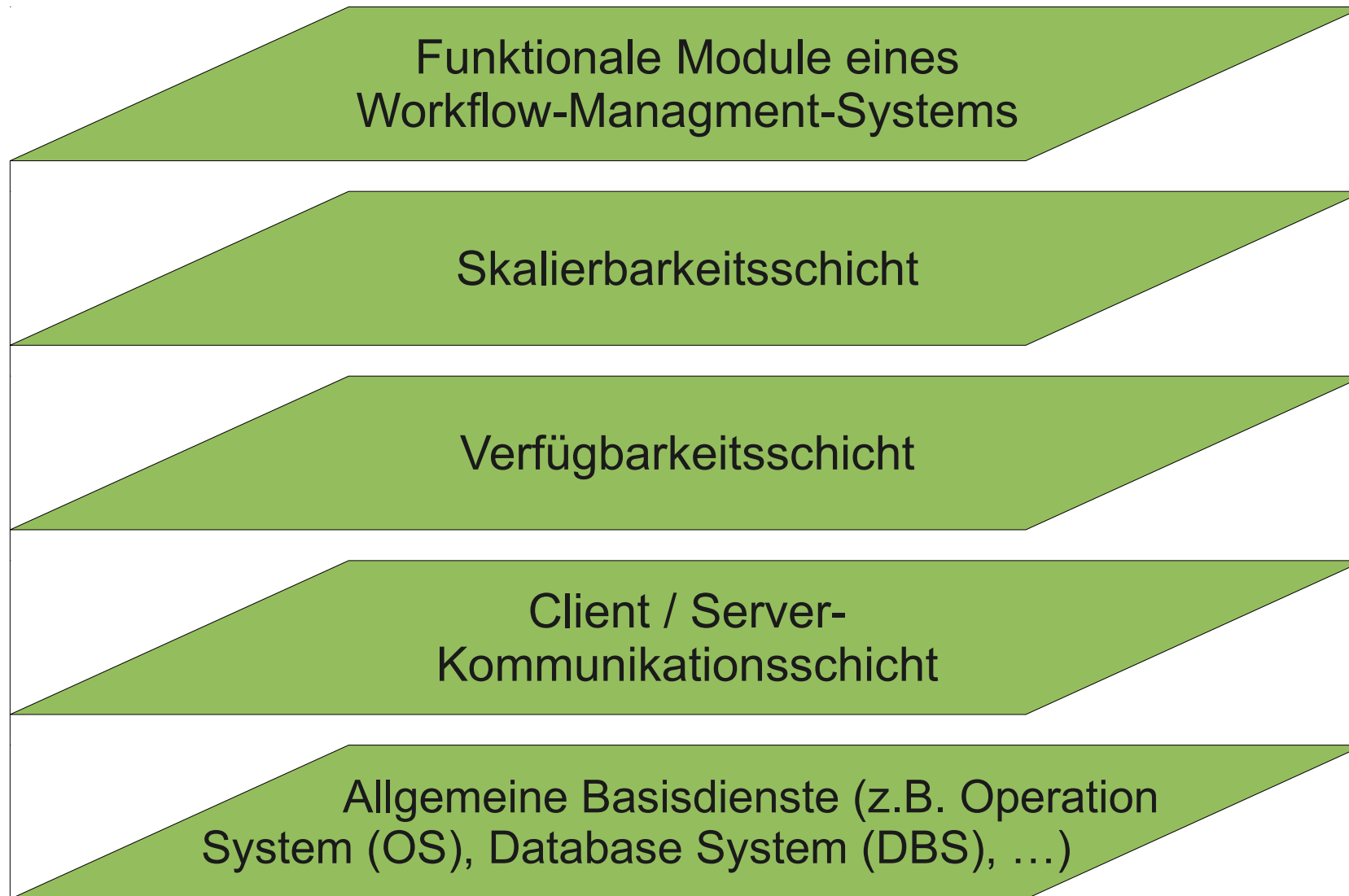
Antwort: Sicherheit ist deswegen orthogonal, da sie alle anderen Aspekte durchdringt. Sicherheit kann nicht getrennt betrachtet werden. Als Konsequenz ist das Einführen von Sicherheitsaspekten in die Workflow-Betrachtung sehr komplex und wird auch heutzutage noch sehr nachlässig berücksichtigt im Kontext der Workflows.

Komponentenarchitektur eines WfMS

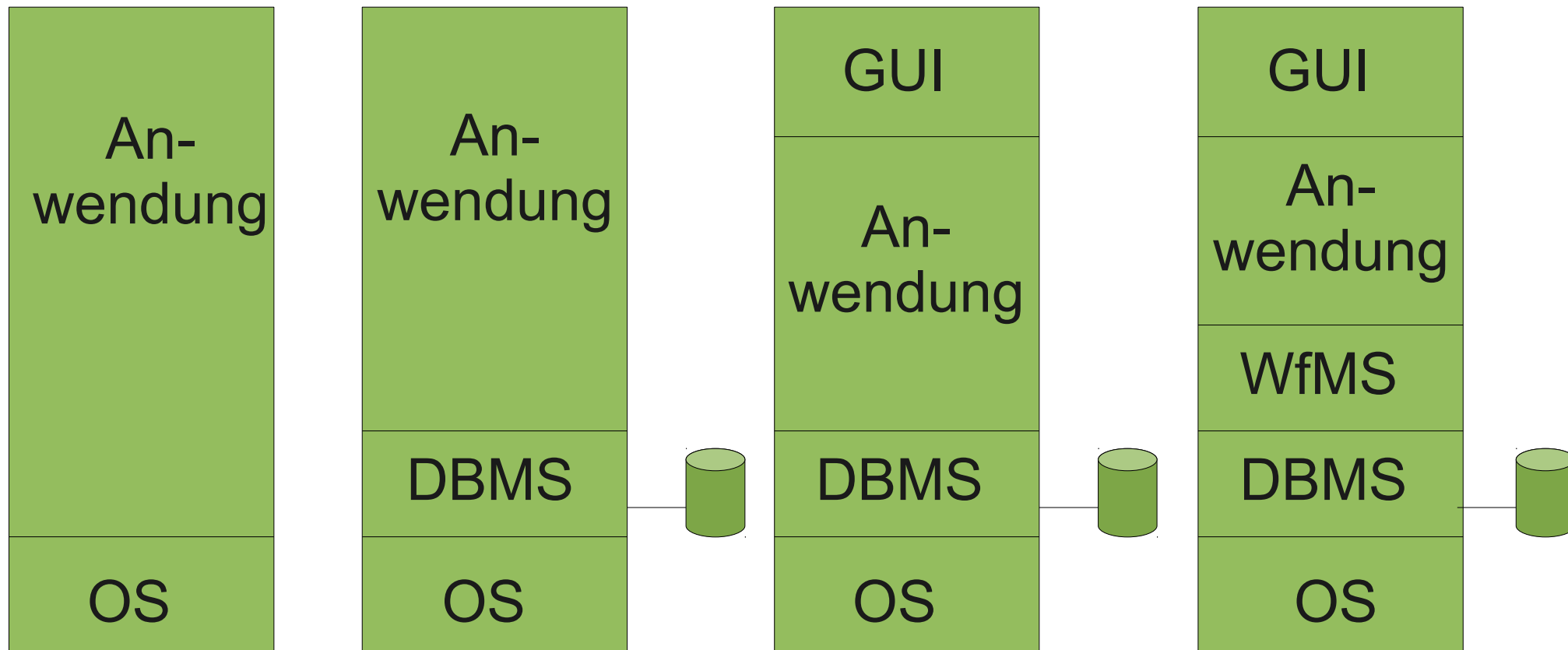
- Architektur für die funktionalen Module in einem WfMS.
- Bildet oberste Schicht in der WfMS-Gesamtarchitektur (nächste Folie).



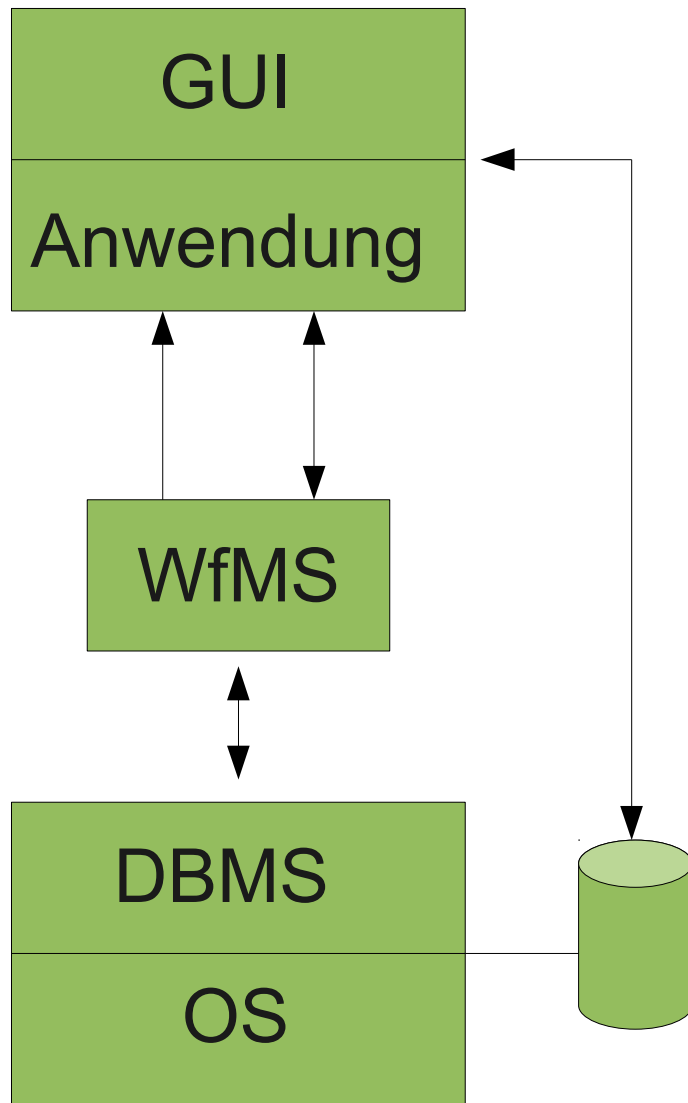
Schichtenarchitektur eines WfMS



Evolution von Anwendungssystemen

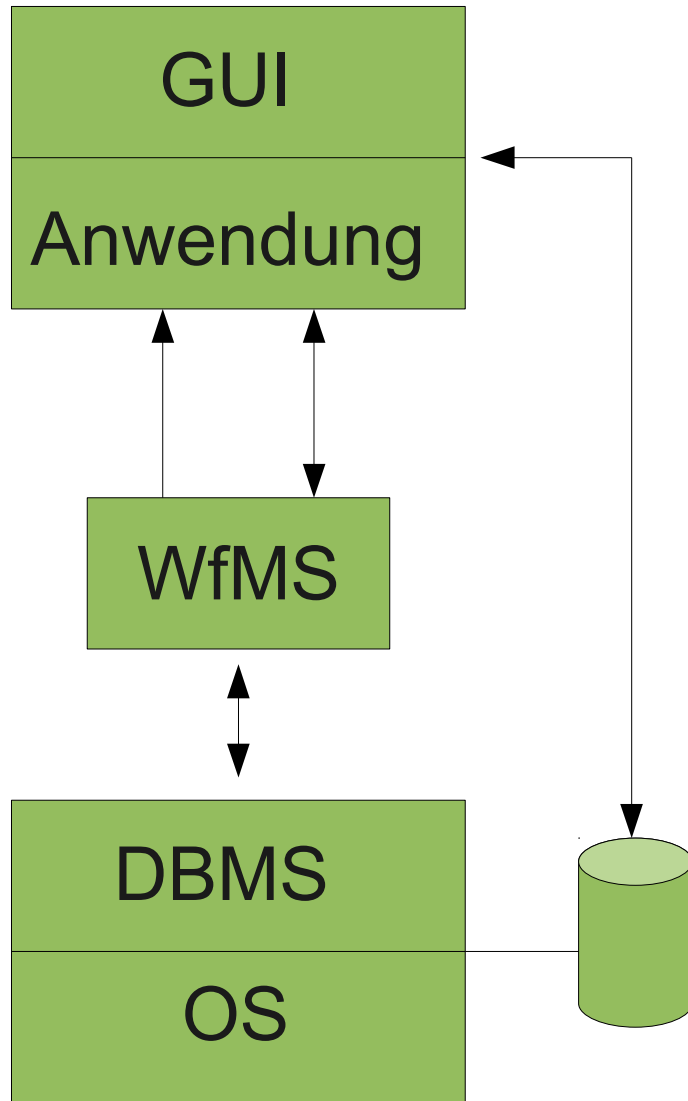


Einbindung von WfMS in Anwendungssysteme (2)



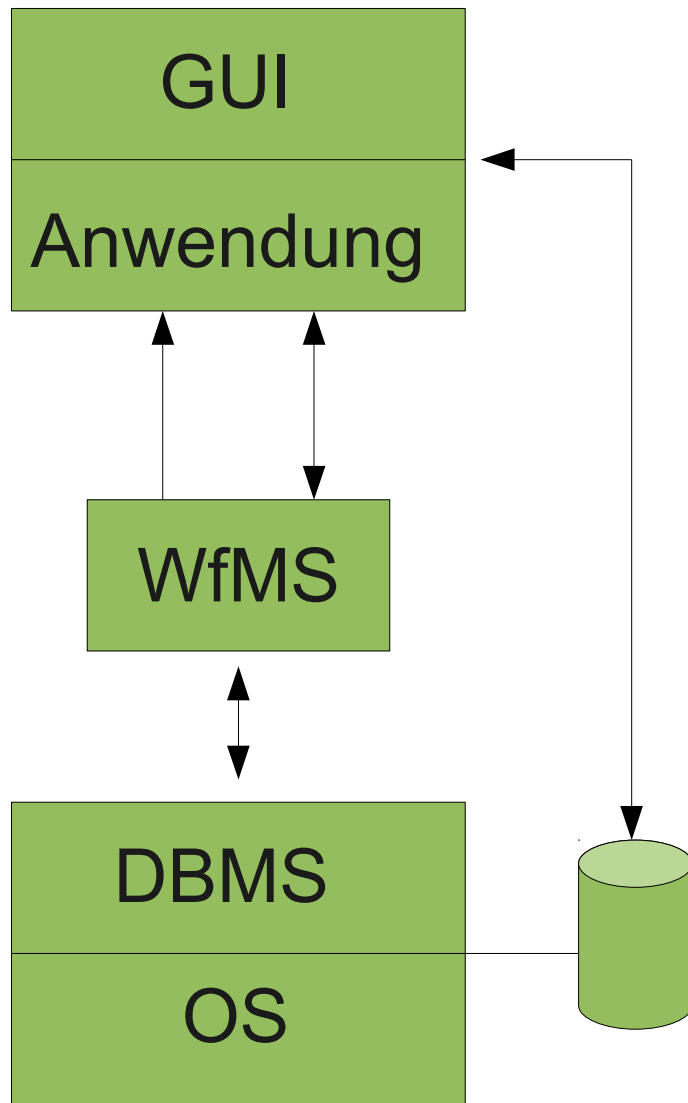
- Explizite Trennung von Workflow-Management und Anwendung
- Workflow-Management:
Zuständig für „Logistik“ des Workflows
 - Zeitliche Steuerung
 - Aufruf von Anwendungen
 - Berücksichtigung von „Constraints“
 - Zuordnung von Endbenutzern (Anwender, Akteure)
- Anwendung:
Ausführen von Arbeitsvorgängen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

Einbindung von WfMS in Anwendungssysteme (3)



Vorteile Trennung Workflow-Management und Anwendungslogik

- ?



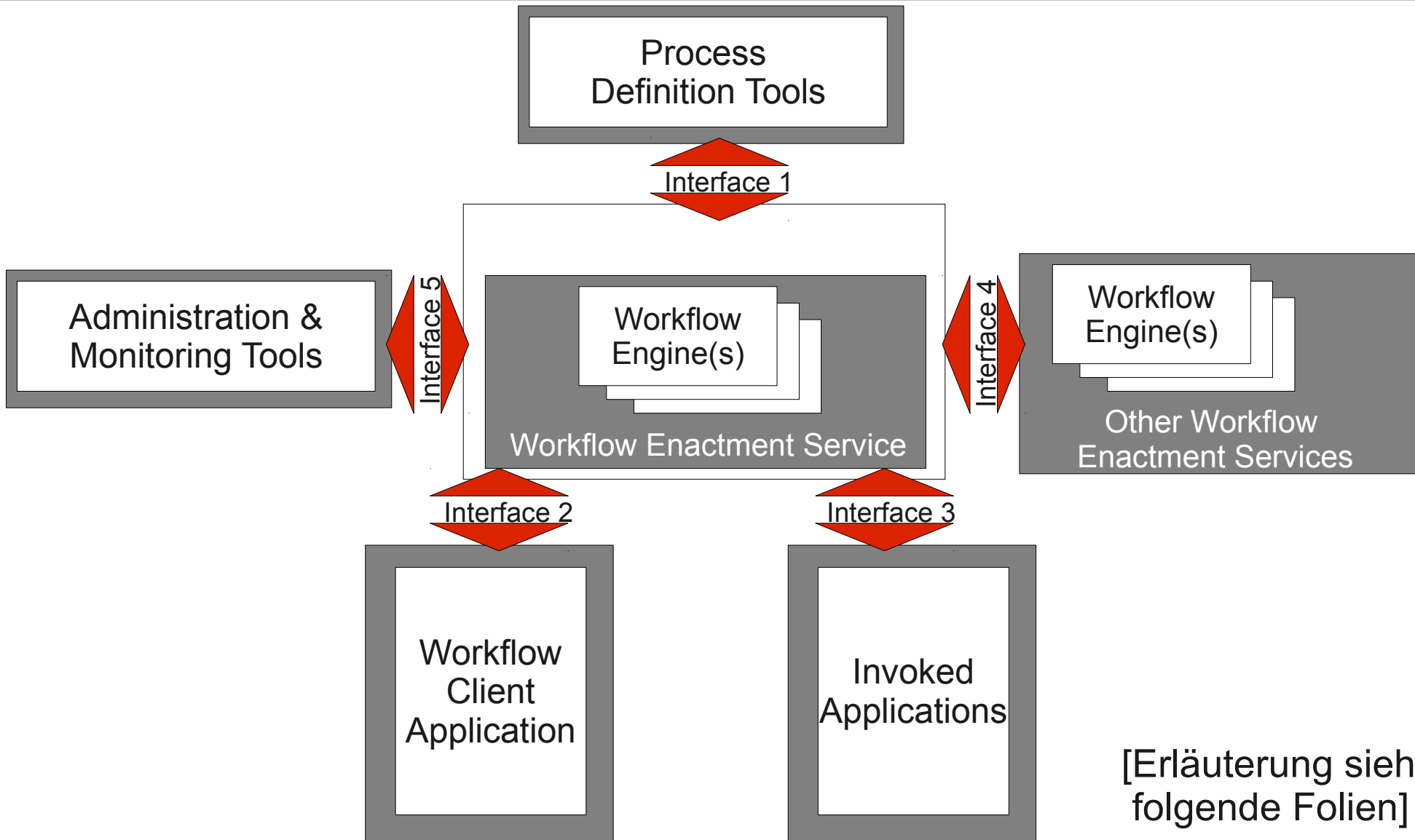
Vorteile Trennung Workflow-Management und Anwendungslogik

- 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 zur Abdeckung der Basisfunktionalität eines WfMS

Referenzmodell der WfMC - Überblick (2)



[Erläuterung siehe
folgende Folien]

„**Workflow Enactment Service**“ (Ausführung):

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

Aufgaben des **Workflow Enactment Service**:

- Initialisieren (Aufruf „**Process Definition Tools**“) und Beenden von Workflowinstanzen
- Planung des Ablaufs („Routing“) von Instanzen
- Zuordnung von Tasks zu Ressourcen (gemäß Ressourcenklassifikation, Teil der „**Process Definition Tools**“)
- Starten von Anwendungen zur Ausführung von Aktivitäten (Aufruf „**Invoked Applications**“)

„Process Definition Tools“

- Prozessdefinitionen im engeren Sinn:
 - Erstellung und Spezifikation von Prozessen: zeitliche (z.B. Sequenz, Parallelität) und konditionale (z.B. Verzweigungen) Abfolge von Komponenten (z.B. Aktivitäten), Constraints, etc.
 - Sprachen: z.B. Petri-Netze, BPMN (Business Process Modeling Notation)
 - Spezifikation von Aktivitäten (z.B. Rollen zur Ausführung, aufzurufende Anwendungen sowie Übergabeparameter)
 - 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 Endbenutzer (Anwender) in Form von sog. Arbeitsliste („worklists“)
- Informationen zu „work items“
- Initiieren und Beenden von Aufgaben
- Übergabe von Zustandsparametern (Aufruf **Workflow Engine**)

„Invoked Applications“

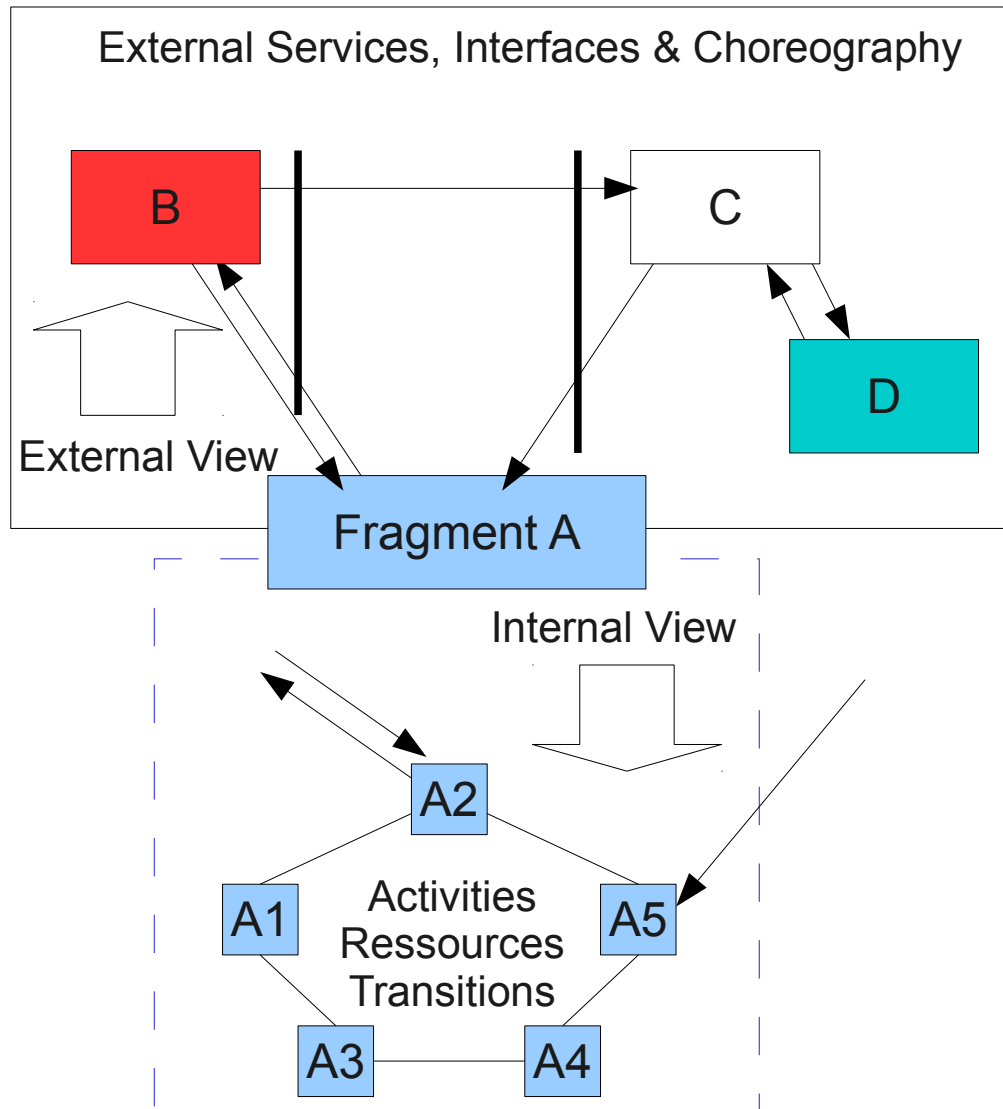
- Anwendungen im eigentlichen Sinne (Start durch Aufruf **Workflow Engine**) sowie Anwendungs-Parameter (Austausch mit **Workflow Engine**);
- Konfigurationsdateien
- Datenzugriff
- Automatisierte sowie interaktive Anwendungen

„Other Workflow Enactment Service(s)“

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

„Administration & Monitoring Tools“

- Operationales Management
 - Verwaltung von Endnutzern (Stammdaten Verfügbarkeiten)
 - Zustandsinformationen einer Prozessinstanz
- Beobachtungs- und Report-Tools
 - Verarbeitung der vergangenheitsbezogenen „Rohdaten“ (Aufruf **Workflow Engine**) zur Berechnung von Performanz-Indikatoren



Externe Sicht:

- Nur Schnittstellen, ansonsten Black Box
- Choreographie

Interne Sicht:

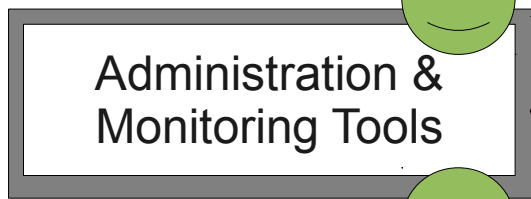
- Spezifikation von Aktivitäten, Ressourcen, Schnittstellen

Einbindung von Rollen am Beispiel des Referenzmodells der WfMC



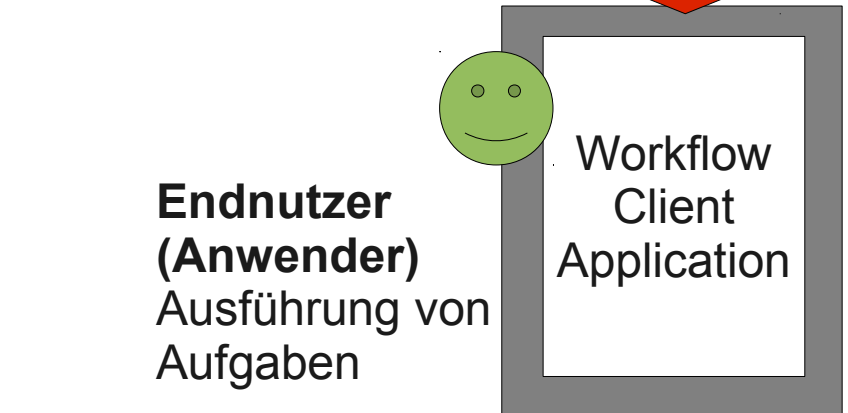
Administrator

- Verwaltung von Ressourcen und Autorisierungen
- Beobachtung (Monitoring)
- Instandhaltung lauffähiges System

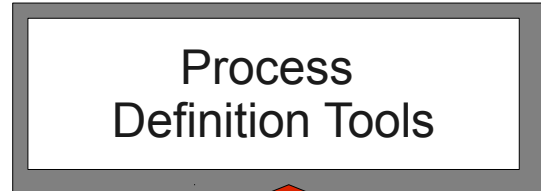


Prozess Analyst

Performanzanalysen

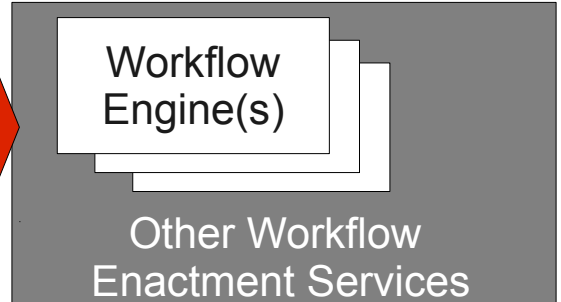
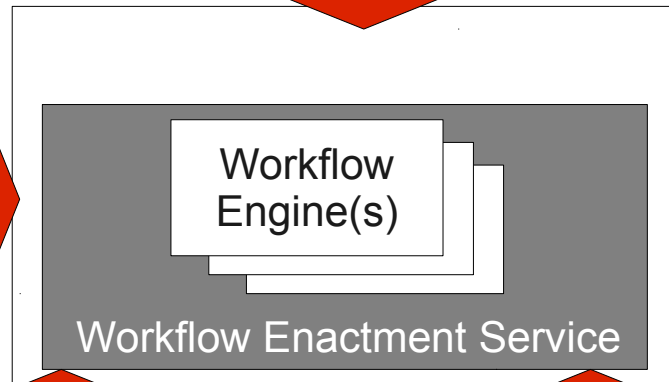


Endnutzer (Anwender)
Ausführung von Aufgaben



Workflow Designer

Workflowstruktur & Ressourcen



Generisches Ziel: Interoperabilität:

- Standards zum Datenaustausch
- Verknüpfung von Ausführungsmaschinen unterschiedlicher Hersteller
- Unabhängigkeit der Anwendungen von konkreten WfMS
- **Schnittstelle 1: Workflow Process Definition Read/Write-Interface**
 - Verbindung von Workflow-Modellierungswerkzeugen mit der zentralen 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 ist die Übergabe eines Teil-Workflows zur Ausführung an ein anderes WfMS.
- **Schnittstelle 5:** Administrations- und Monitoring-Werkzeuge anderer Hersteller zur Information über den aktuellen Ausführungszustand von Workflows und zur Administration laufender Workflows Administration laufender Workflows.

Schnittstellen sind nicht völlig unabhängig voneinander spezifizierbar.

Vorgehensweise: minimale Menge an Funktionen bzw. Funktionalität standardisieren.

- 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



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

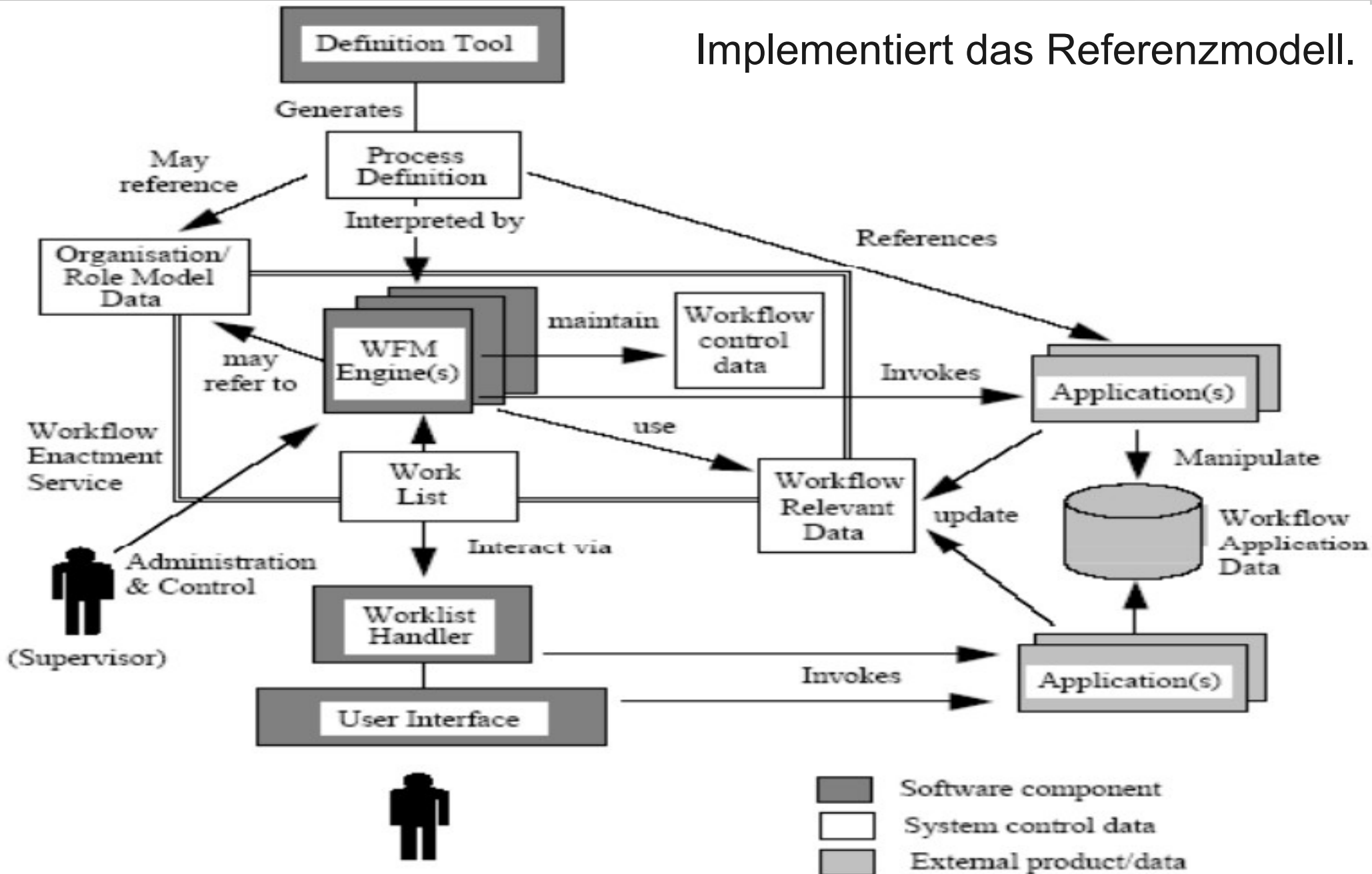
Die Verknüpfung von Workflow Enactment Services findet man in der Praxis selten. Woran könnte das liegen ?

Antwort: Die Prozesse müssen in der Regel über alle Workflow Enactment Services modelliert werden. Das geschieht dann in einer Notation. Dann macht es aber oft Sinn, diese kompletten Prozessmodelle einfach allein in dem passenden WES laufen zu lassen, anstatt sie über mehrere hinweg zu verteilen und damit Heterogenität zu fördern. Gefördert wird dieses Vorgehen oft noch durch die fehlende Interoperabilität der einzelnen WES.

- Datensicht etwas vernachlässigt
- Spezifikation der Schnittstellen „nach außen“
 - Schnittstelle 3: Synchronisation von Anwendungen mit Arbeitslisten und Engine (ACID-Eigenschaften von Transaktionen)
 - 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)

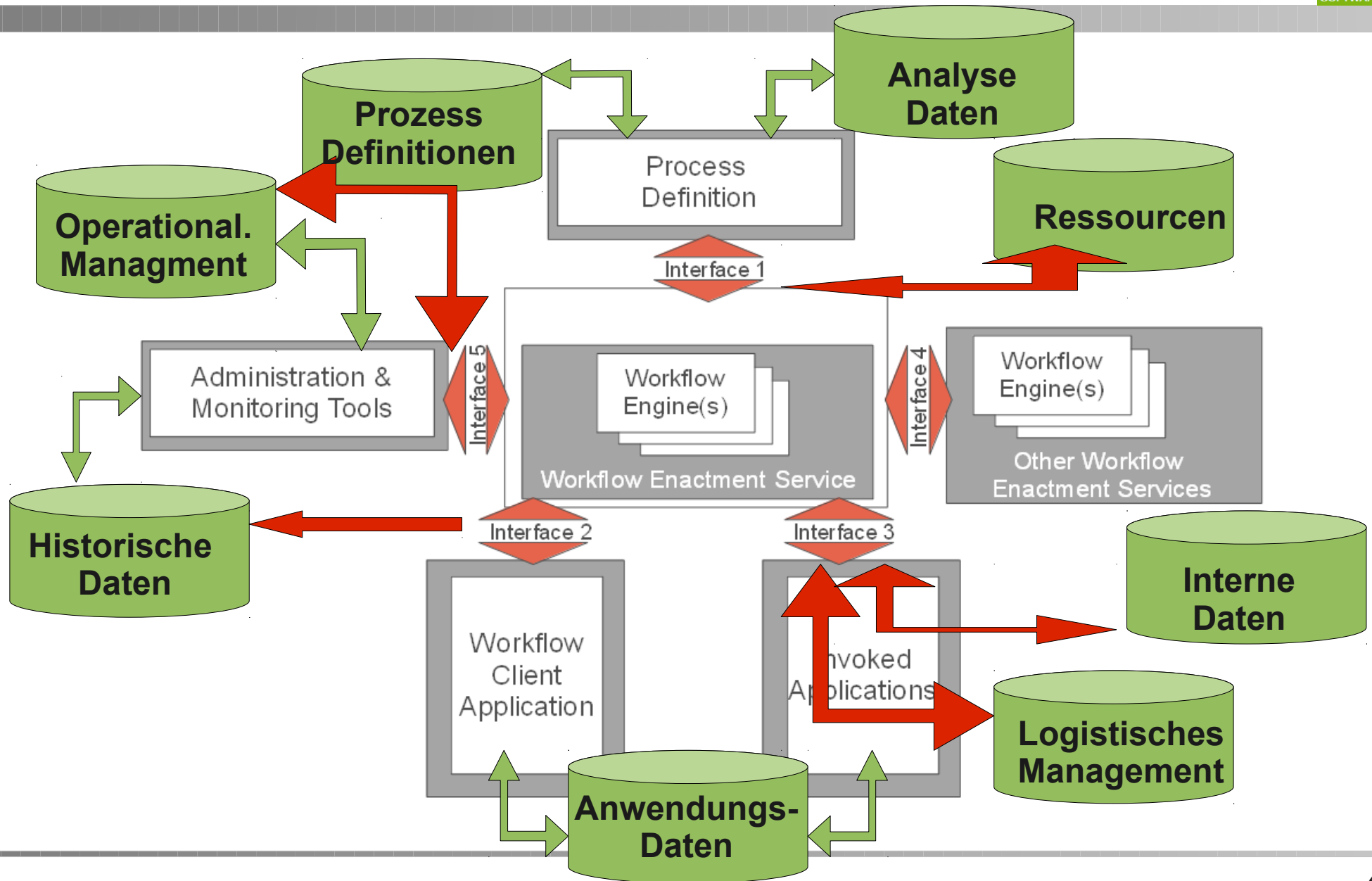
Implementiert das Referenzmodell.

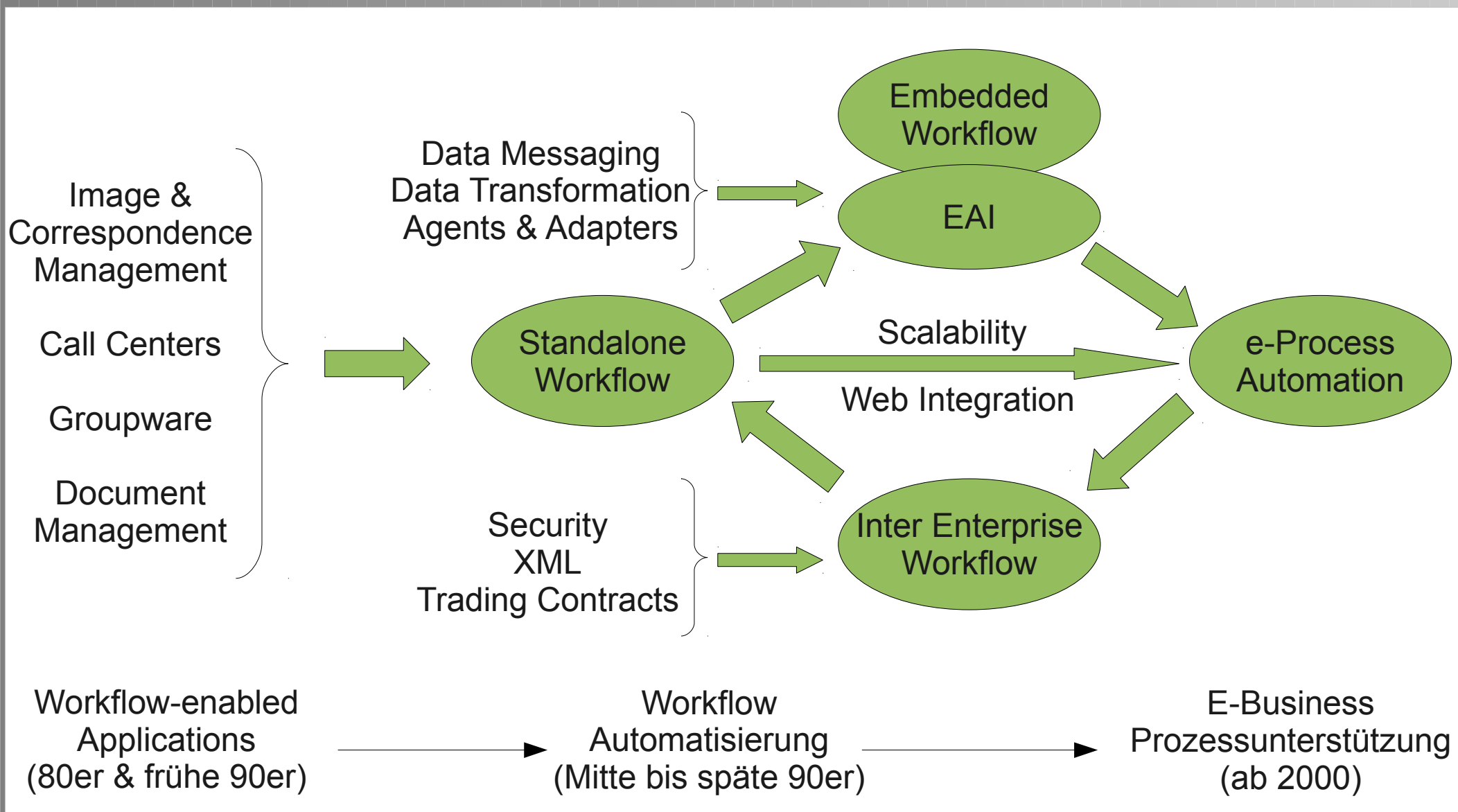


- **Prozess-Definitionen** (Schemata), inkl. Definitionen der Aktivitäten
- **Ressourcen** (Rollen, Organisationen)
- **Analysedaten**
- **Operationales Management** (technische Parameter)
- **Historische Daten**: Nachverfolgung einer Prozessinstanz
- **Anwendungsdaten**
 - Zugriff nur über Applikation,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

[Zuordnung in Referenzmodell siehe nächste Folie.]

Daten am Beispiel des WfMC Referenzmodells



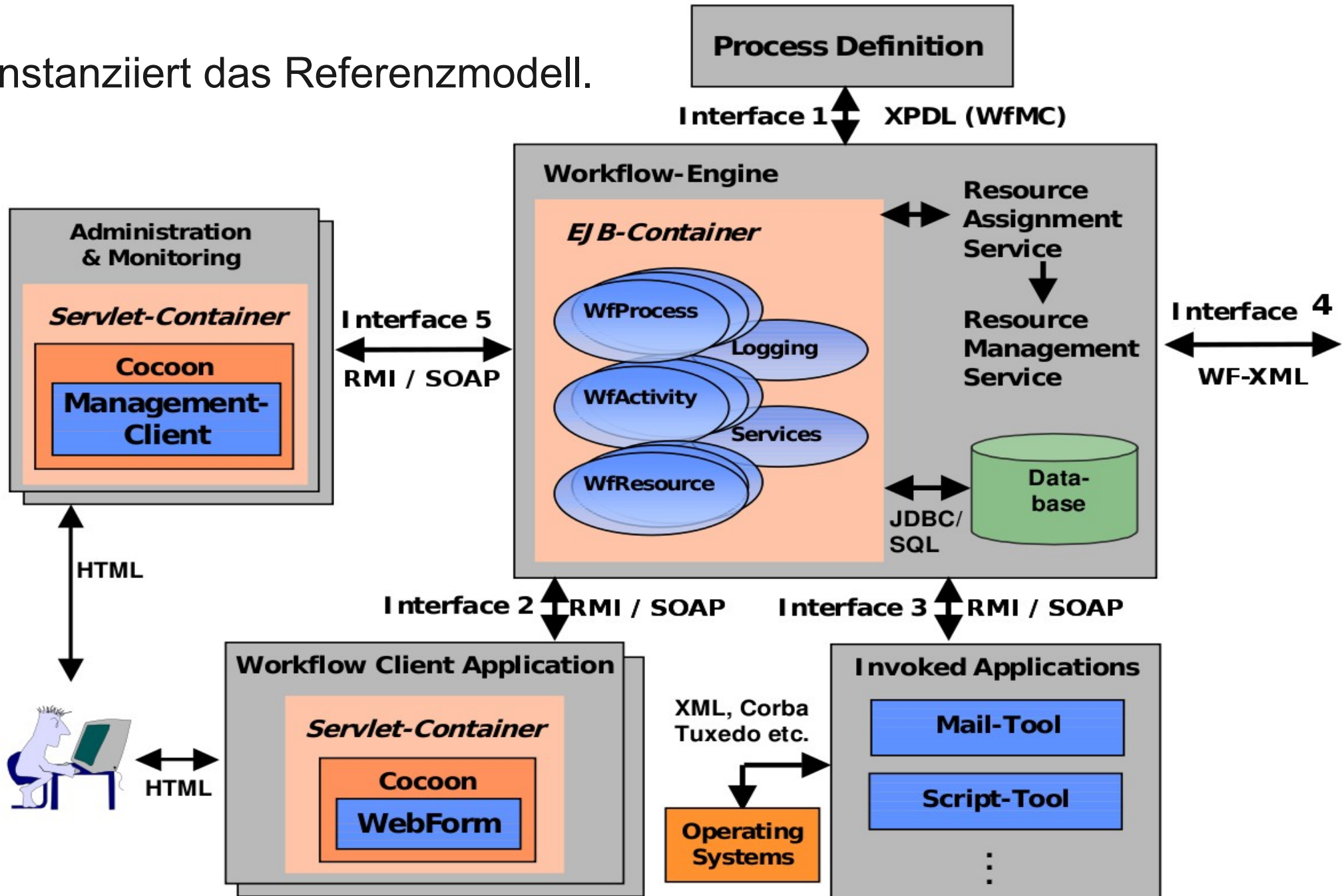


- Open-Source
 - <http://wfmopen.sourceforge.net/>
- J2EE-basiert
 - EJB Container, JBoss
- Beherrscht nativ XML Process Definition Language (XPDL)
 - WfMC-Standard zum Austausch von Prozessdefinition zwischen Werkzeugen
 - Wird aktuell nicht mehr separat weiterentwickelt, sondern an BPMN 2.0 ausgerichtet.
- Stark modulare Architektur, die die WfMC Referenzarchitektur umsetzt.

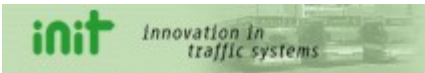

WfMC-Implementierungen

Beispiel WfMOpen

Instanziert das Referenzmodell.



Wird unter anderem eingesetzt von

- STAIRS
 - Structured Audio Information Retrieval System
 - Forschungsprojekt SAP und TU Darmstadt
 - Augmented Working Environments
- IniT 
 - Im Bereich öffentliche Verkehrssysteme
- T Mobile 
 - Unterbau für T-Traffic (digitale Karten für Navis)

- 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 und implementiert Interfaces
 - Bietet Schnittstellen konform zu anderen Standards

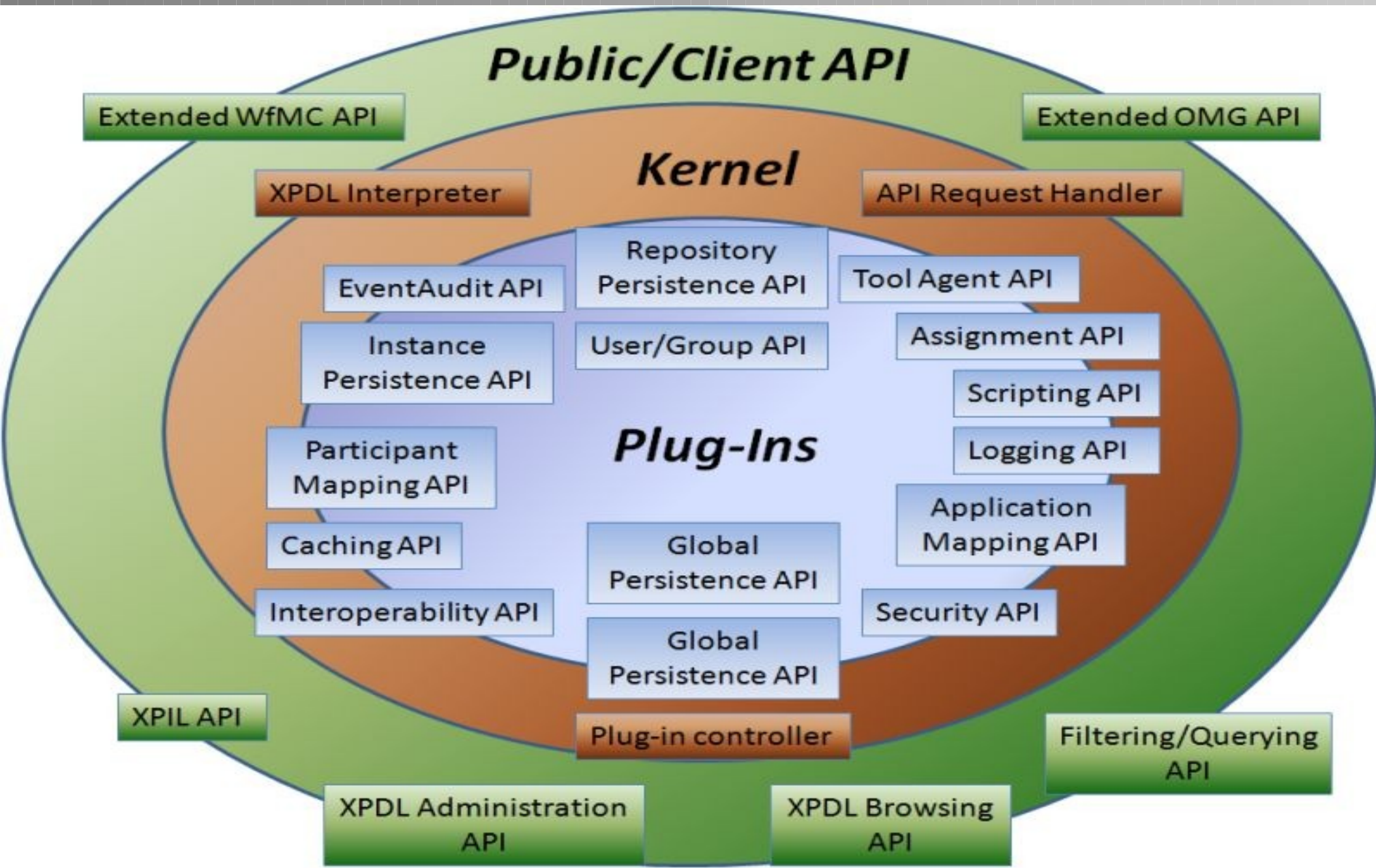
WfMC-Implementierungen

Beispiel Enhydra Shark

Methodische Grundlagen
des Software-Engineering
SS 2012

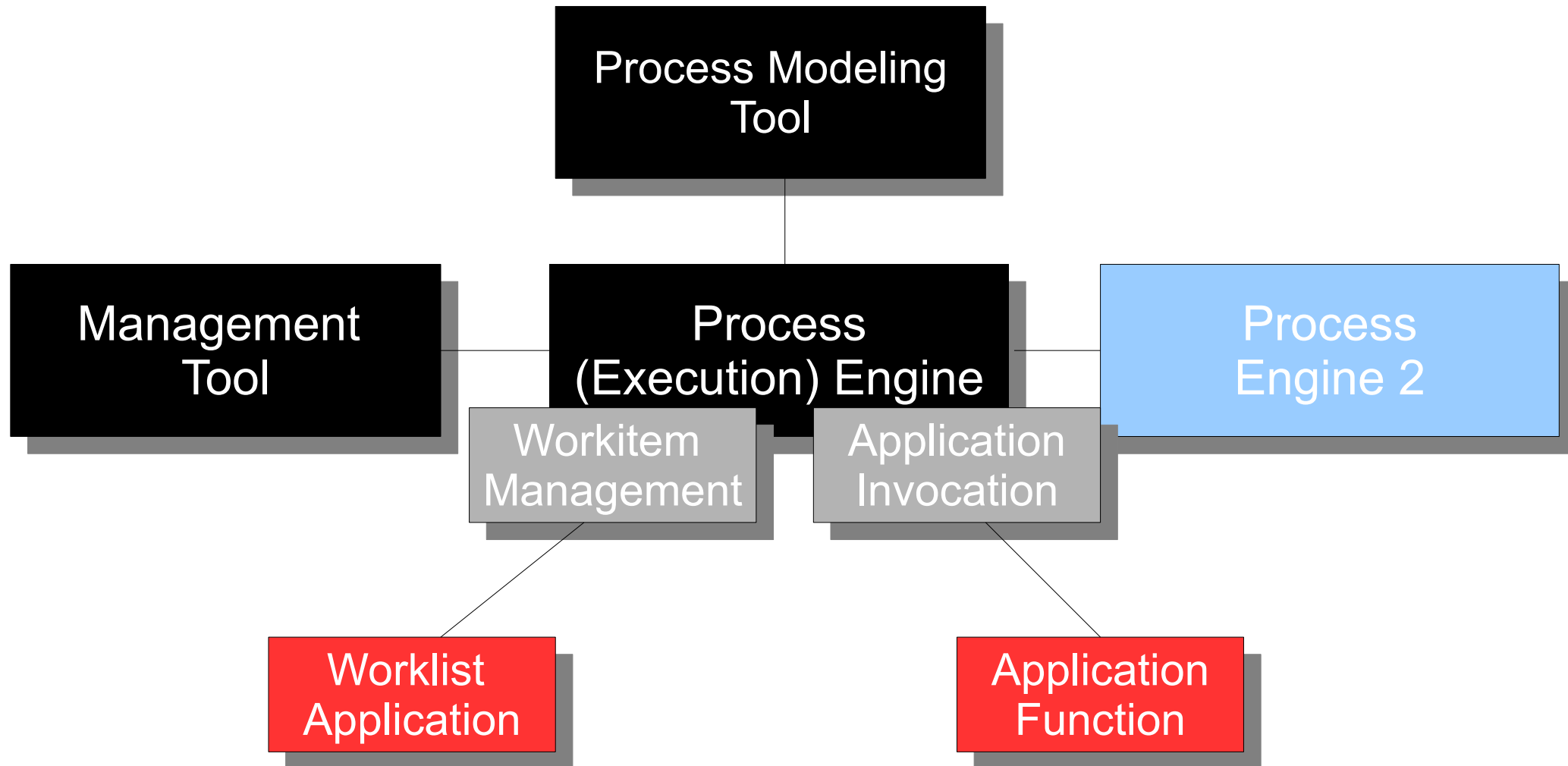


LEHRSTUHL 14
SOFTWARE ENGINEERING

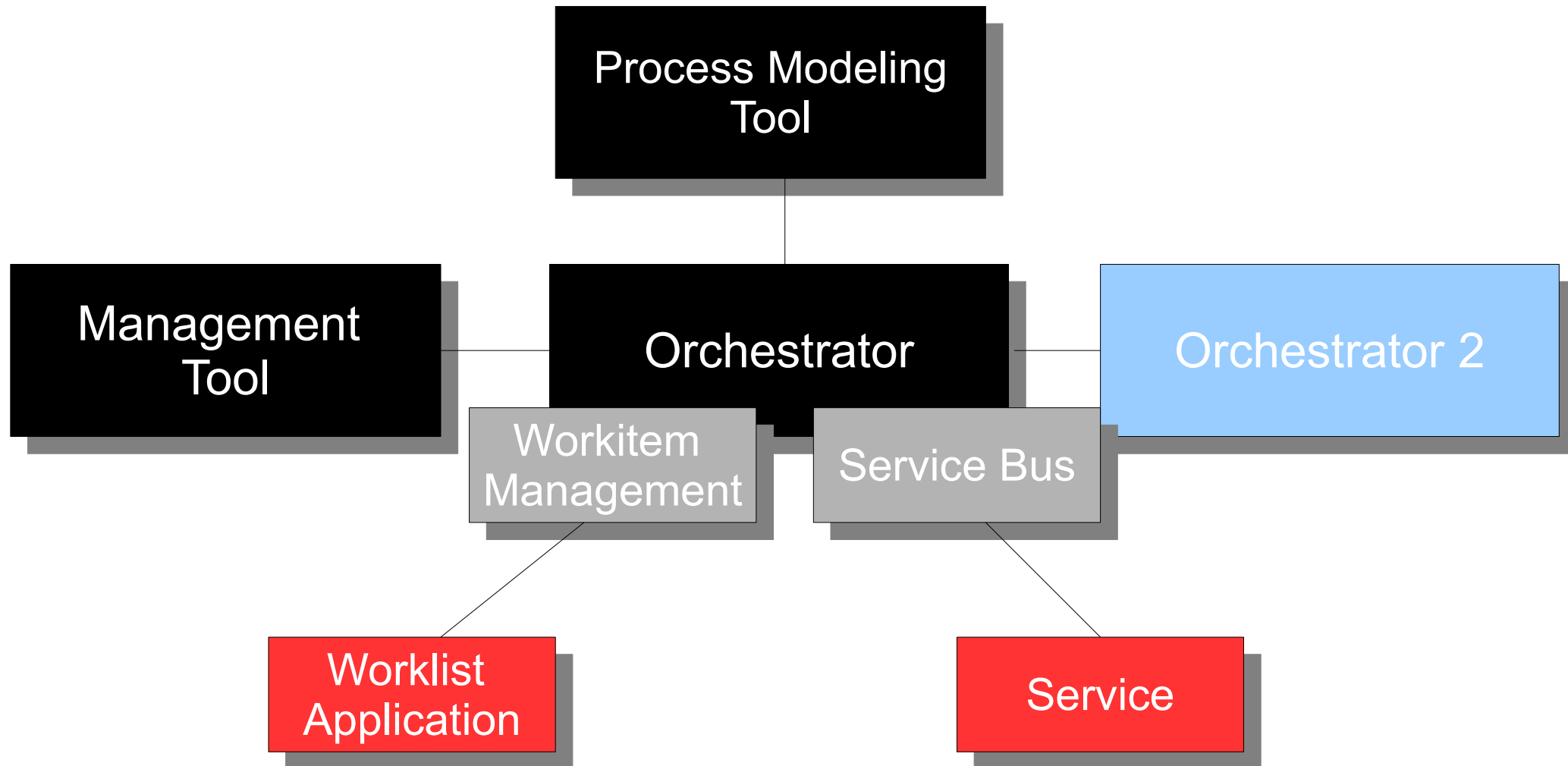


- Es existieren noch weitere Open Source Implementierungen, z.B.:
 - Open Business Engine (<http://obe.sourceforge.net/>)
 - PL/Flow (<http://plflow.sourceforge.net/>)
- Viele kommerzielle Anbieter haben den Conformance-Test der WfMC durchlaufen.
 - Testen allerdings nur die Existenz der Schnittstellen, nicht die eigentliche Architektur.

WfMC Referenzmodell: Moderne Terminologie

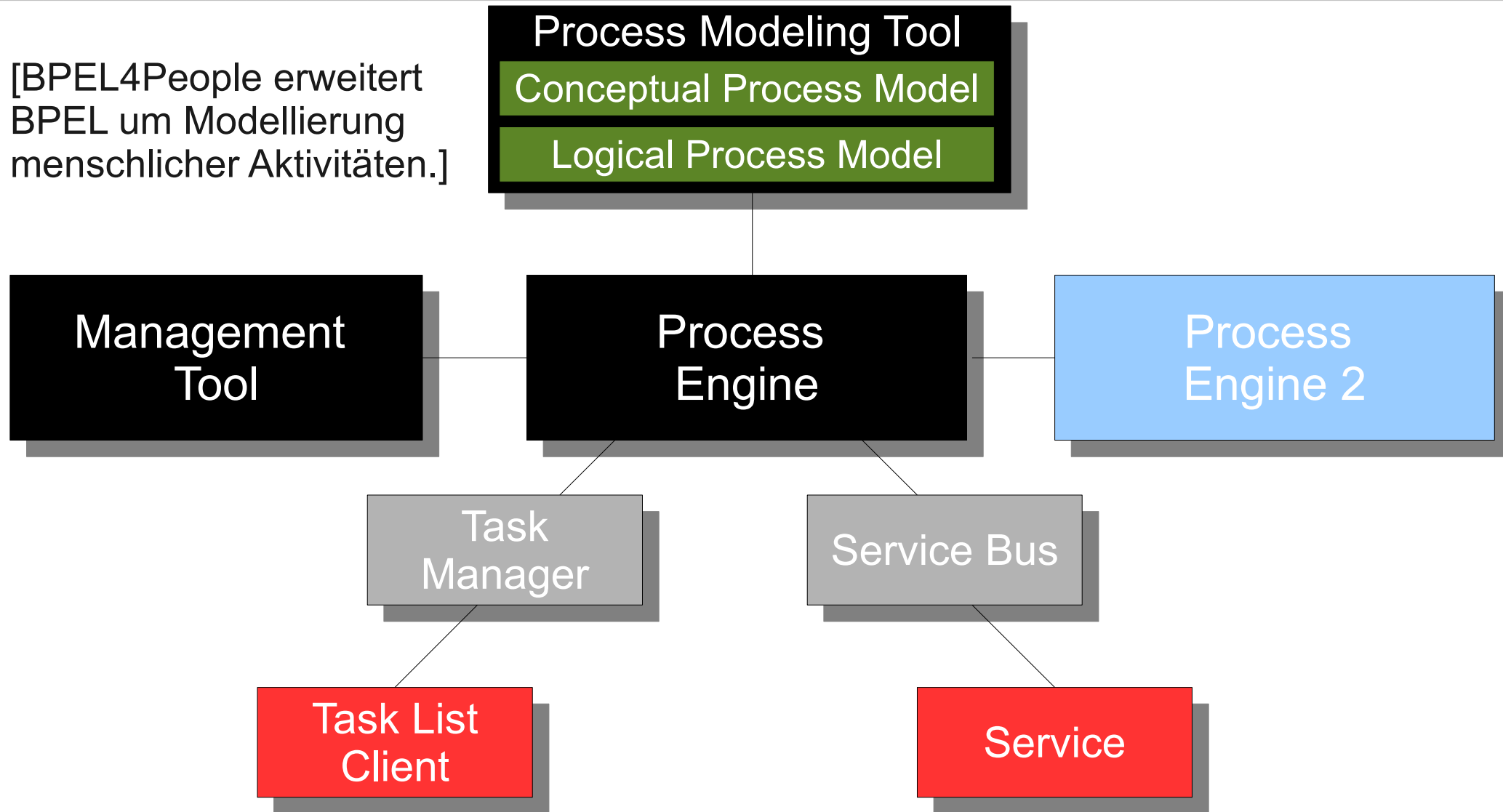


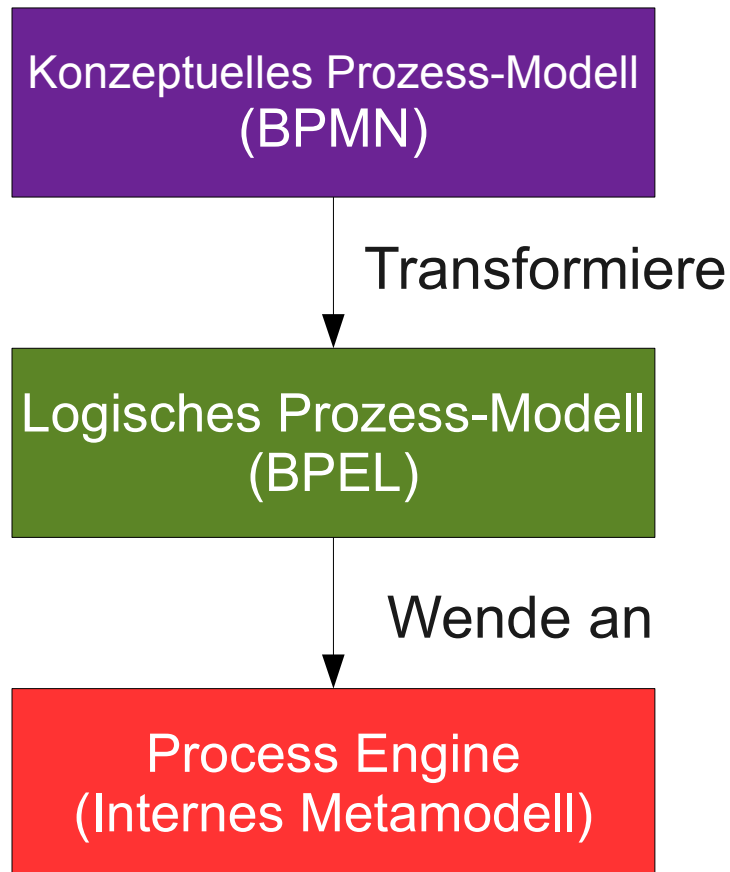
WfMC Referenzmodell: Noch modernere Terminologie



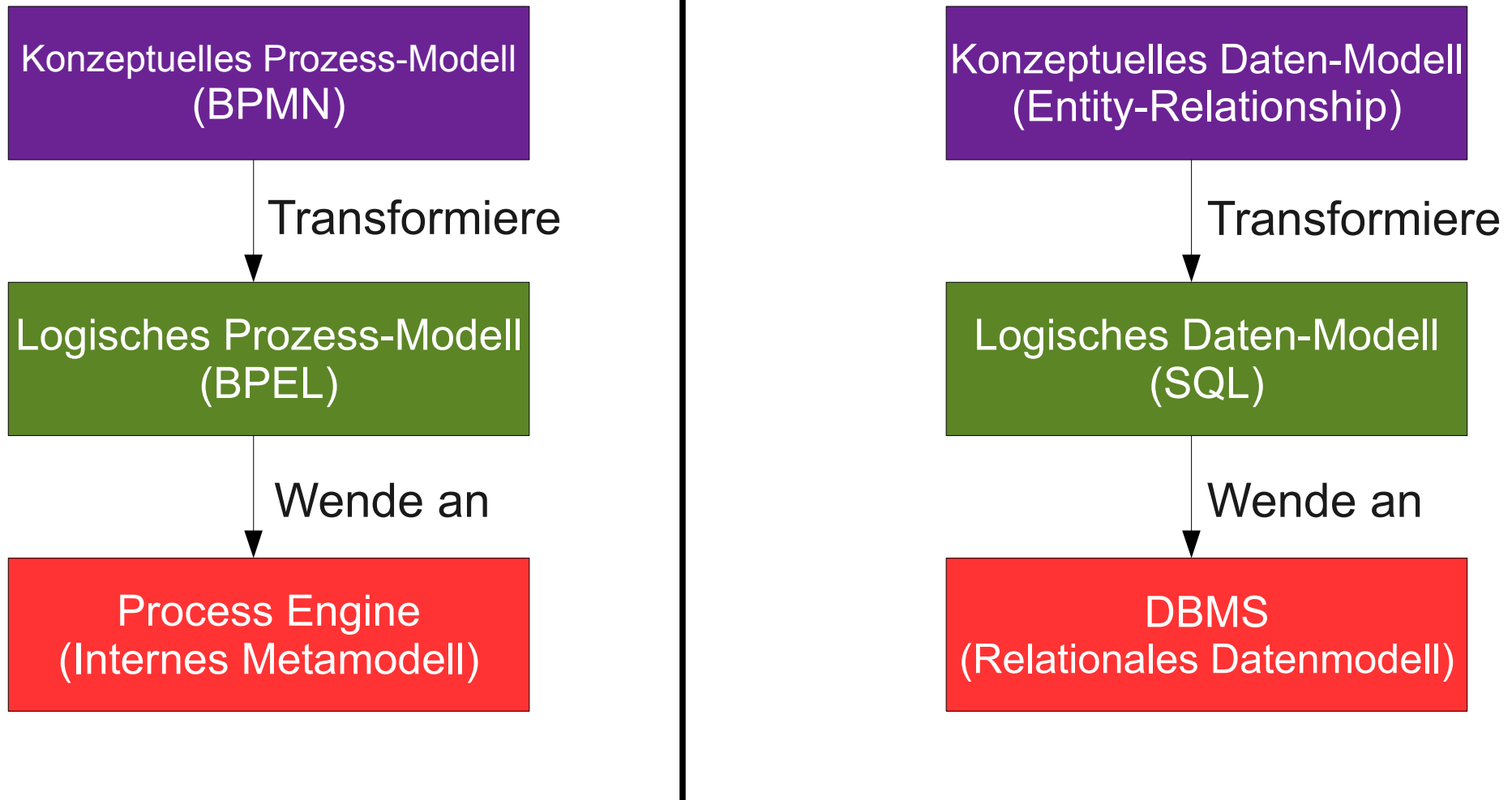
Terminologie im Kontext von BPEL4People

[BPEL4People erweitert BPEL um Modellierung menschlicher Aktivitäten.]





Vergleich: Prozess- vs. Datenmodellierung



In diesem Abschnitt haben wir behandelt:

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

Im nächsten Abschnitt werden wir dann das Thema Workflow-Automatisierung betrachten (im Kontext der Business Process Execution Language (BPEL)).