Vorlesung Wintersemester 2010/11

Workflow-Management-Systeme

Kapitel 12:

Architektur und Implementierung von WfMS

Lehrstuhl für Systeme der Informationsverwaltung, Prof. Böhm
Institut für Programmstrukturen und Datenorganisation (IPD)

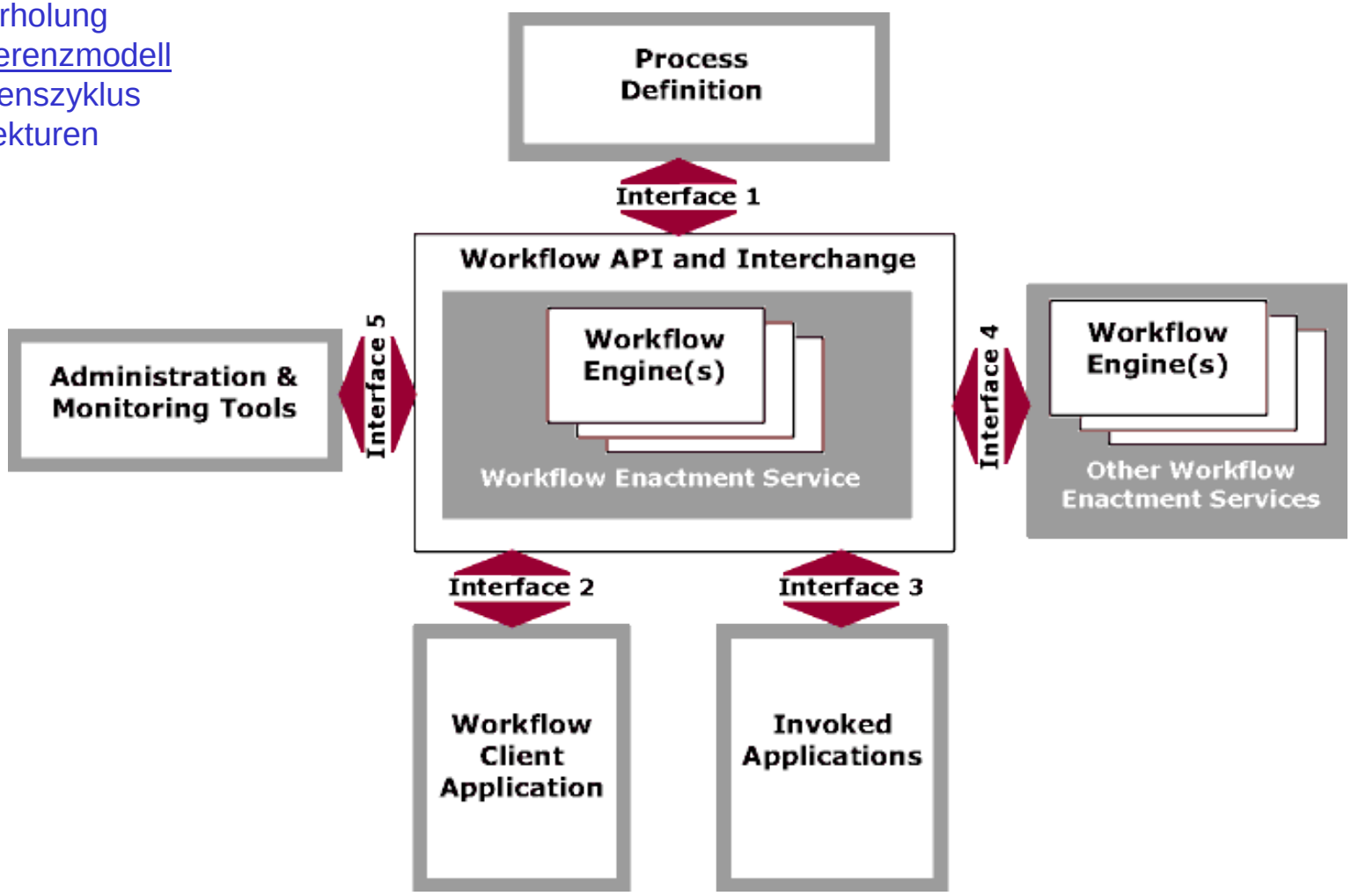
Überblick Kapitel 12 – 1. Teil: Architektur

- Wiederholung Kapitel 2
  - Referenzmodell der WfMC
  - Lebenszyklus von Workflows
- Beispiele für Architekturen
  - MENTOR, Mentor lite
  - YAWL
  - WfMOpen

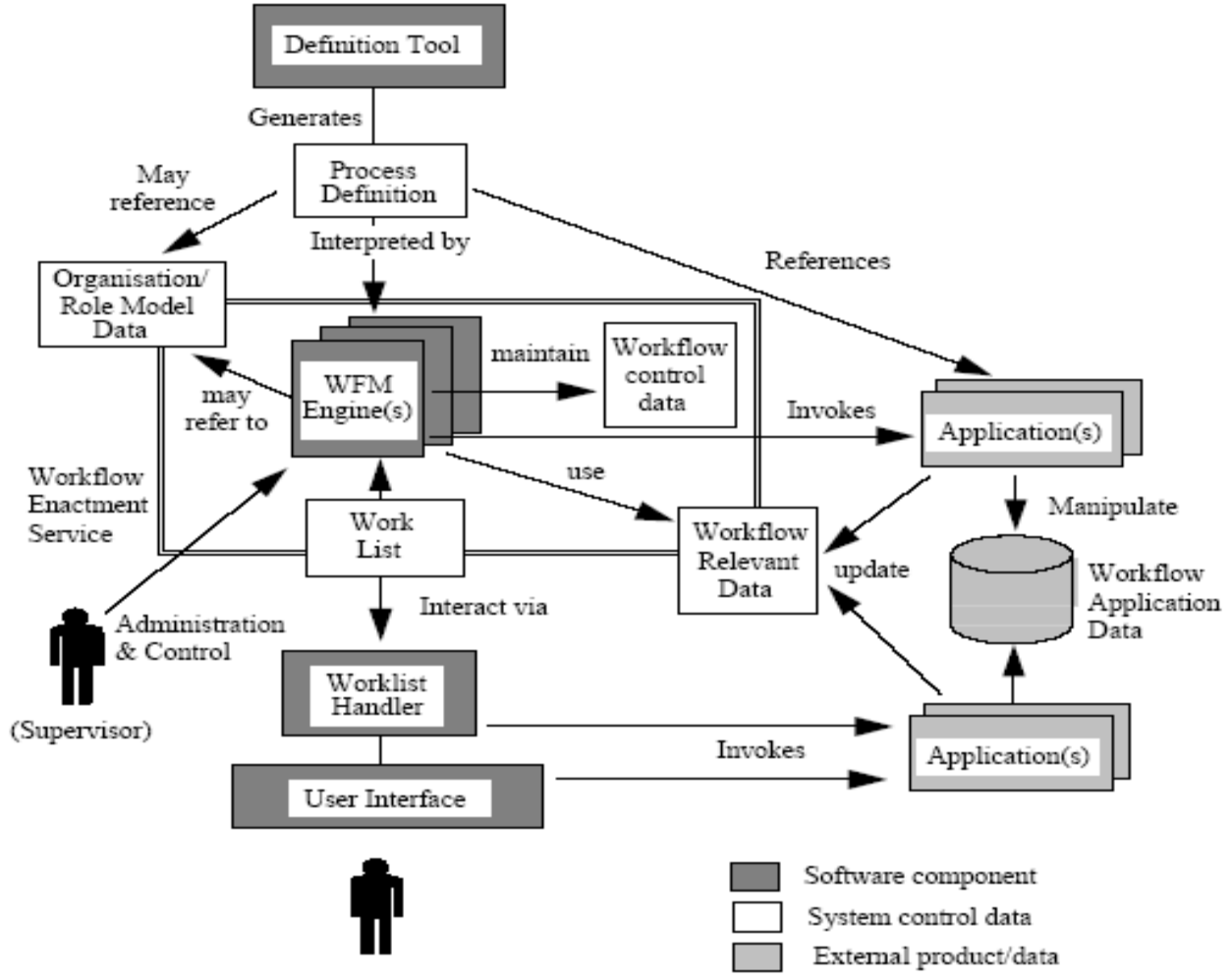
Referenzmodell der WfMC (Wdh. Kap. 2)

Wiederholung
- Referenzmodell
- Lebenszyklus

Architekturen



Generische Architektur nach WfMC (Wdh. Kap. 2)

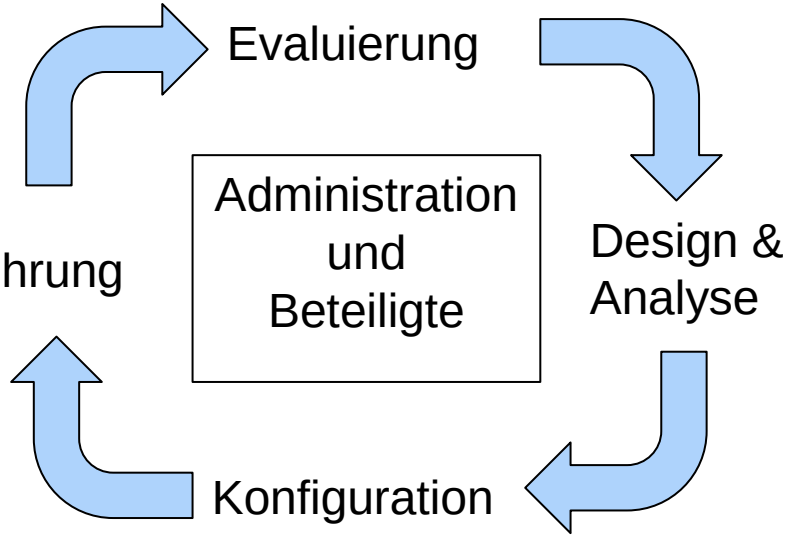


Lebenszyklus von Workflows (Wdh. Kap. 2)

Wiederholung
- Referenzmodell
- Lebenszyklus

Architekturen

Evaluierung:
Workflow „Mining“
Monitoring von Aktivitäten



Ausführung:
Durchführung („Enactment“)
Monitoring
Wartung

Design:
Identifikation & Modellierung von Workflows

Analyse:
Validierung
Simulation
Verifizierung

Konfiguration:
Systemauswahl
Implementierung
Test & Umsetzung („Deployment“)

Quelle: M. Weske, 2007

Implementierungsmodell eines WfMS (Wdh. Kap. 2)

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

Wie kommt man zur Architektur eines Workflow Management Systems? (Wdh. Kap. 2)

- Beim Übergang vom Implementierungsmodell zur Implementierungsarchitektur sind folgende Entscheidungen zu treffen:
  - Werden Module durch Basissysteme (z.B. Betriebssystem, Middle-ware) 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

Anforderungen an verteilte WfMSe (Wdh. Kap. 2)

- Funktionale Anforderungen, die vor allem durch die Workflow-Sprache und ihre Ausdrucksfähigkeit und durch Vorgaben bezüglich der Benutzerschnittstellen bestimmt sind.
- Nichtfunktionale Anforderungen, die vor allem durch das beabsichtigte Einsatzgebiet eines Workflow-Management-Systems beeinflusst werden.

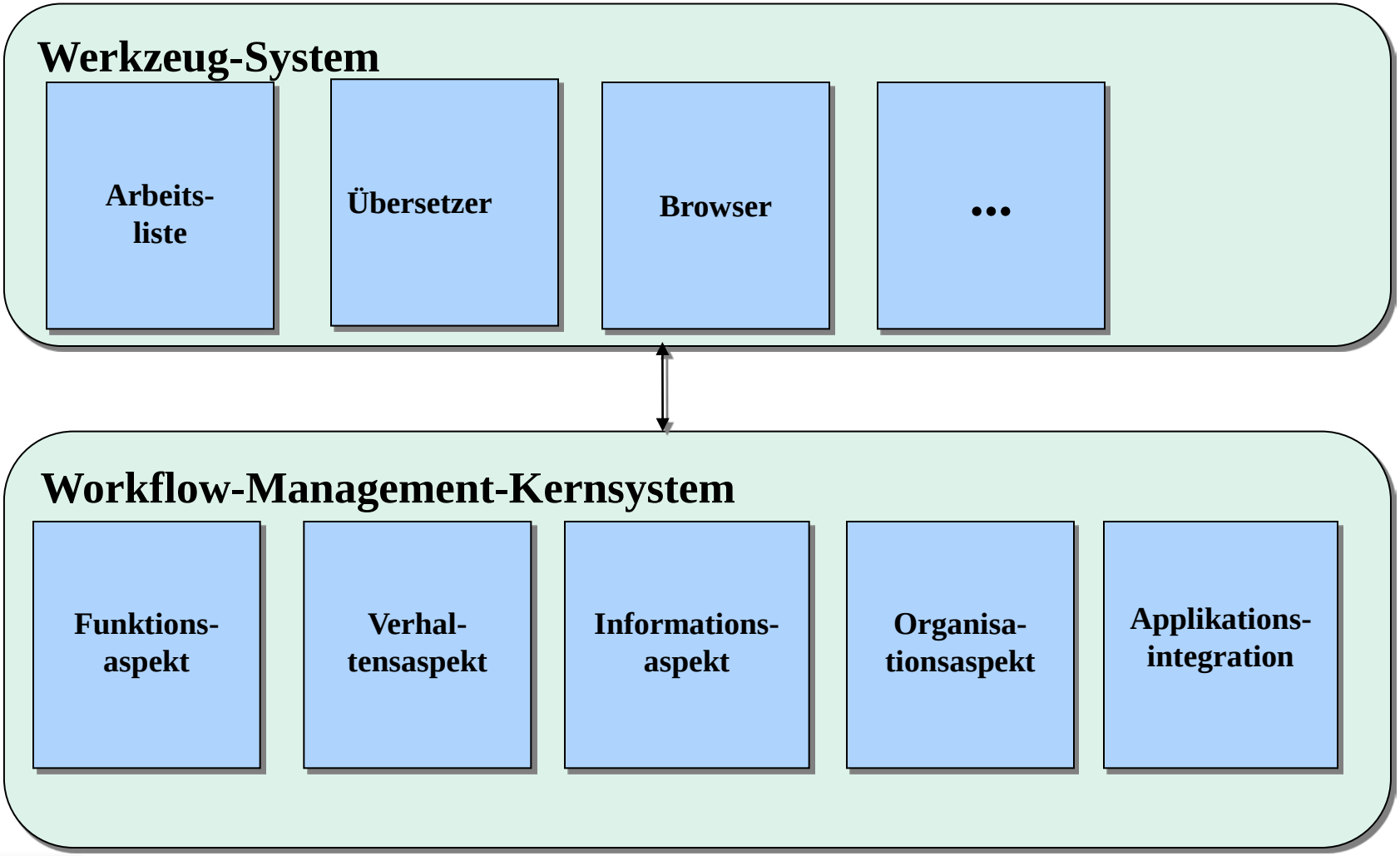
Funktionale Anforderungen (Wdh. Kap. 2)

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

Nichtfunktionale Anforderungen (Wdh. Kap. 2)

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

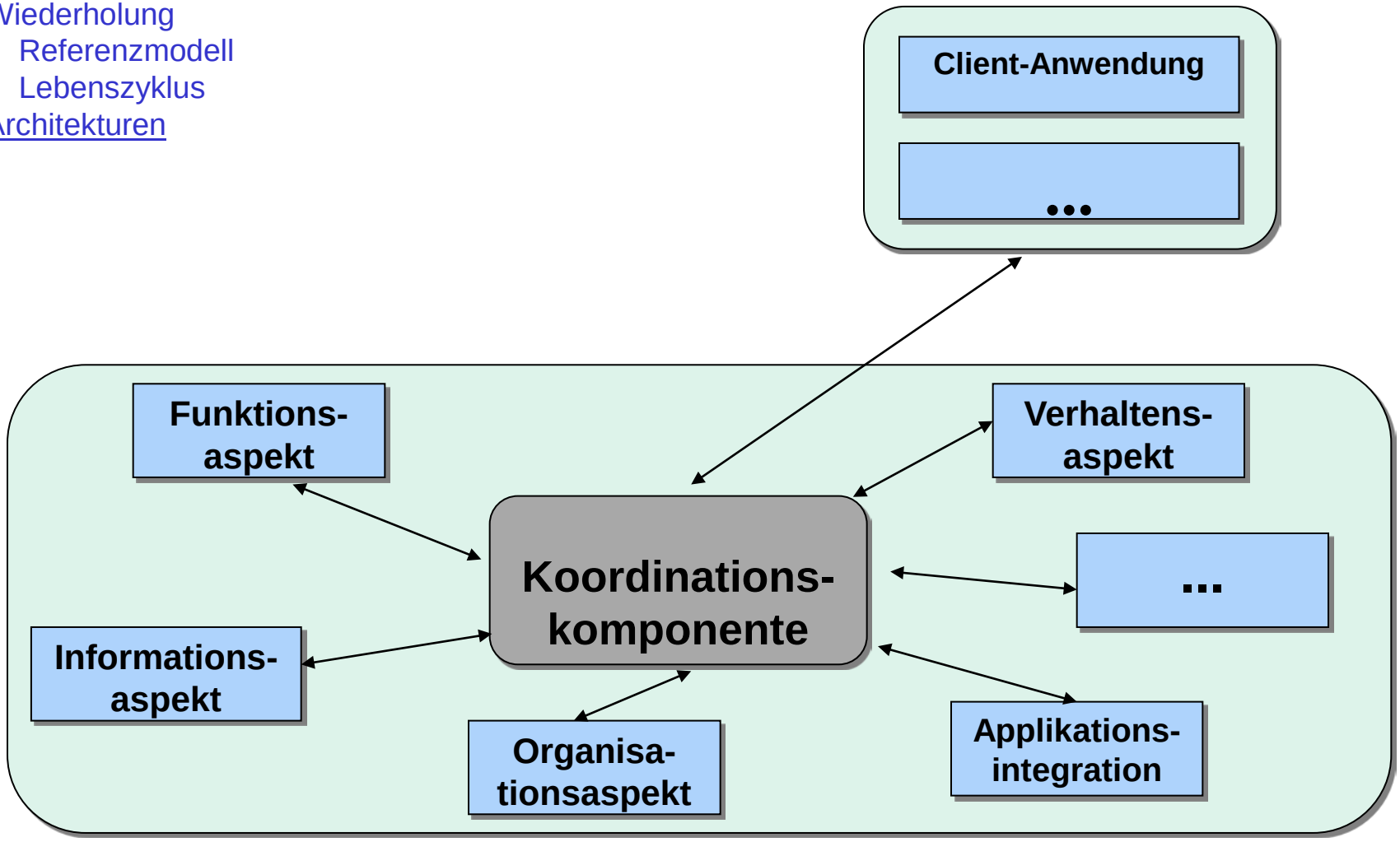
Funktionale Komponenten eines WfMSs (Wdh. Kap. 2)



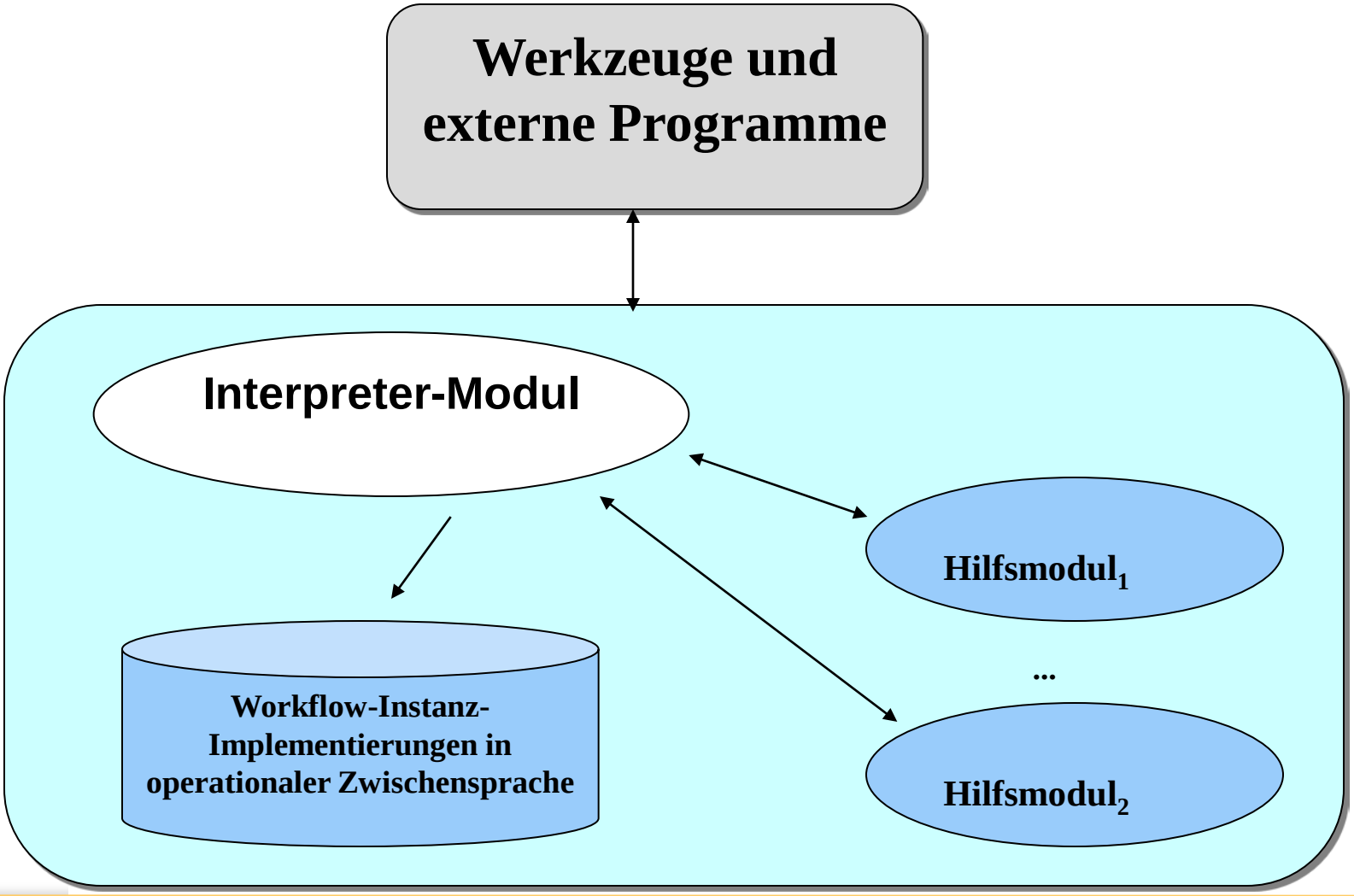
Komponentenarchitektur eines WfMSs (Wdh. Kap. 2)

Wiederholung
- Referenzmodell
- Lebenszyklus

Architekturen



Implementierungsarchitektur mit Workflow-Interpreter



Schichtenarchitektur eines WfMSs

Middleware-Unterstützung für Implementierungs-architekturen

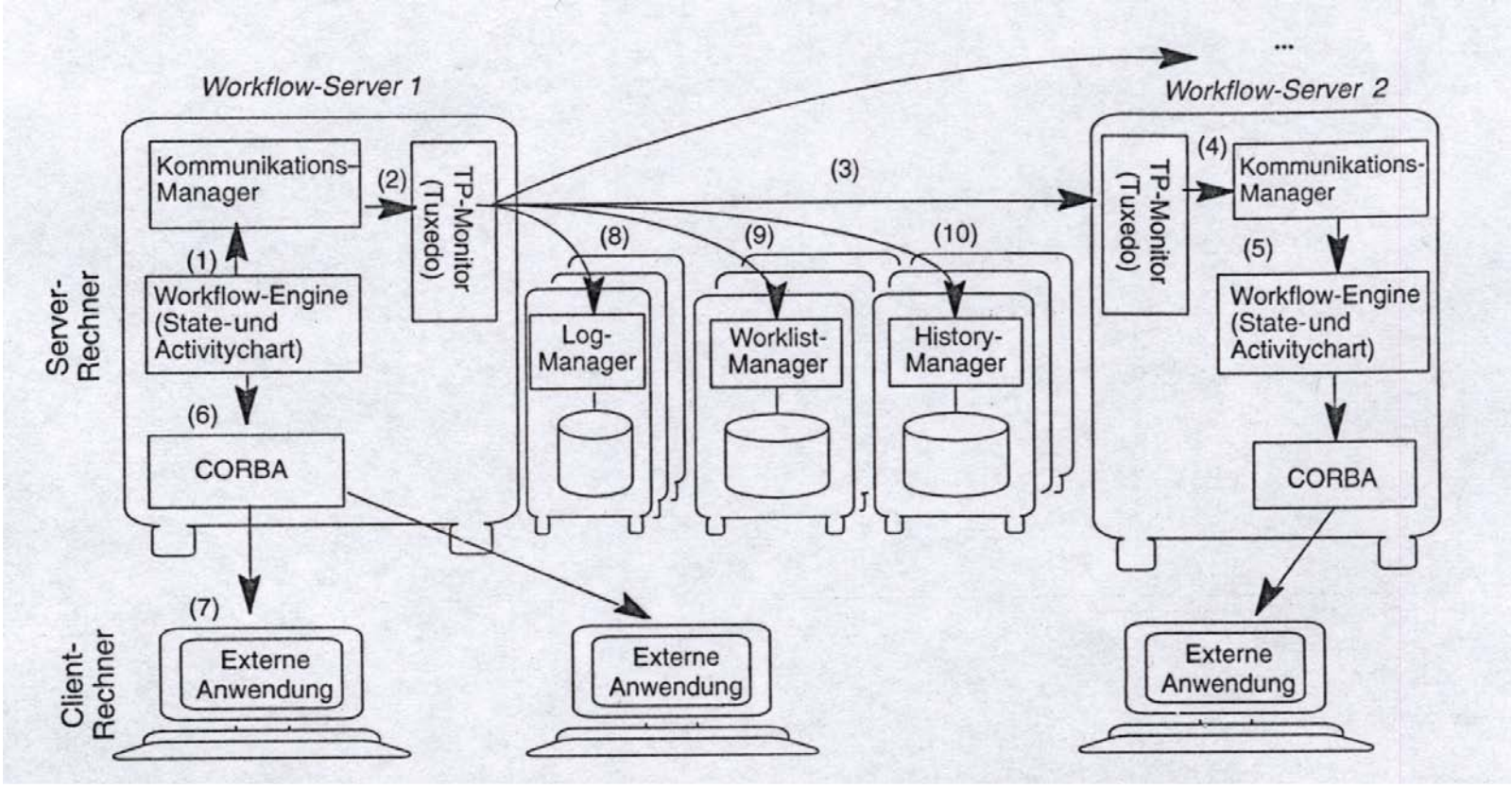
- Middleware als Verteilungsplattform

- Erfüllung nicht funktionaler Eigenschaften stehen im Vordergrund
- Middleware als Basis für die Kommunikation zwischen den Komponenten des verteilten Systems und zur Kommunikation mit Fremdapplikationen hat sich in der Praxis bewährt.
- Verteilung von Servern eines WfMS ist zwar Grundlage aber nicht Lösung für die nichtfunktionalen Anforderungen.
- Komponenten plattformunabhängig und in verschiedenen Programmiersprachen

Mentor Workflow Management System

- Client/Server Architektur
  - Client: externe Applikationen
  - (mehrere) Server:
    - Steuerung des Prozesses; Überwachungs- und Administrations-Prozesse
- Standard-Middleware-Komponenten
  - ORB (Object Request Broker)
  - TP-Monitor
    - Transaktionen, persistenter Speicher
- Orientierung am Referenzmodell der OMG
  - Darüber hinaus: jede Komponente kann als verteilte Komponente realisiert sein

Komponenten der MENTOR Ausführungs-Engine



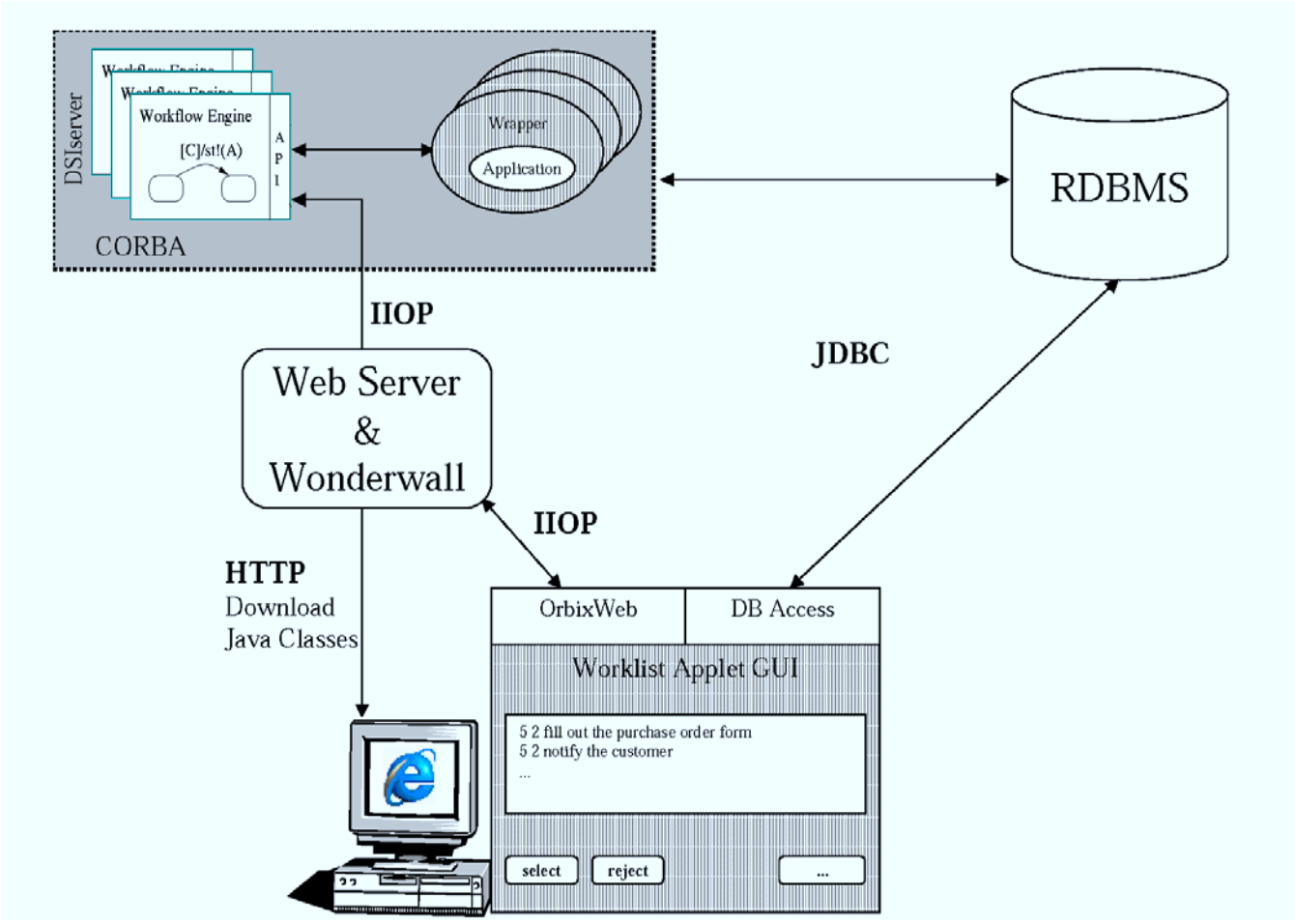
Arbeitsschritte zur Kommunikation von Workflow-Engines in MENTOR (I)

1. Zustandsänderung, z.B. infolge Beendigung einer Aktivität, an lokalen Kommunikationsmanager melden.
2. Kommunikationsmanager kennt Workflow-Server-Rechner, TP-Monitor stellt Primitive zum Starten der Kommunikation zur Verfügung
3. Übertragung der Nachricht an den empfangenden Kommunikations-Manager
4. Ankommen der Nachricht auf Workflow-Server und weiterleiten an lokalen Kommunikationsmanager
5. Übergabe an Workflow-Engine, die die gemeldete Änderung lokal gültig macht.

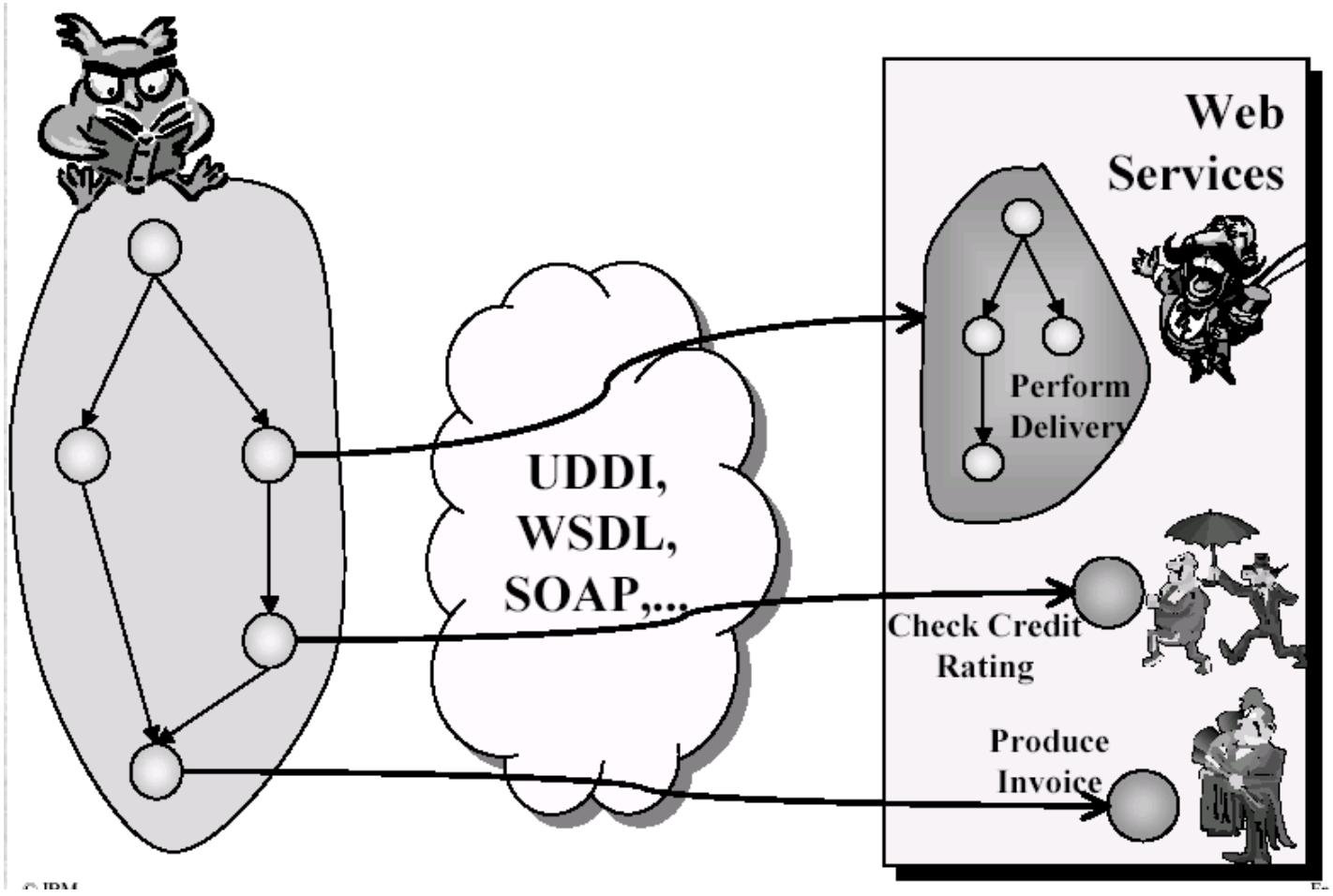
Arbeitsschritte zur Kommunikation von Workflow-Engines in MENTOR (II)

6. Aufruf einer CORBA-Komponente zur Durchführung einer Aktivität
7. Abbildung der Aufruf-Parameter der gestarteten Aktivität auf die Aufruf-Schnittstelle der Applikation
8. Protokollieren der Zustandsänderungen jeder Workflow-Engine durch Log-Manager
9. Worklist-Manager ermittelt geeignetes Ausführungsorgan (Ressource) unter all den Rolleninhabern, die die Aktivität ausführen dürfen
10. History-Manager protokolliert für Anfragen an die Workflow-Historie

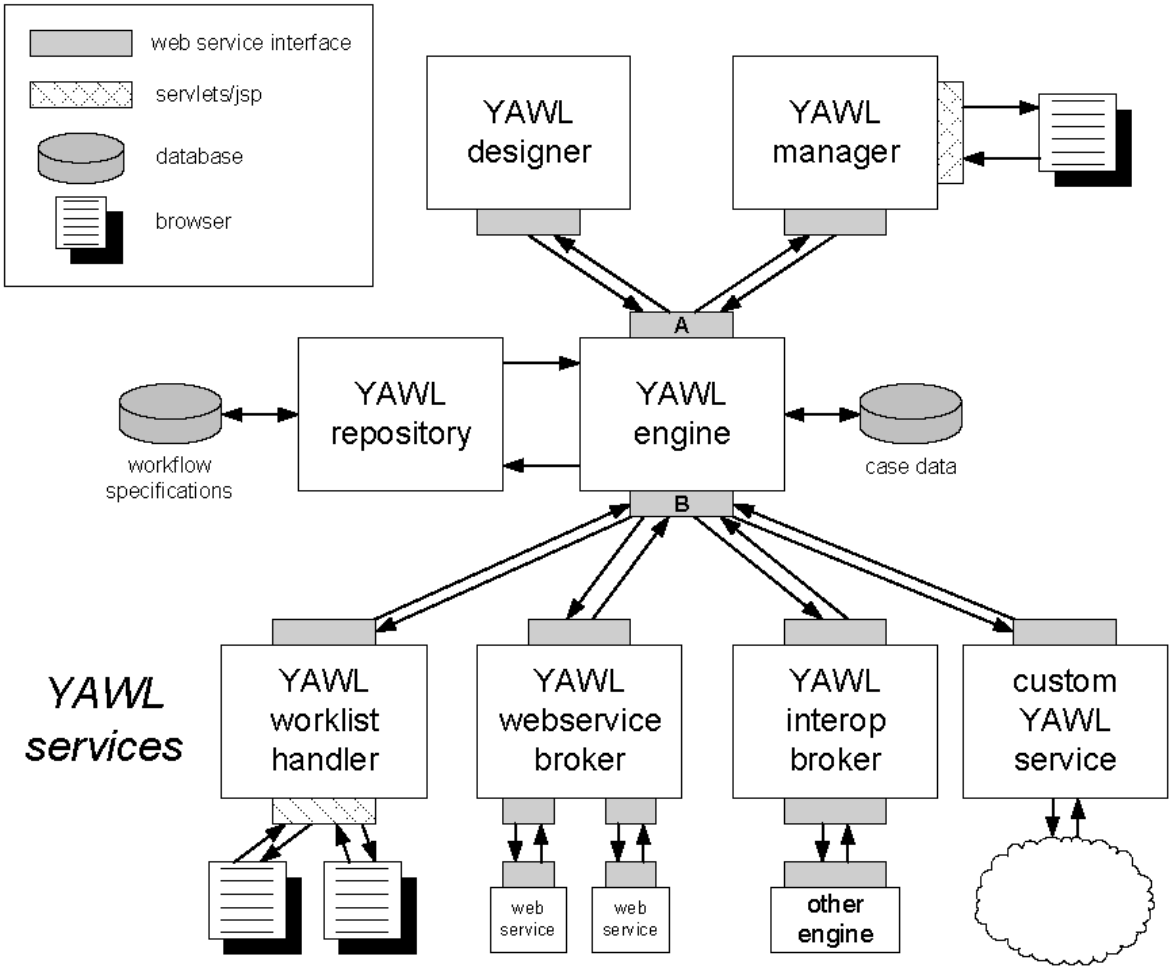
Mentor lite


DSIserver
Workflow Engine
[C]/st!(A)
API
CORBA
Wrapper
Application
RDBMS
IIOP
JDBC
Web Server
&
Wonderwall
IIOP
HTTP
Download
Java Classes
OrbixWeb
DB Access
Worklist Applet GUI
5 2 fill out the purchase order form
5 2 notify the customer
...
select
reject
...

SOA in Aktion (Wdh. Kap. 6)



Beispiel YAWL Engine (Wdh. Kap. 5)



Komponenten in YAWL (I)

- Workflow-Spezifikationen werden mit dem *YAWL Designer* entwickelt
- *YAWL Engine* instantiiert und führt Workflows, die „deployed“ sind, aus
- *YAWL Repository* dient zur Speicherung von ausführbaren Workflows
- *YAWL Manager* ermöglicht das manuelle Verwalten von Workflow-Instanzen oder Workflow Spezifikationen sowie das Verwalten von Daten über beendete Instanzen

Komponenten in YAWL (II)

- Services in YAWL
  - *YAWL Worklist Handler*
    - Zuordung von Work Items zu Personen
    - Besonderheit: im Ggs. zu traditionellen WfMS Entkopplung von der Engine
  - *YAWL Web Services Broker*
    - Mediator zwischen Engine und Web Services
    - Grundgedanke, dass Web Services zu generisch sind, als dass sie direkt eingebunden werden können
  - *YAWL Interoperability Broker*
    - Verknüpfung mit anderen Engines
  - Custom *YAWL Services*
    - Erweiterbarkeit von Services (z.B. Telefondienste, Druckerdienste, etc.)

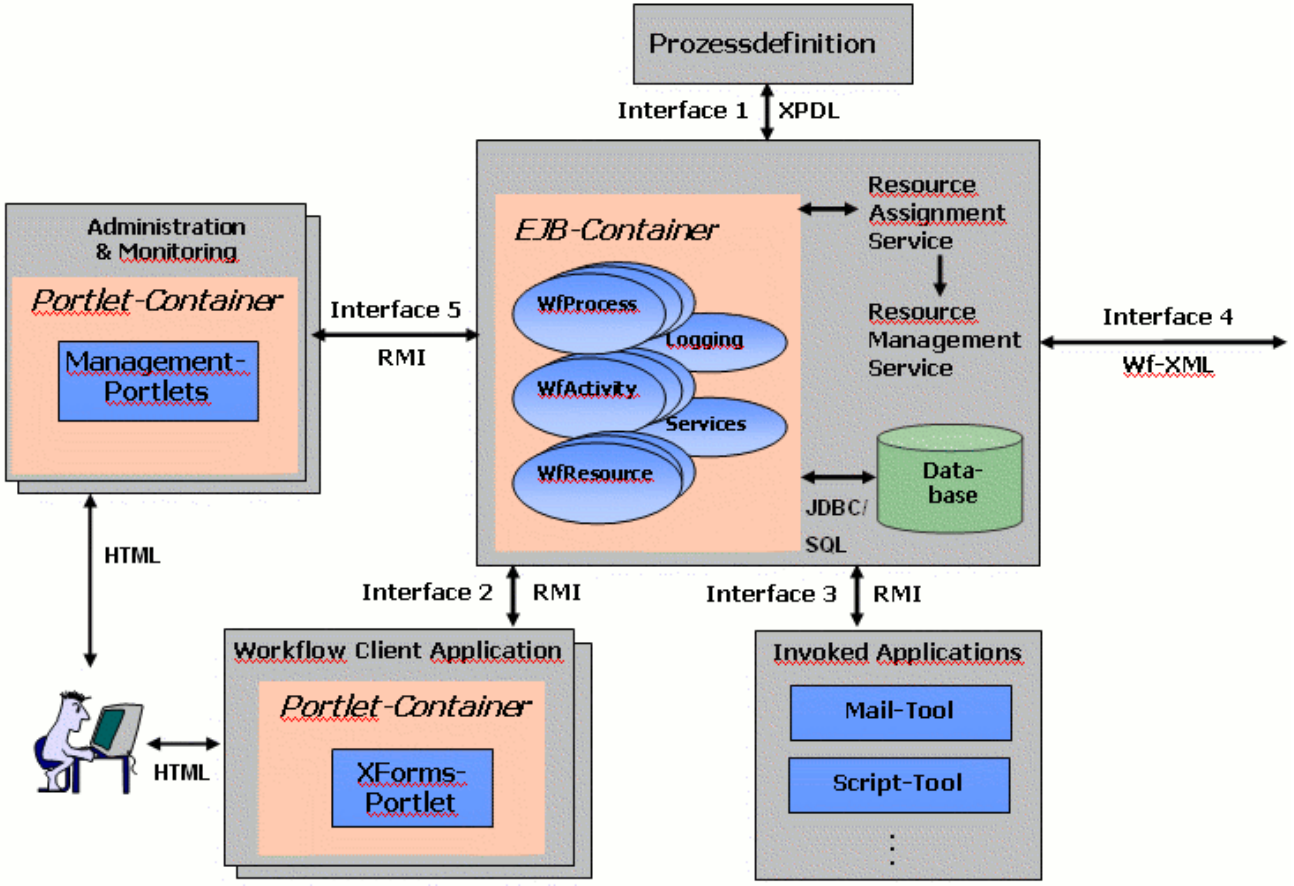
Anmerkungen zu YAWL

- Schnittstellen des WfMC Referenzmodells sind abgebildet
- Umsetzung der Sprache YAWL (Kap.5) mit den Besonderheiten
  - OR-Join
  - Multiple Instanzen
  - Entfernen von Token
- Implementierung der Engine
  - Java
  - XML-Standards (XPath, XQuery, XML Schema)
    - Umsetzung des Datenaspektes (siehe Kap. 5)

WfMOpen

- Benutzt XPDL (XML Process Definition Language (WfMC) zur Prozessdefiniton (Modellierung)
- Java-Interfaces
- Java-Platform Enterprise Edition
- Open Source

WfMOpen


Prozessdefinition
Interface 1
XPDL
Administration & Monitoring
Portlet-Container
Management-Portlets
Interface 5
RMI
EJB-Container
WfProcess
Logging
WfActivity
Services
WfResource
Resource Assignment Service
Resource Management Service
Interface 4
Wf-XML
Data-base
JDBC/ SQL
HTML
Interface 2
RMI
Interface 3
RMI
Workflow Client Application
Portlet-Container
XForms-Portlet
HTML
Invoked Applications
Mail-Tool
Script-Tool

Exemplarische Fragen zu Kapitel 12 – 1. Teil

- Beschreiben Sie die Architektur von MENTOR.
- Beschreiben Sie die Komponenten von YAWL.
- Wofür werden DBMS in WfMS benötigt?

Ergänzende Literatur zu Kapitel 12 – 1. Teil

- Zu Architekturen (generell)
  - S. Jablonski, M. Böhm, W. Schulze (Hrsg.): Workflow-Management - Entwicklung von Anwendungen und Systemen. dpunkt-Verlag, Heidelberg, 1997 (Kap. 14)
- Zu MENTOR
  - S. Jablonski, M. Böhm, W. Schulze (Hrsg.): Workflow-Management - Entwicklung von Anwendungen und Systemen. dpunkt-Verlag, Heidelberg, 1997 (Kap. 15.1)
- Mentor lite
  - German Shegalov, Michael Gillmann, Gerhard Weikum: XML-enabled Workflow Management for E-Services across Heterogeneous Platforms. In: The International Journal on Very Large Data Bases, Volume 10 Issue 1, August 2001
- YAWL System
  - Design and implementation of the YAWL-system. Proc. of CAISE, 2004