

Einführung in z/OS und OS/390

**Dr. rer. nat. Paul Herrmann
Prof. Dr.-Ing. Wilhelm G. Spruth**

WS 2006/2007

Teil 13

z/OS Internet Integration

Java Ausprägungen

Java Script

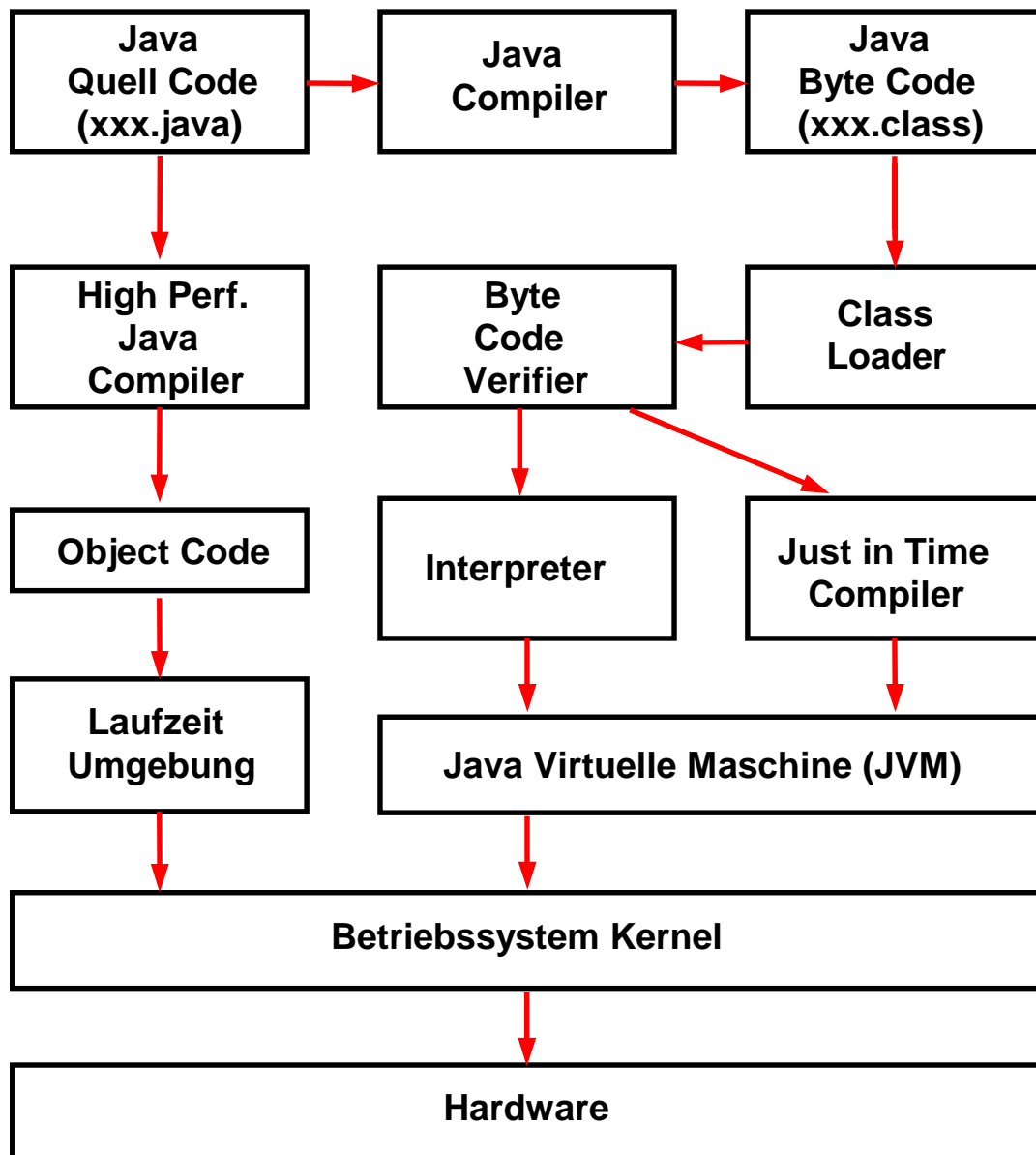
Java Applet

Java Servlet

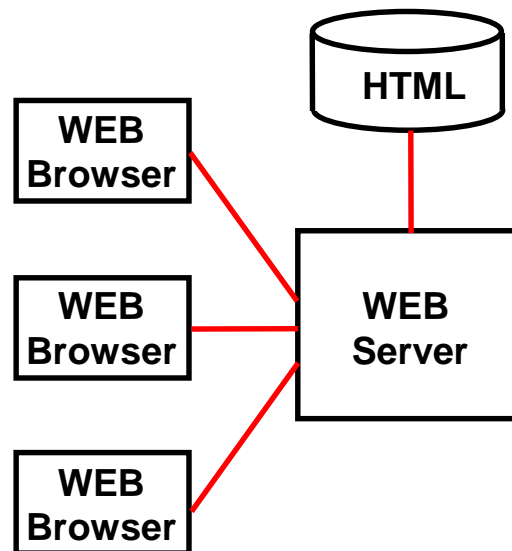
Java Server Pages

Java Bean

Enterprise Java Bean



Java Virtuelle Maschine

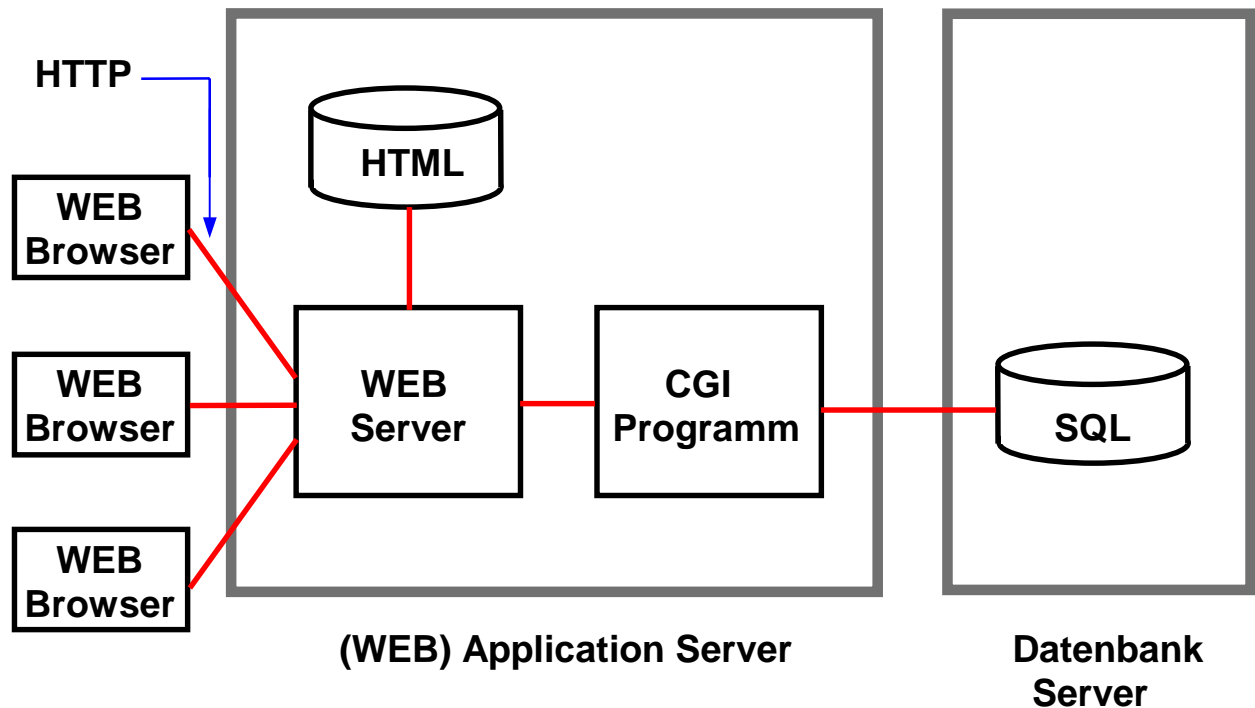


Java und der Web Browser

HTML Seiten werden zum Web Browser mit Hilfe des HTTP Protokolls übertragen.

HTTP wird auch als „Web RPC“ betrachtet. Wie der eigentliche RPC ist HTTP zustandslos: Request/Response. Keine Session.

Erlaubt die Übertragung selbstbeschreibender Daten. Bei jeder Verbindungsaufnahme müssen Datenformate neu ausgehandelt werden.



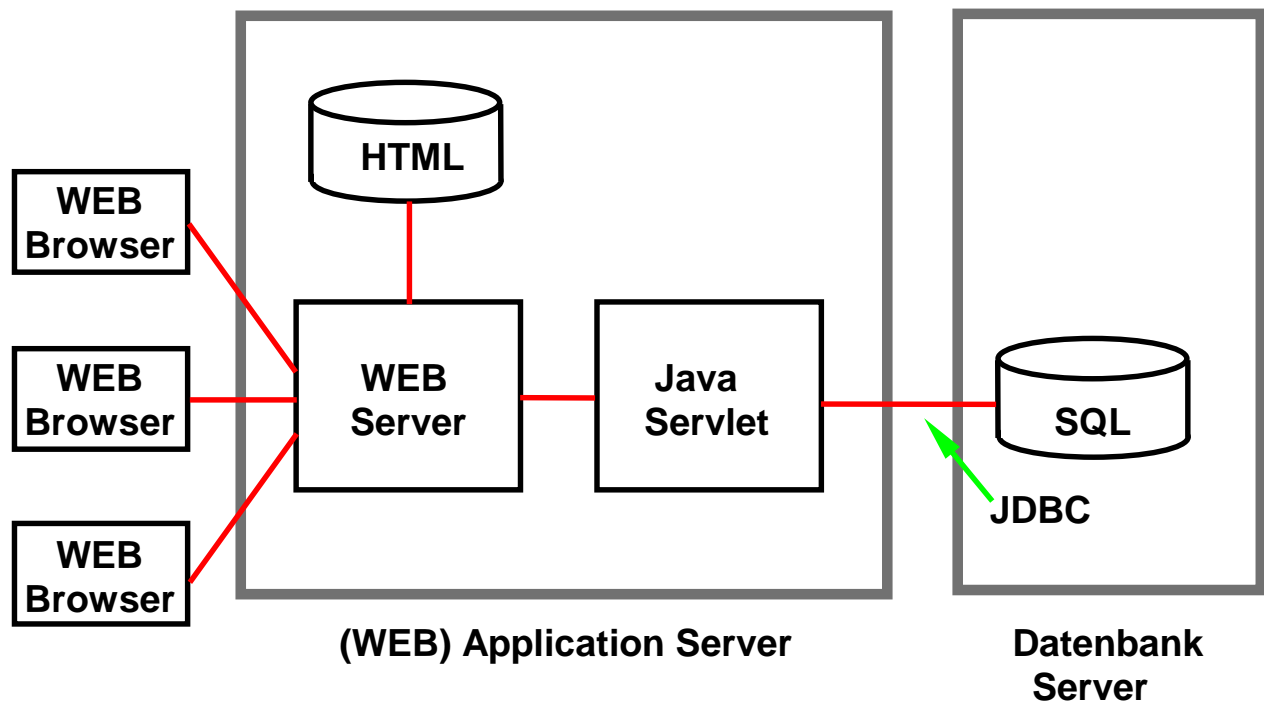
Dynamischer WEB Seiten Inhalt (1)

Der Web Browser kommuniziert mit dem Web Server über das HyperText Transfer Protokoll (HTTP). HTTP ist das ursprüngliche Transport Protocol für das World Wide Web.

Zwei Alternativen:

1. Web Server sendet statische Seite aus der HTML Datenbank an den Web Browser.
2. Web Server erstellt dynamische Seite. Hierzu ruft er über die CGI Schnittstelle ein Anwendungsprogramm auf. Dieses kann z.B. Daten aus einer OS/390 DB2 Datenbank verwenden, um eine dynamische HTML Seite zu erstellen.

CGI Programme werden häufig in einer Script Sprache, z.B. PERL, PHP, REXX oder Tcl/Tk erstellt, können aber auch in einer beliebigen anderen Sprache, z.B. C++, Cobol oder Java geschrieben werden.

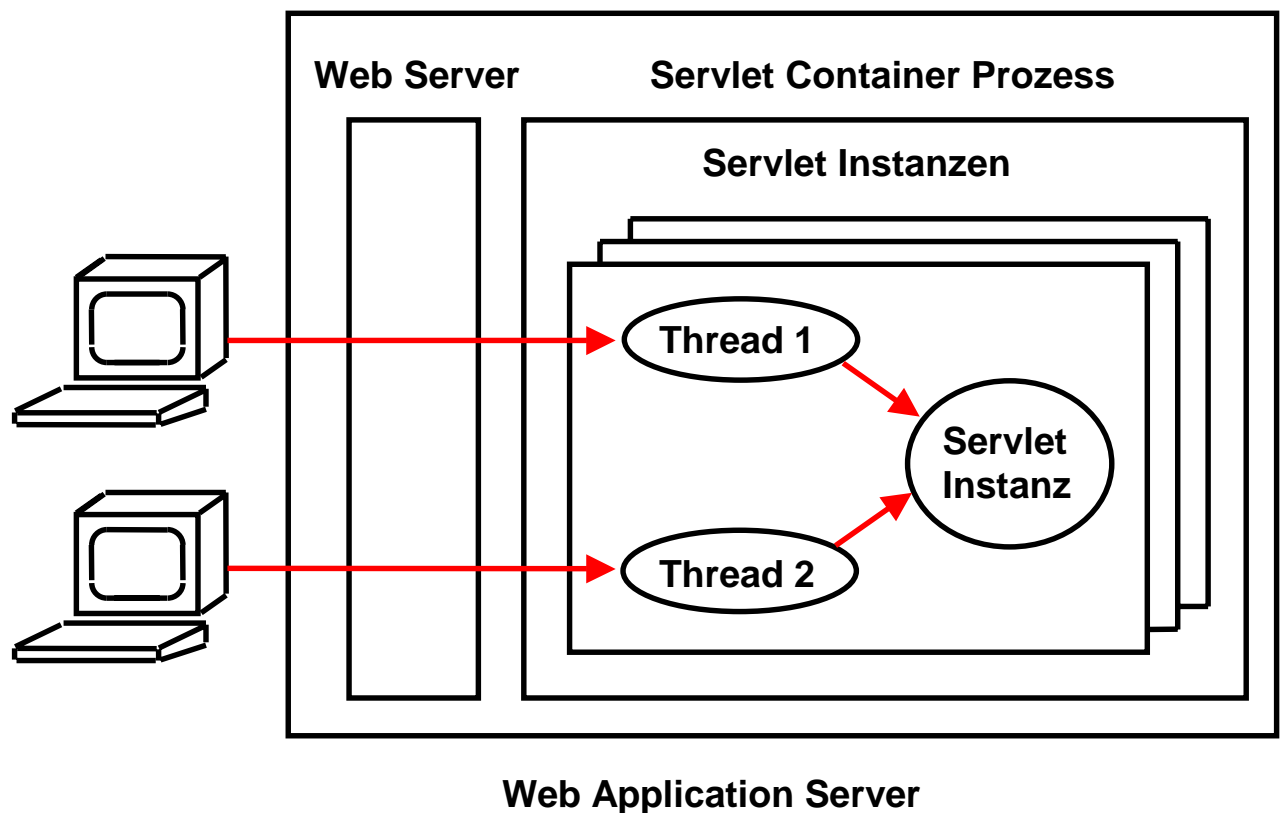


Dynamischer WEB Seiten Inhalt (2)

Im Gegensatz zu CGI erfordert das Java Servlet nur light weight Context Switches. Daher deutlich besseres Leistungsverhalten.

Servlets verfügen über alle Java API's, einschließlich JDBC (Java Data Base Connectivity).

Java Server Pages (JSP) sind eine Erweiterung der Servlet API. Verwenden in Java geschriebene XML - ähnliche Tags und Scriptlets.



Servlet Instanzen und Threads

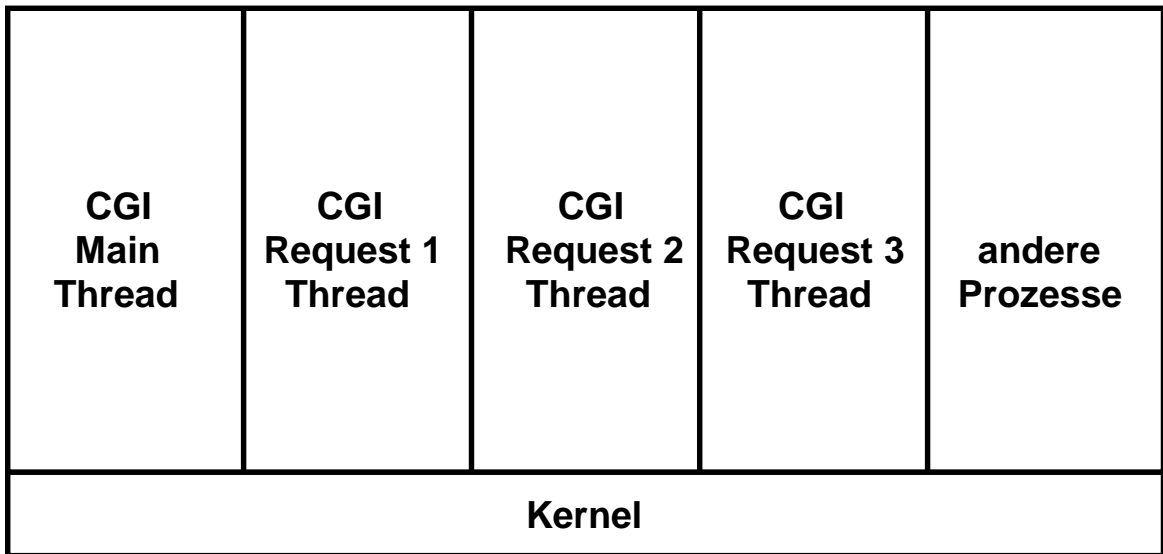
Es existiert nur eine einzige Instanz des Servlets

mehrfache Threads, die mehrere Klienten Requests gleichzeitig befriedigen können

Servlets werden dynamisch geladen, wenn sie erstmalig angefordert werden

oder statisch laden beim Hochfahren des Containers

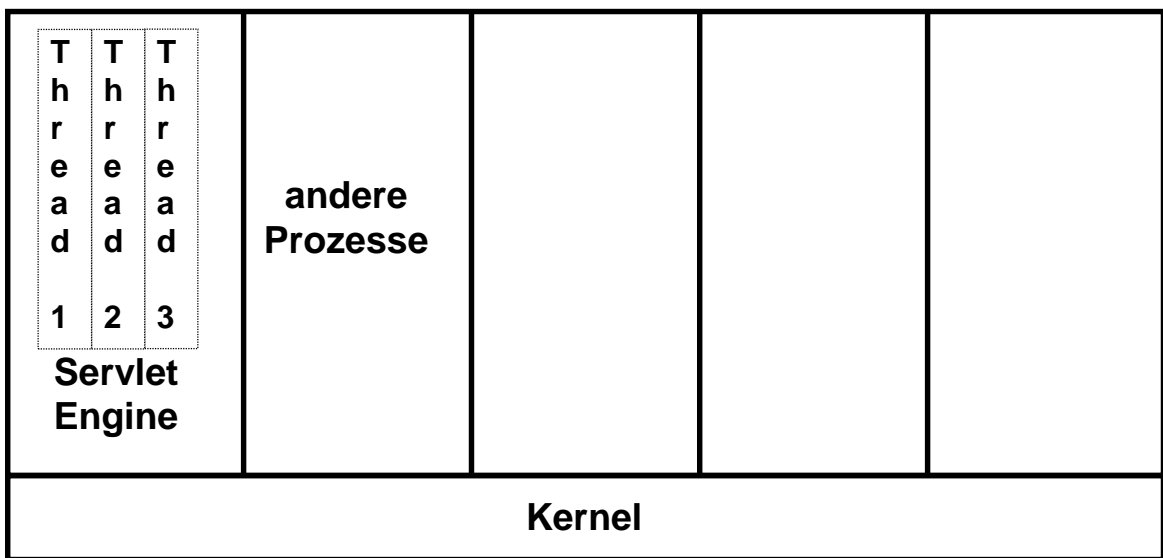
FF..FF



00..00

CGI Ansatz

FF..FF



00..00

Servlet Ansatz

Ein Servlet kann über ein Session Objekt verfügen, welches Session Information über mehrfache Zugriffe verwaltet

Ein Servlet kann einen Pool von Data Base Connections verwalten und eine Data Base Connection über Zugriffsgrenzen aufrechterhalten

Servlet Container

Ein Servlet ist eine reguläre Java Klasse, die über **extends HttpServlet** die Servlet Eigenschaften erbt, und nur in einer Servlet Laufzeitumgebung ausführbar ist. Die Servlet Laufzeitumgebung wird auch als Container oder Servlet Engine bezeichnet.

Ein Servlet Container ist ein Programm innerhalb einer JVM, das Requests für Servlets und Java Server Pages (JSP) behandelt. Der Servlet Container ist verantwortlich für:

- Erstellung von Servlet-Instanzen,
- Initialisierung von Servlets,
- Dispatching von Requests,
- Verwaltung des Servlet-Kontextes für die Nutzung durch die Web-Anwendungen.

Servlet Container haben keine Transactions-, Persistence- und Sicherheitseigenschaften. Sie verbessern u.a. die Servlet-Ausführungszeit.

HelloWeltServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public
class HelloWeltServlet extends HttpServlet
{

    public final static String message = "<html>\n" +
        "<head><title>Hallo Welt</title></head>\n" +
        "<body>\n" +
        "<h1>Hallo Welt</h1>\n" +
        "</body></html>\n";

    public void init()
    {
        System.out.println("In HelloWeltServlet init");
    }

    public void destroy()
    {
        System.out.println("In HelloWeltServlet destroy");
    }

    public void service(ServletRequest req, ServletResponse res)
    throws ServletException, IOException
    {
        PrintWriter out = res.getWriter();
        out.println(message);
    }
}
```

Basic Servlets

Try our basic servlets for common tasks:

- [Echo a request \(RequestInfo\)](#)
- [Query a database \(JDBC\)](#)
- [Process form input \(FormProcessing\)](#)
- [Display form input \(FormDisplay\)](#)

Travel

Are you looking for an exciting vacation? Xtreme Adventures knows which site you were visiting and offers you trips you'll love. And while you're browsing, you can exchange messages with other potential travelers.

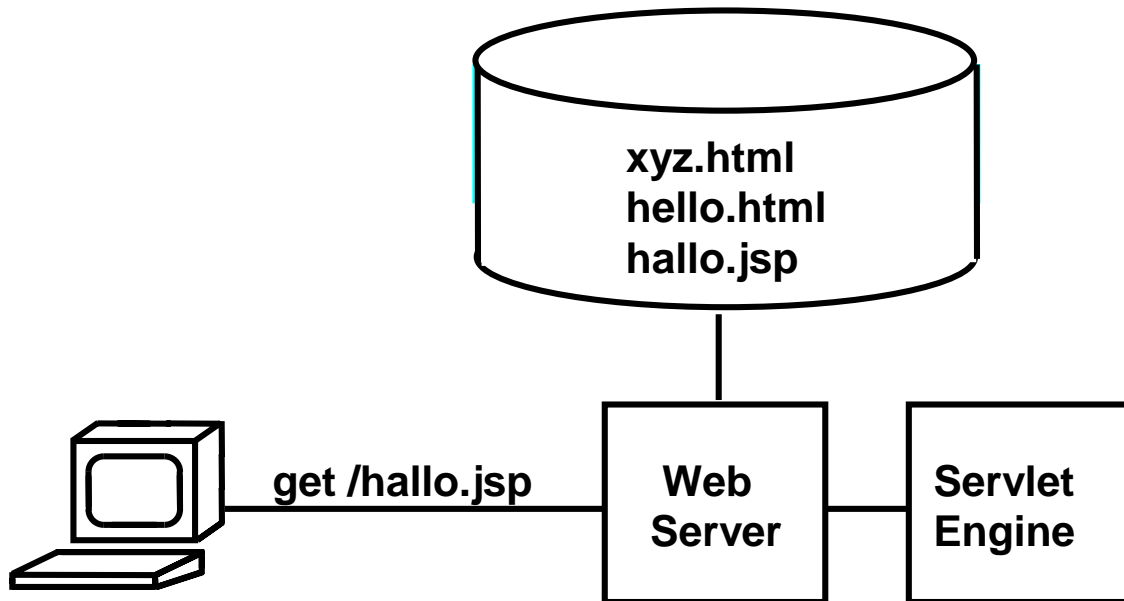
WOM Bank

Open accounts, deposit, transfer, or withdraw money. WOM Bank connects to a DB2 database to authenticate customers and maintain transaction information.

Ticket Server

TicketCentral searches pre-loaded DB2 database tables to find available seats at your favorite events. Order your tickets and purchase them with your credit card.

<http://jedi.informatik.uni-leipzig.de/IBMWebAS/samples/>



Einfache Java Server Page

Den folgenden Text in einer Datei mit der .jsp extension speichern (z.B. hallo.jsp) und mit einem Browser ansehen:

```
<html>
<head>
<title>JSP Example </title>
</head>
<body>
<%= new java.util.Date() %>
</body>
</html>
```

Die Zeichenfolgen **<%** und **%>** schließen Java Ausdrücke ein. Diese werden zur Run Time ausgewertet.

Mit dem Ausdruck **<%= new java.util.Date() %>** wird die gültige Zeit bei jedem Reload der HTML Seite in den Browser wiedergegeben.

JSP Beispiel

```
<html>
<head><title>First Example</title></head>
<body>
<h3>Hello World-JSP</h3>
```

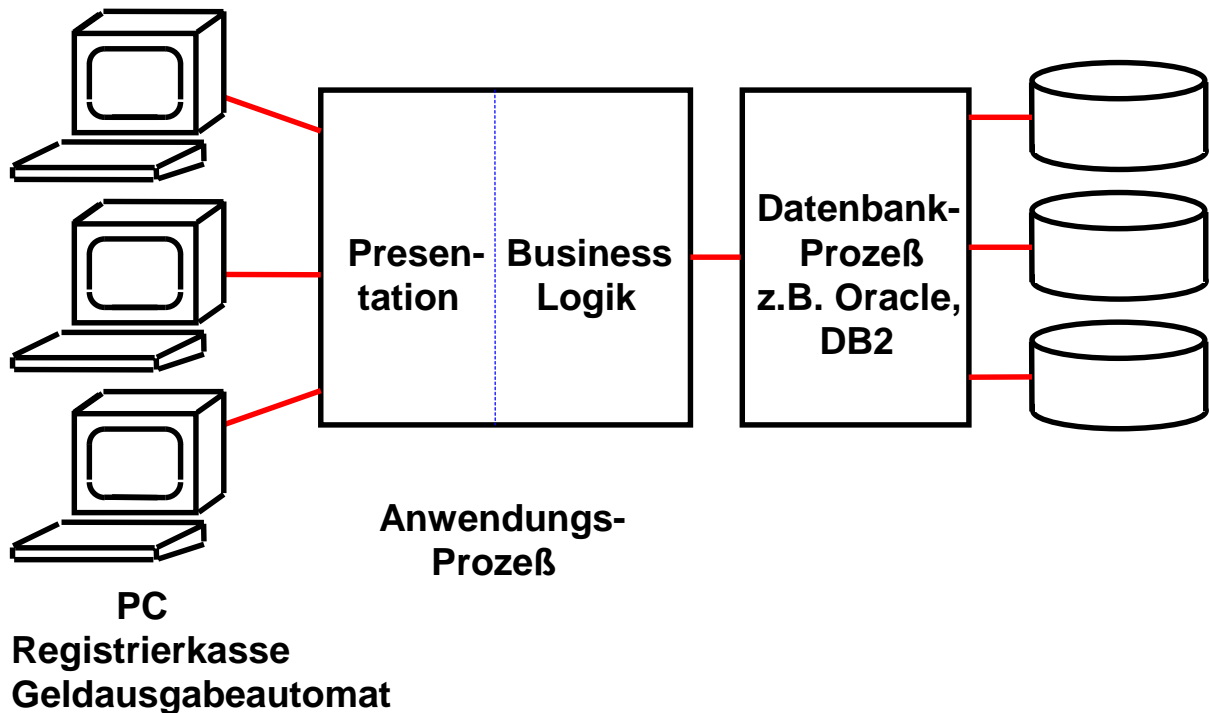
```
Your browser is: <%= request.getHeader("User-
Agent") %><br>
```

```
Your IP address is: <%= request.getRemoteAddr() %><br>
```

```
</body>
</html>
```

Erzeugt diese Ausgabe





Business- und Präsentationslogik

Ein sauber strukturiertes CICS Programm besteht aus zwei Teilen: Business Logik und Präsentations-Logik.

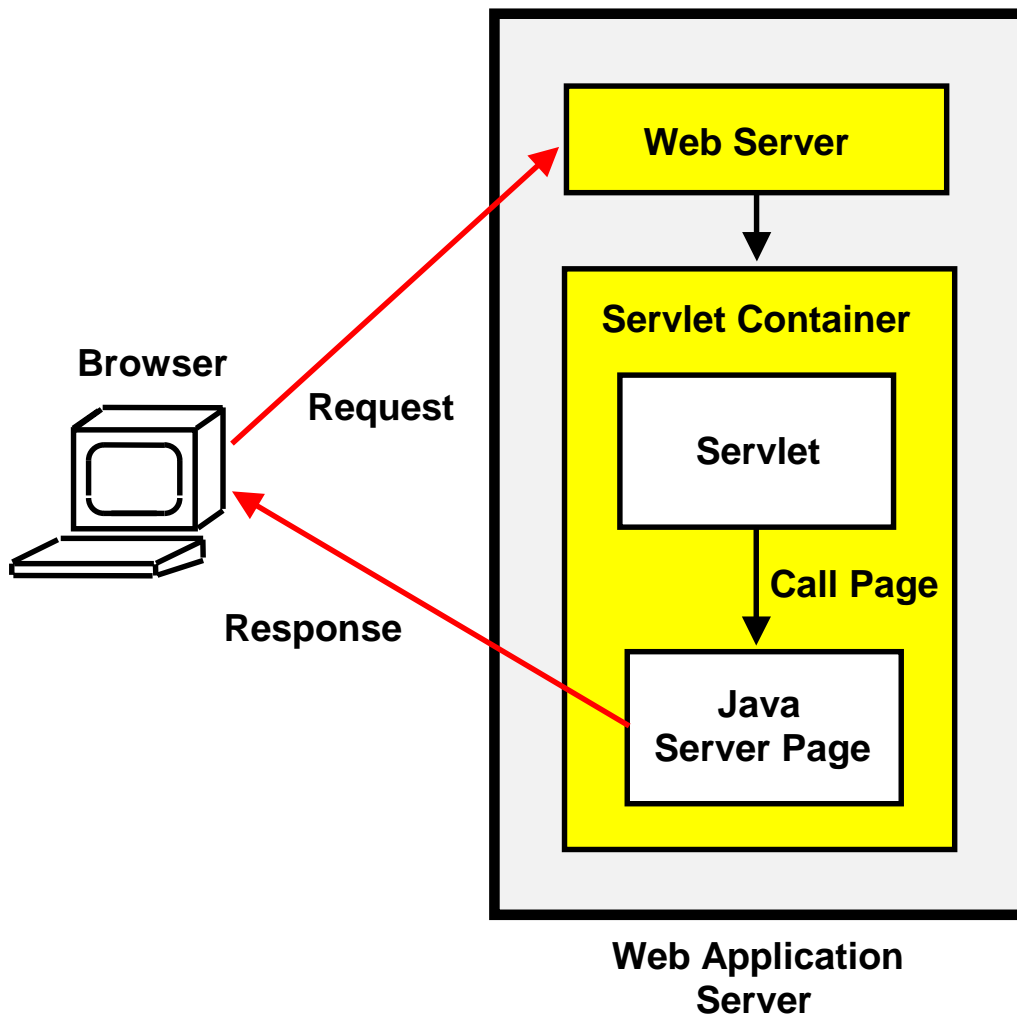
Business Logik ist der Teil, in dem Berechnungen erfolgen und Daten in einer Datenbank gelesen/geschrieben werden.

Präsentations- Logik ist der Teil, in dem die Ergebnisse der Berechnungen so aufgearbeitet werden, daß sie dem Benutzer in einer ansprechenden Art auf dem Bildschirm dargestellt werden können.

Business Logik wird in Sprachen wie C, C++, COBOL, PL/1, Java usw. geschrieben.

Für die Präsentations - Logik gibt es viele Möglichkeiten. Die modernste Alternative benutzt Java Server Pages und einen Web Application Server um den Bildschirminhalt innerhalb eines Web Browsers darzustellen.

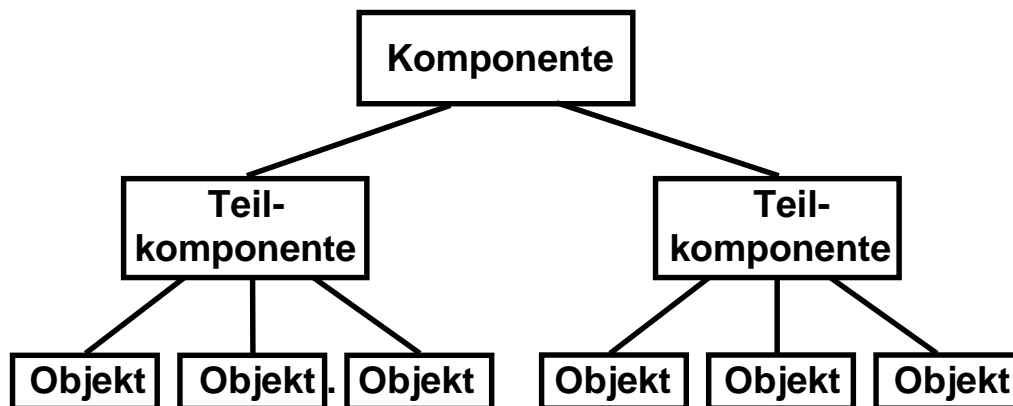
Die älteste (und einfachste) Alternative verwendet das CICS BMS (Basic Mapping Support) Subsystem. BMS Programme werden in der BMS Sprache geschrieben.



Interaktion Servlet - JSP

Es ist durchaus möglich, innerhalb eines Browsers eine .jsp Seite mittels ihrer URL direkt aufzurufen.

Üblicher ist der Aufruf eines Servlets mittels eines Form Tags. Das Servlet wiederum ruft eine Java Server Page auf, welche die Ausgabedaten erzeugt.



Komponenten

Komponenten sind unabhängige, in sich abgeschlossene, wohl definierte Software Einheiten, die eine spezifische Leistung über standardisierte Schnittstellen bieten.

Komponenten lassen sich mit anderen Komponenten zu größeren Einheiten zusammenfügen, die wiederum Komponenten oder eigene Anwendungen sind.

Komponenten setzen sich typischerweise aus Objekten zusammen.

Komponenten lassen sich durch geeignete Parameterisierung in einer Vielzahl von Entwicklungsumgebungen einsetzen, und sind unabhängig von Sprache, Betriebssystem und Hardware.

Eine Komponente hat

- **eine Art "Stecker", mit dem sie sich verbinden kann.**
- **eine Art "Steckdose", welche benutzt wird, um verschiedenen Komponenten die Möglichkeit zu geben, sich "einzustecken".**
- **die Möglichkeit Informationen über sich selbst bekannt zu geben.**
- **eine spezifizierte Menge von Eigenschaften**

Begriffe

Business Object, System Level Object, Metadaten

Ein Business Object ist eine Komponente der Anwendungsschicht, die in nicht voraussehbaren Kombinationen eingesetzt werden kann.

Ein Business Object repräsentiert auf eine erkennbare, nachvollziehbare Art ein Objekt (eine Entity) des täglichen Lebens.

Das Konzept eines Business Objektes steht dem betriebswirtschaftlich motivierten Anwendungskontext sehr viel näher als dem technisch motivierten Konstrukt eines programmiersprachlichen Objektes.

Ein System Level Objekt repräsentiert eine Komponente, mit der der Benutzer nie direkt etwas zu tun hat. Sie wird lediglich von Informationssystemen und Programmen genutzt.

Componentware ist Software, deren Teile sich baukastenartig zusammensetzen lassen.

Metadaten sind sich selbst beschreibende Information, welche die dynamische Struktur eines Systems definieren. Im Falle einer Datenbank beschreiben Metadaten die Datenbank Struktur und werden im „Repository“ verwaltet.

Business Objekt „Kunde“

Komponente	1	Name, Vorname	
	2	Titel, Anrede	
	3	Straße	
	4	PLZ	}
	5	Ort	
	6	Straße	Firmenanschrift 1
	7	PLZ	}
	8	Ort	
	9	Straße	Firmenanschrift 2
	10	PLZ	}
	11	Ort	
	12	Straße	Privatanschrift
	13	Kunden-Nr. =	
		f(Object Nr., Objekt Identifikation	
	14		
	•		
	•		
	•		
	•		

ca. 500 Attribute:

Fax Nr.
Tel. Nr.
e-mail Adresse
Info über Partner
Info über Kinder
Zeiger auf Kinder
Arbeitgeber
Stellung im Betrieb
Mitglied im Tennisclub Blau Weiss
Rentendaten
•
•
•
•

Java Beans

Objektorientiertes Java Komponenten Modell, JavaBeans sind Java binary parts

häufig für visuelle Komponenten eingesetzt (etwa Buttons und Scrollbalken)

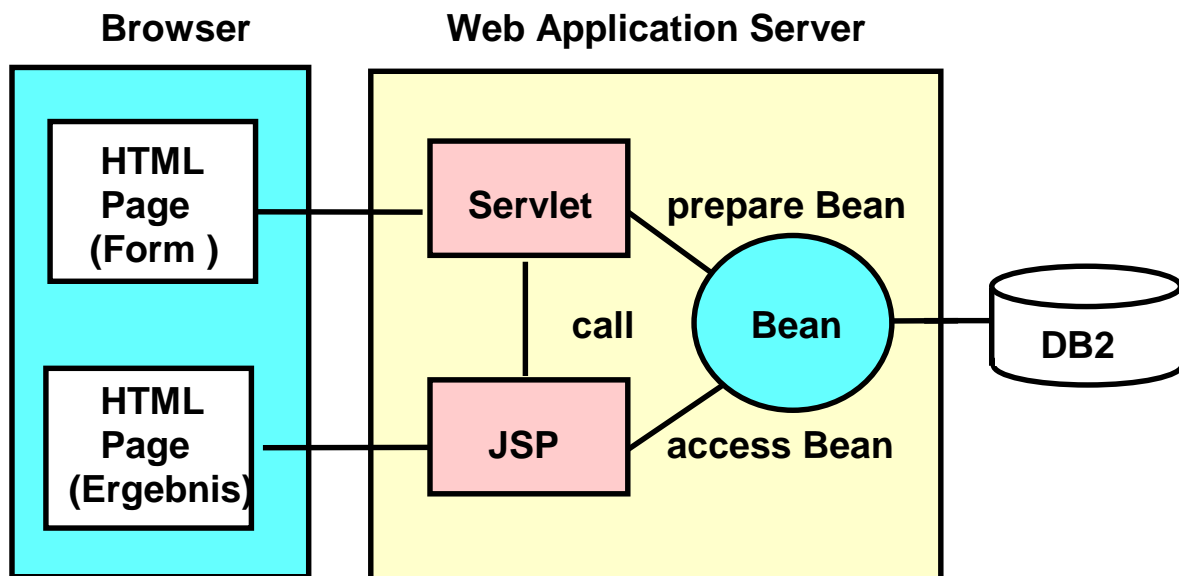
Hauptmerkmale :

- **Properties (Eigenschaften, z.B. get und set)**
- **Methoden und**
- **Events (Ereignisse)**
- **Namens Konventionen**
- **Introspection (BeanInfo Klasse)**

Der JavaBeans Komponenten Modell Teil des JDK Lieferumfangs unterstützt:

- **Sicherheit (benutzt den Java Security Manager)**
- **Versionsmanagement**
- **Life-Cycle Management**
- **Event Notification,**
- **Configuration und Property management**
- **Scripting**
- **Meta-Daten und Introspection**
- **Persistenz (über Serialisierung)**
- **Benutzbarkeit (die „BeanBox“ des JDK ist ein Prototyp einer grafischen Umgebung für das Zusammensetzen von Beans)**
- **Eigeninstallation(über Java Archiv Files)**

Für unternehmensweite Anwendungen (Enterprise Applications) fehlen Schlüsseleigenschaften, z.B Transaktionsdienste, Namensdienste und Sicherheitsdienste. Werden JavaBeans hiermit angereichert, spricht man von Enterprise JavaBeans.



Nutzung von Java Beans

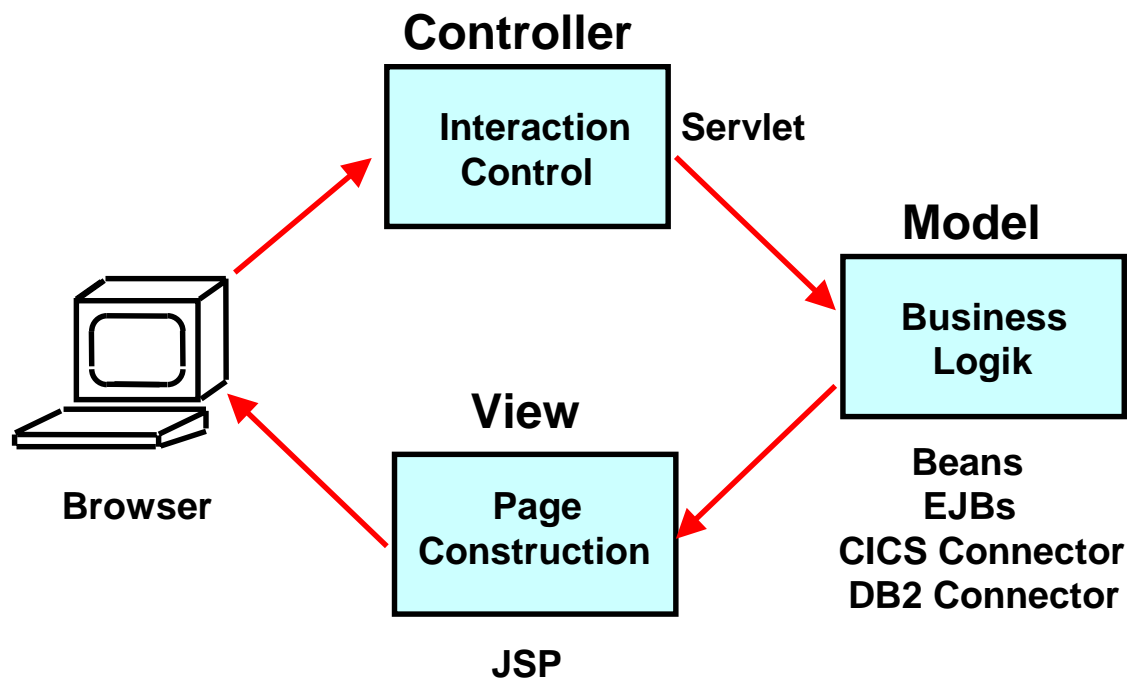
Ein Servlet ist ein Java Programm, das Bildschirm Output in der Form einer HTML Datei produziert.

Eine JAVAServerPage ist eine HTML Seite mit zusätzlichen JSP Tags.

Wird eine JSP Seite aufgerufen, so kompiliert sie ein JSP Übersetzer in ein Servlet.

In der Praxis: Servlets und JSP werden von verschiedenen Leuten erstellt (Model-View-Controller Ansatz). Eine JSP ist zwar eine vollwertige Java Komponente, aber der Java Code Anteil innerhalb der JSP wird in der Regel auf ein Minimum reduziert.

Es existieren (wie für HTML Seiten) spezielle Werkzeuge für das Erstellen von JSP's, die das Hand-coding von HTML Statements automatisieren.



Model/View/Controller Triade (MVC)

Das „Modell“ ist ein Anwendungsobjekt und kapselt die Business Logik. Der „View“ ist die Screen Darstellung dieses Objektes. Der „Controller“ definiert, wie die Benutzerschnittstelle auf Benutzereingaben reagiert.

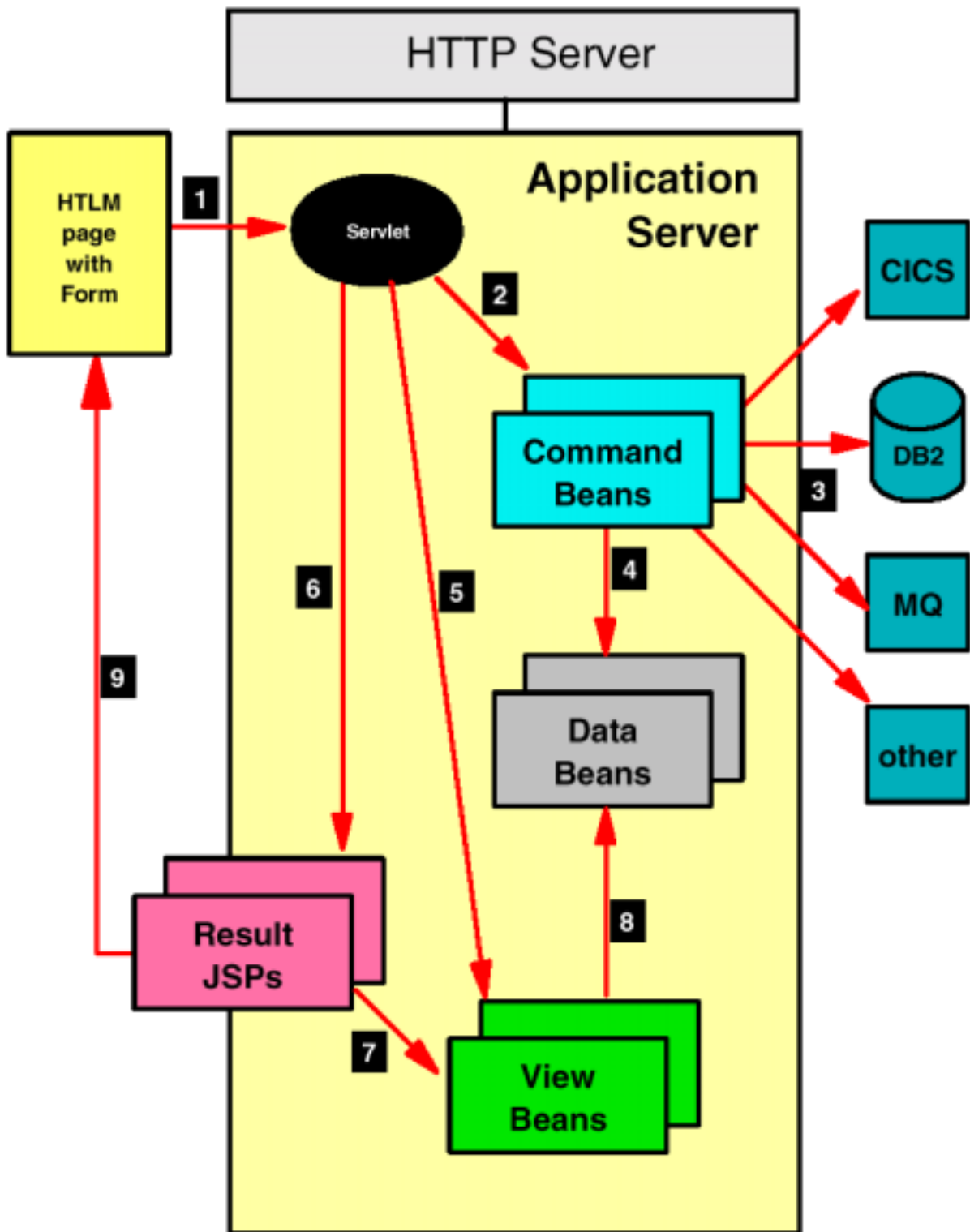
Command- und Data Beans oder Enterprise Java Beans (plus häufig CICS, IMS Programme, oder Stored Procedures) sind das „Modell“ (=Business Logik).

JSP's und View Beans sind der „View“

Das Servlet ist der „Controller“

MVC entkoppelt Modell und View zur Verbesserung von Flexibilität und Re-Use. Der Entwickler der Browser Darstellung arbeitet nur mit der Java Server Page.

Zentrisches Programmier Modell. Die gesamte Anwendungslogik (EJB, Servlet, JSP) läuft auf dem Server. Der Klient (*thin client*) braucht nur einen Browser.

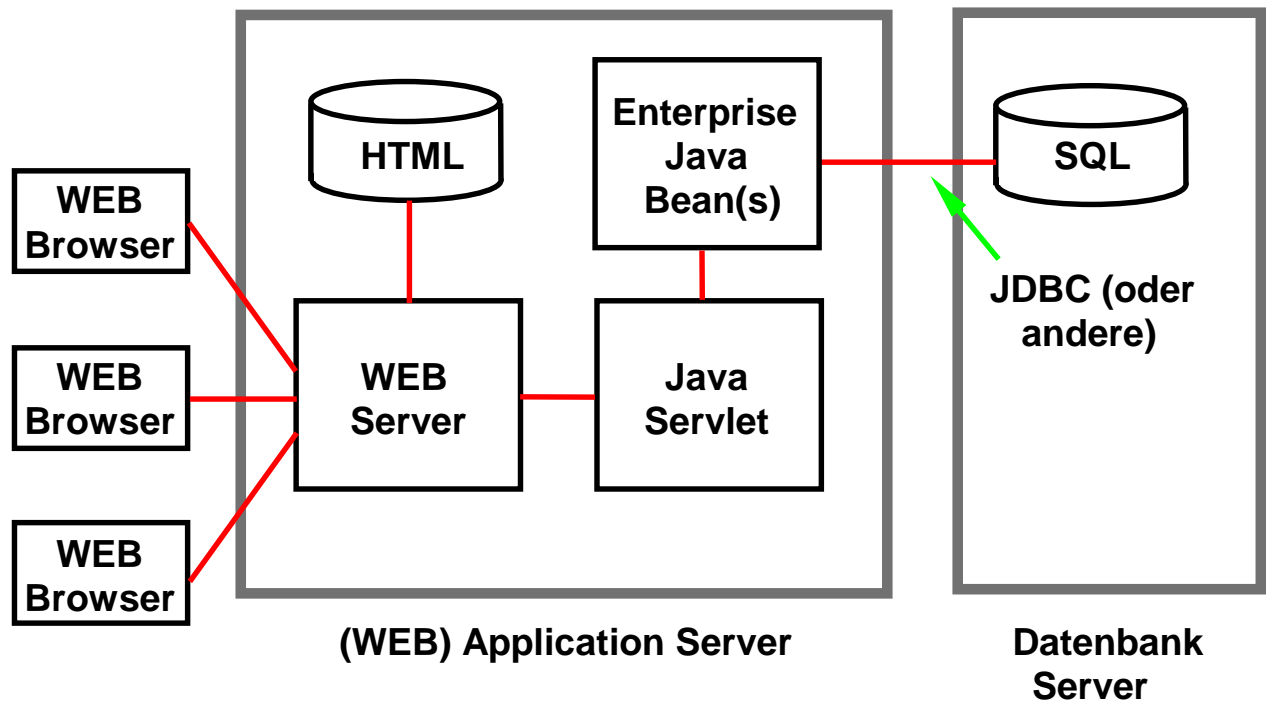


Architektur einer JSP Web Anwendung

Architektur einer JSP Web Anwendung

1. **HTML page:** static or dynamic HTML page, created from a previous step, contains one or multiple forms that invoke a servlet for processing of the next interaction.
2. **Servlet gets control** from the Application Server to perform validation and control of flow; sets up and calls command beans that perform the business logic.
3. **Command beans control** the processing of the business logic; logic may be imbedded in the command bean, or it can be delegated to back-end or enterprise systems, such as relational databases, transactions systems (CICS, MQSeries, IMS, and so forth); command bean may perform one specific function or it may contain many methods, each for a specific task (task wrappers). Command beans invoke database and transaction systems using „connectors“.
4. **Results of command beans** (or back-end systems) processing are stored in data beans. Data beans could contain an SQL result or a CICS communication area (COMMAREA).
5. **View beans provide** the contract between the output producing JSPs and the data beans that contain the dynamic data to be displayed in the output; servlet initializes the view beans and registers them with the request block so that the JSPs can find them.
6. **Servlet calls a JSP** for output processing and formatting depending on the results of the command beans; JSPs generate the output for the browser.
7. **JSP use tags** to declare the view beans and get access to all the dynamic data that must be displayed in the output.
8. **View beans contain** one or multiple data beans and provides tailored methods so that the JSP has access to the data stored in the data beans; data beans may not provide the necessary methods for a JSP to access the data.
9. **JSP assembles** the output and sends it back to the browser as an HTML page with dynamic data; in many cases, that output again contains form(s) to enable the user to continue the dialog with the application.

Servlet is the controller
Command beans provide the model
JSP is the view



Dynamischer WEB Seiten Inhalt (3)

Im einfachsten Fall enthält das Java Servlet die Anwendungslogik. In komplexeren Fällen lohnt es sich, die Anwendung in Komponentenform zu implementieren. Java Beans implementieren das Java Komponentenmodell.

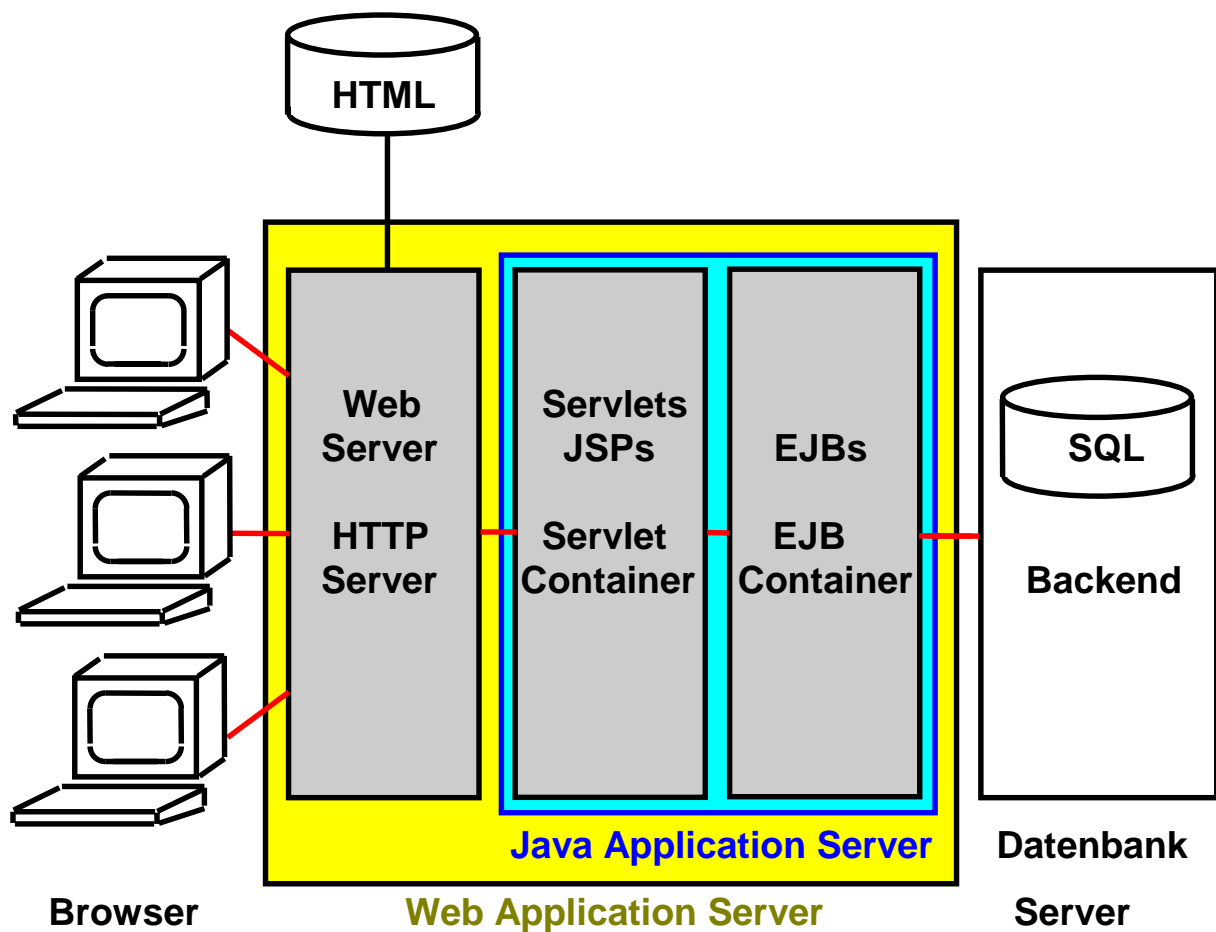
Enterprise Java Beans sind Java Beans mit zusätzlicher Funktionalität, besonders Transaktionseigenschaften (ACID), Persistenz und Sicherheit.

Enterprise Java Beans (EJB)

Java basiertes Server Komponentenmodell, März 1998. Teil der J2EE-Architektur. Version 3.0 verfügbar seit Februar 2005.

- EJB Komponenten sind serverseitige Komponenten, die ausschließlich in Java geschrieben sind. EJB Komponenten enthalten nur Business Logik, keine Präsentationslogik und keine Systemfunktionen.
- EJBs sind in einen „Container“ eingebettet (Laufzeitumgebung). Die EJB Architektur ist inhärent
transaktionsorientiert,
distributed,
portierbar,
multi-tier,
scalierbar und
sicher.
- Diese und weitere Systemfunktionen wie
Life-cycle management,
threading und
Persistenz
werden automatisch für die EJB Komponente von dem EJB Container zur Verfügung gestellt.
- EJB Komponenten werden deklarativ (über einen *Deployment Descriptor*) zur Laufzeit angepaßt. Die Anpassung bezieht sich auf Transaktionsverhalten, Sicherheitseigenschaften, life-cycle und state management und Persistenz.
- Interoperabilität von EJBs und CORBA konformen Objekten ist möglich.

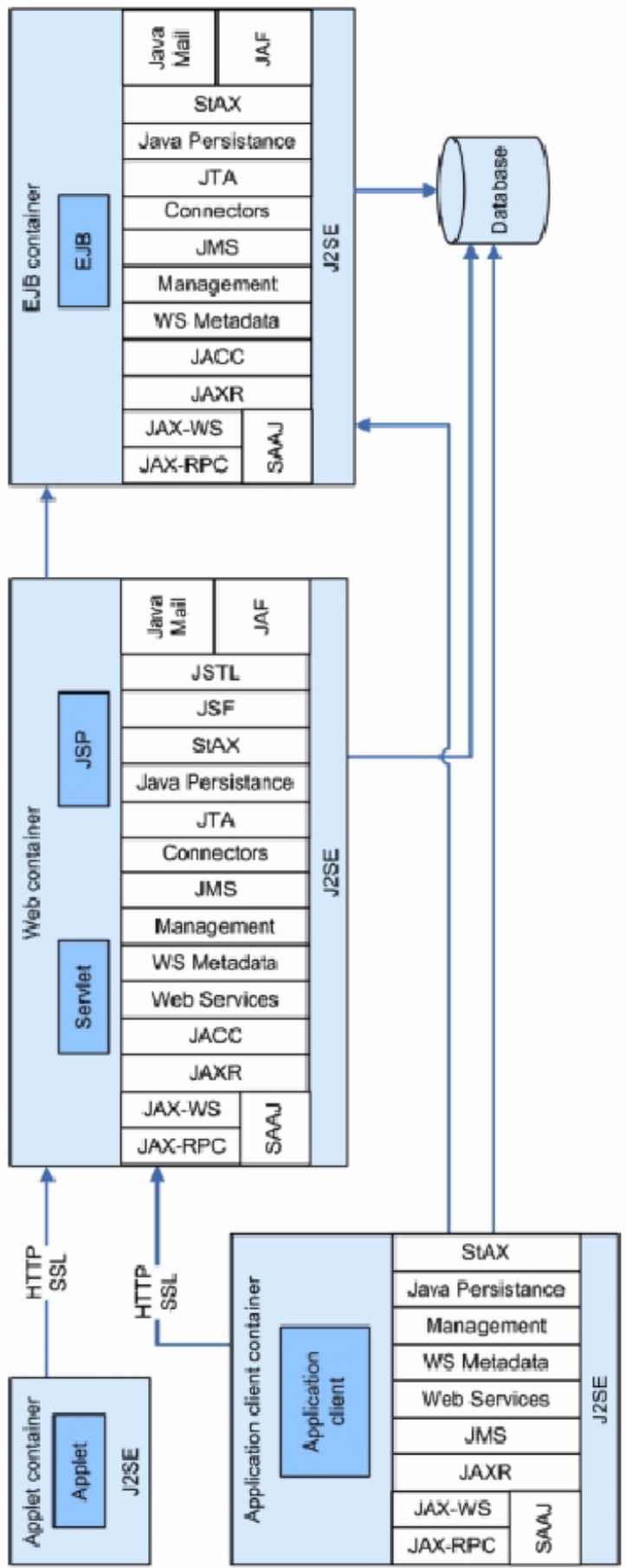
Der J2EE (Java 2 Platform, Enterprise Edition) Standard kombiniert Technologien wie Servlet, JSP, EJB, JMS, JCA und den JDK.



Application Server Hierarchie

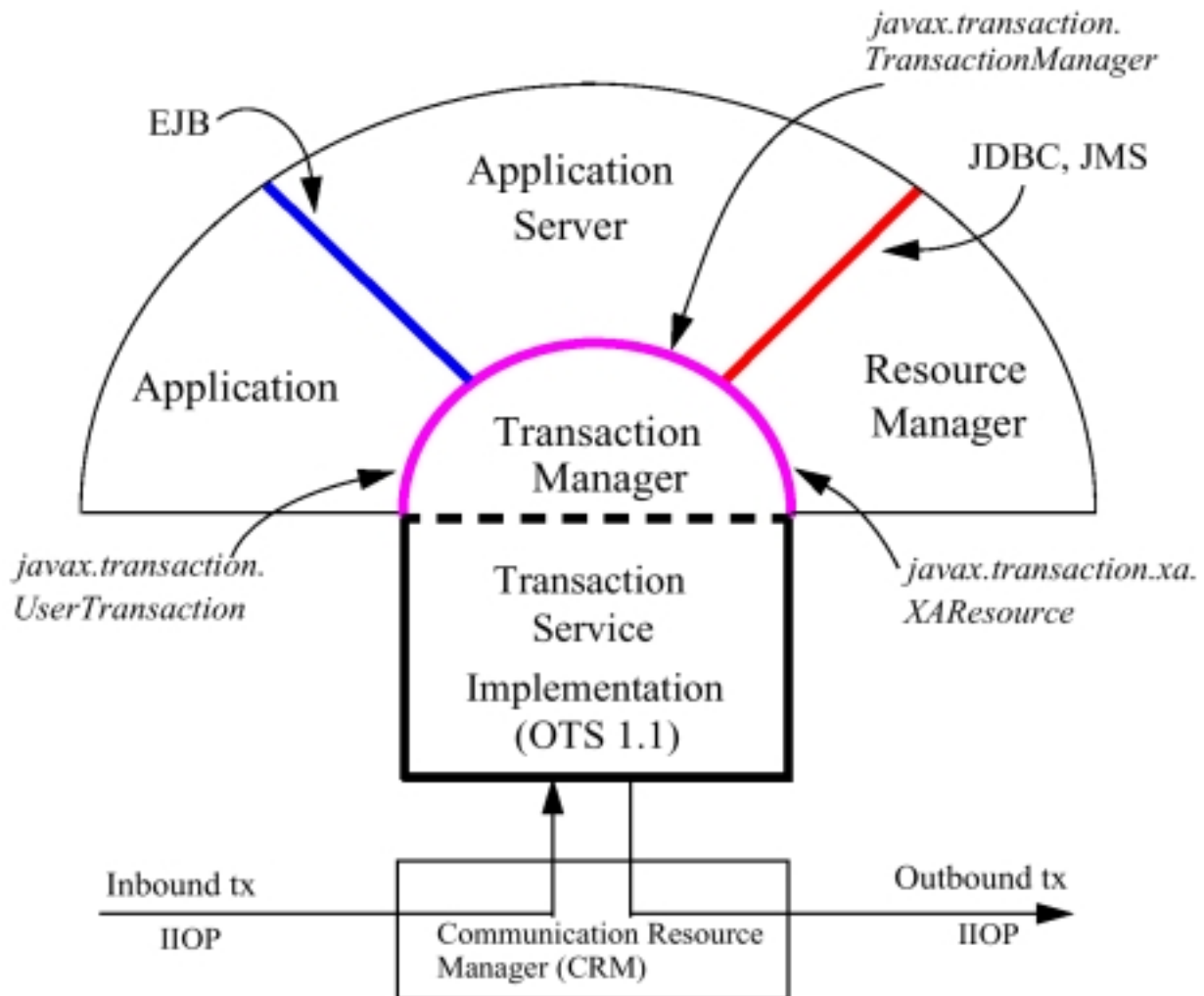
Der Web Application Server ist ein Prozess, der normalerweise in seinem eigenen virtuellen Adressenraum läuft. Er besteht aus mehreren Komponenten:

1. Der Web Server ist vielfach Apache.
2. Der Java Application Server unterhält u.a. eine Java Virtuelle Maschine
3. Der Servlet Container (Servlet Engine) ist eine Java Laufzeit Umgebung (runtime component) für die Ausführung von Servlets und Java Server Pages.
4. Der EJB Container ist eine Laufzeit Umgebung für die Ausführung von deployed Enterprise Java Beans.



Java 2EE Overview

Java Transaction Service (JTS) Java Transaction API (JTA)



Der Java Transaction Service (JTS) ist eine Java Abbildung (mapping) des OTS. JTS Kommunikation zwischen zwei Web Application erfordert daher IIOp (Beispiel 2 Phase Commit) .

Der Transaction Manager implementiert die JTA. Die Schnittstelle zwischen Transaction Manager und Transaction Service ist Hersteller abhängig. In der Praxis wurde der Transaction Service häufig durch einen existierenden Transaktionsmonitor implementiert:

- **BEA Web Logic benutzte Tuxedo**
- **IBM WebSphere benutzte Encina**

WebSphere Funktionen unter z/OS und OS/390

Web Application Server sind Laufzeitumgebungen für die Ausführung unternehmenskritischer Anwendungen, vergleichbar mit CICS.

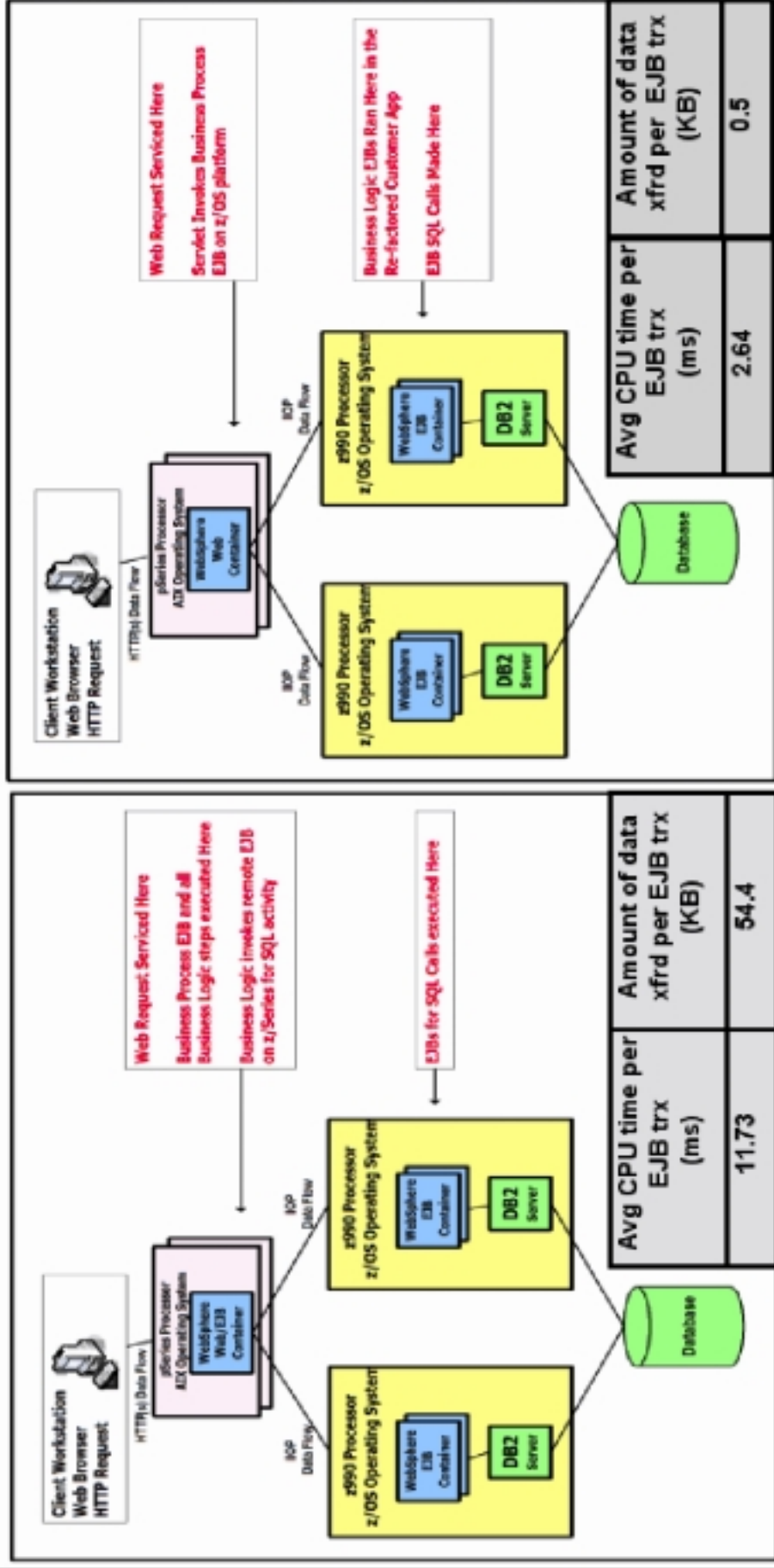
Im Gegensatz zu CICS sind die Anwendungen in Java programmiert, und ein Web Browser dient als Klient.

IBM WebSphere ist der Marktführer unter den Web Application Servern. Andere führende Web Application Server Produkte sind Weblogic der Fa. BEA und Netweaver der Firma SAP.

WebSphere ist unter Windows, Linux, AIX, Solaris, HP-UX und anderen Plattformen verfügbar. Die z/OS Version unterstützt u.a.

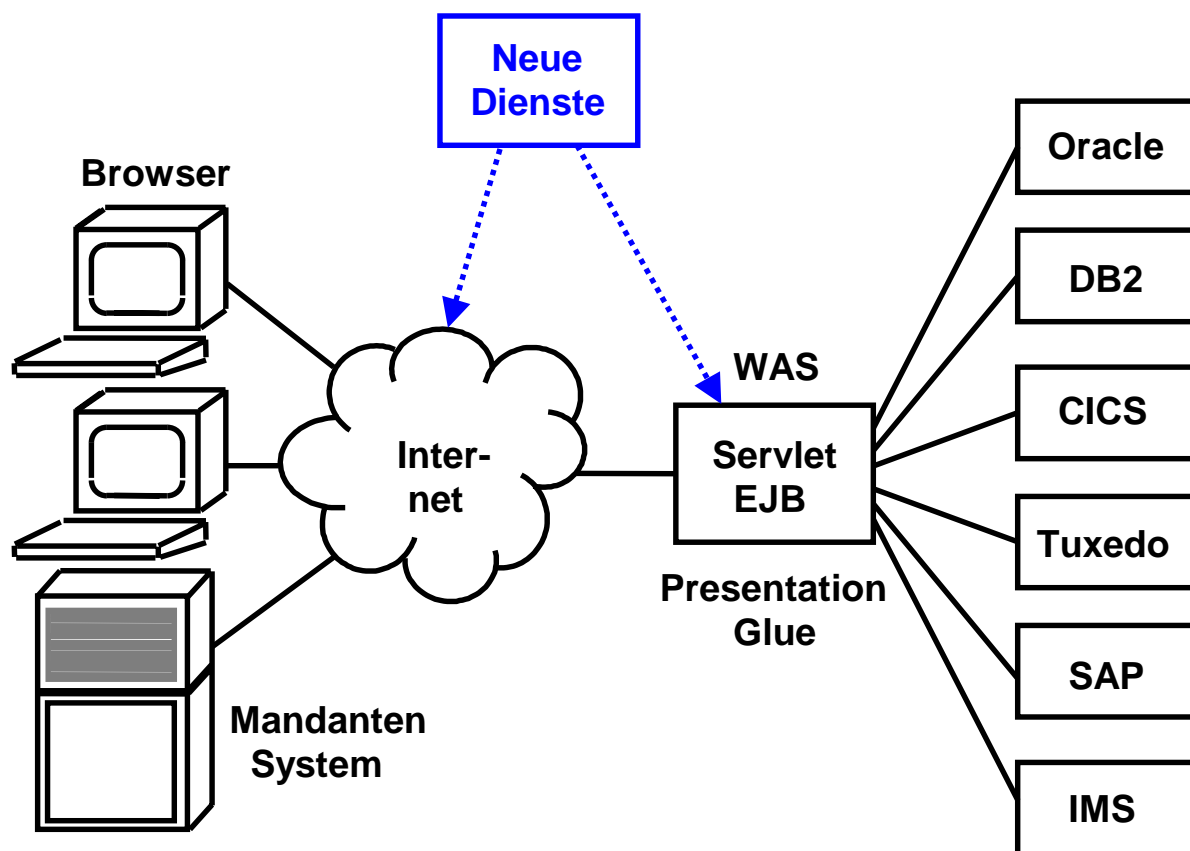
- **Parallel Sysplex**
- **Workload Manager**
- **RACF**
- **Crypto**
- **Java Connection Architecture**
- **Virtuelle Server**

Proximity to Data - Transportation Industry Benchmark



Advantages of Business Logic Proximity to z/OS data

- Average CPU time per EJB transaction was reduced by over 77%
- Number of bytes of data transferred per EJB transaction was reduced by 99%



Aufgabenstellungen

Internet Integration existierender Anwendungen

Business Process Reengineering

Unternehmenskritische Business Logik in Java

J2EE Connector Architecture JCA

Standard Architektur für die Integration von existierenden Business Logik Komponenten :

- **ERP Systeme, z.. SAP R/3**
- **Mainframe Transaktions Monitore, z.B. CICS, IMS**
- **Non- Java Legacy Anwendungen
(z.B. Stored Procedures)**
- **Datenbank Systeme**

Wichtigste Bestandteile der Architektur sind:

- **JCA Konnektoren, als Resource Adapter (RA) bezeichnet**
- **Common Client Interface (CCI)**

A Resource Adapter is a system level software library that is used by an application server or client to connect to a Resource Manager. A Resource Adapter is typically specific to a Resource Manager. It is used within the address space of the client using it. An example of a resource adapter is the JDBC driver to connect to relational databases.

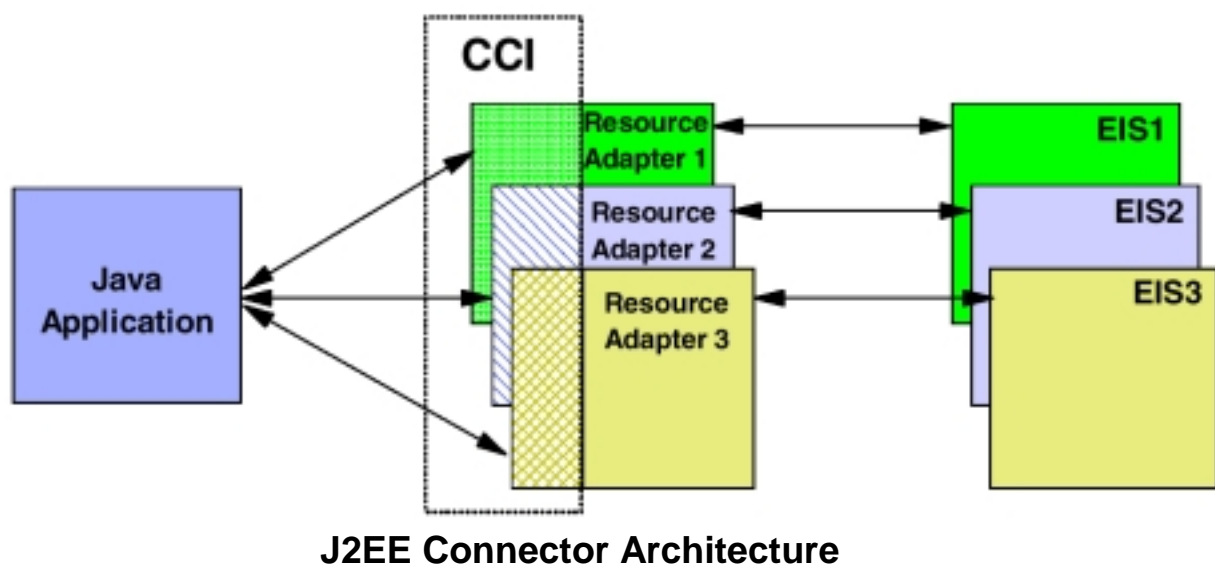
Für CICS existieren:

- **ECI resource adapter (für COMMAREA Zugriff)**
- **EPI resource adapter**

Das CICS Transaction Gateway (CTG) enthält ECI RA, EPI RA und CCI, sowie weitere Zugriffsmechanismen ausserhalb der JCA

JCA Common Client Interface CCI

Die Common Client Interface (CCI) definiert eine Standard Client API. Die CCI ist eine EIS unabhängige API. Anwendungen können mit einer identischen API unterschiedliche Resource Adapter verwenden. Ähnlich JDBC Interface

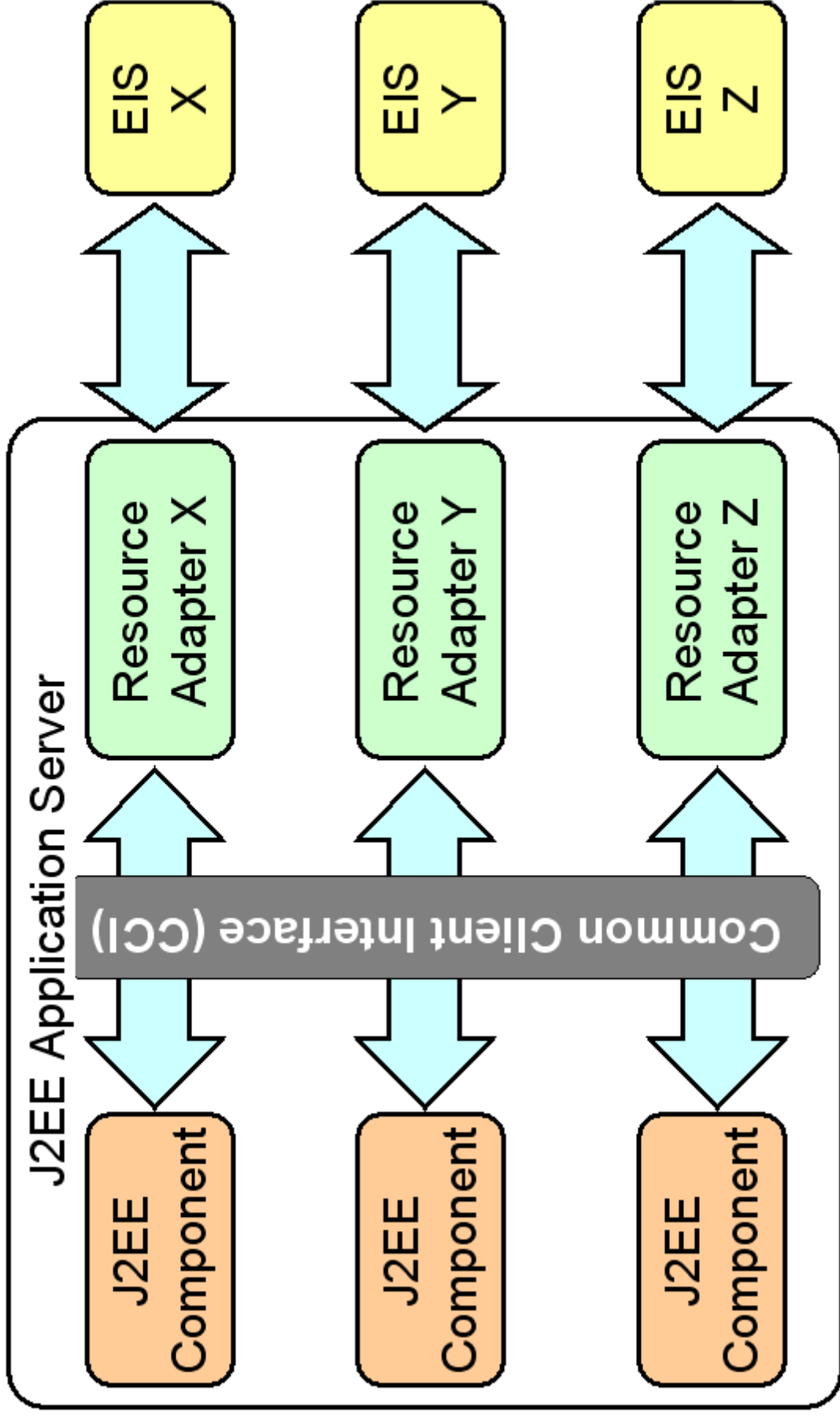


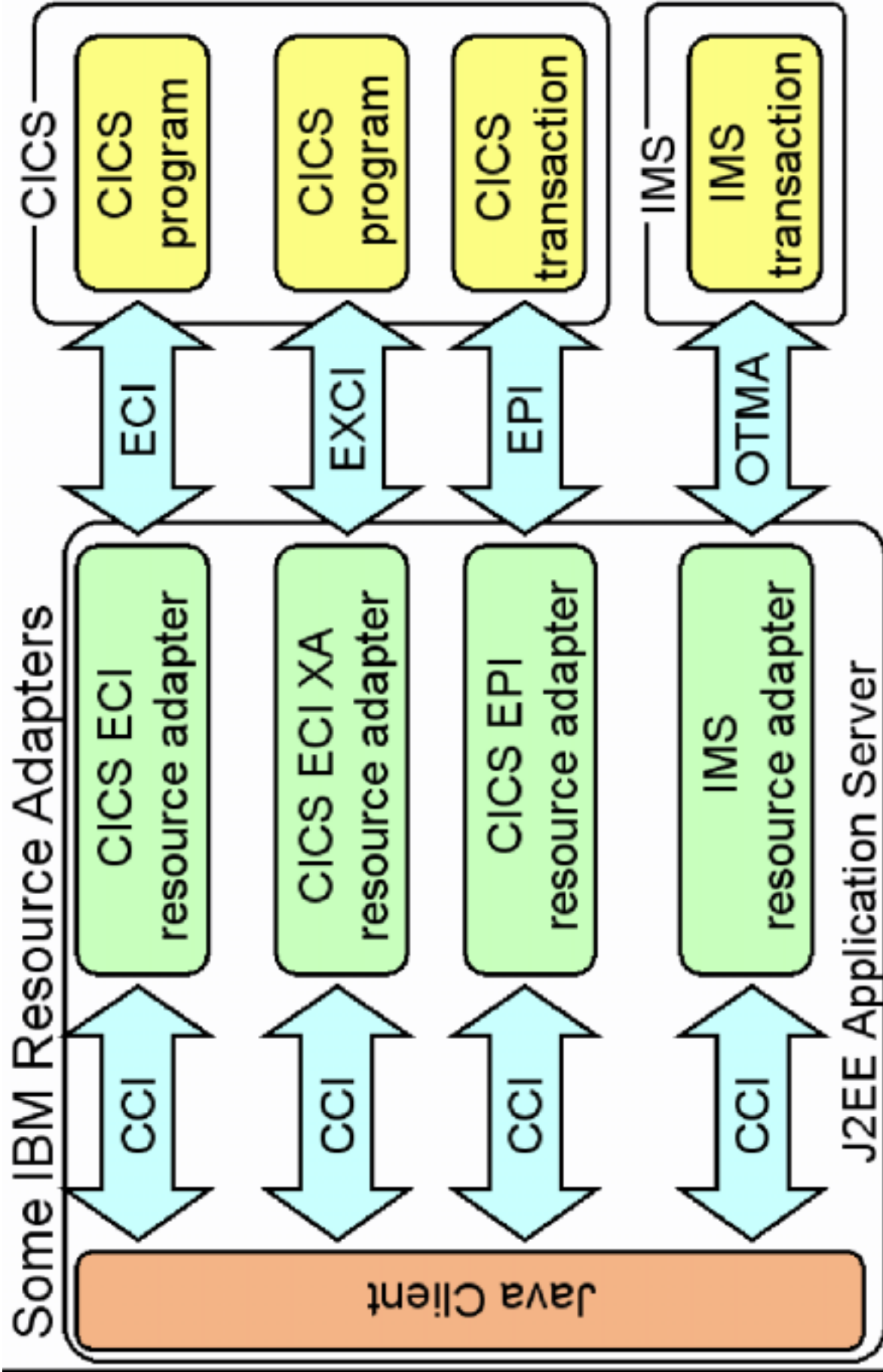
CCI Common Client Interface
EIS Enterprise Information System

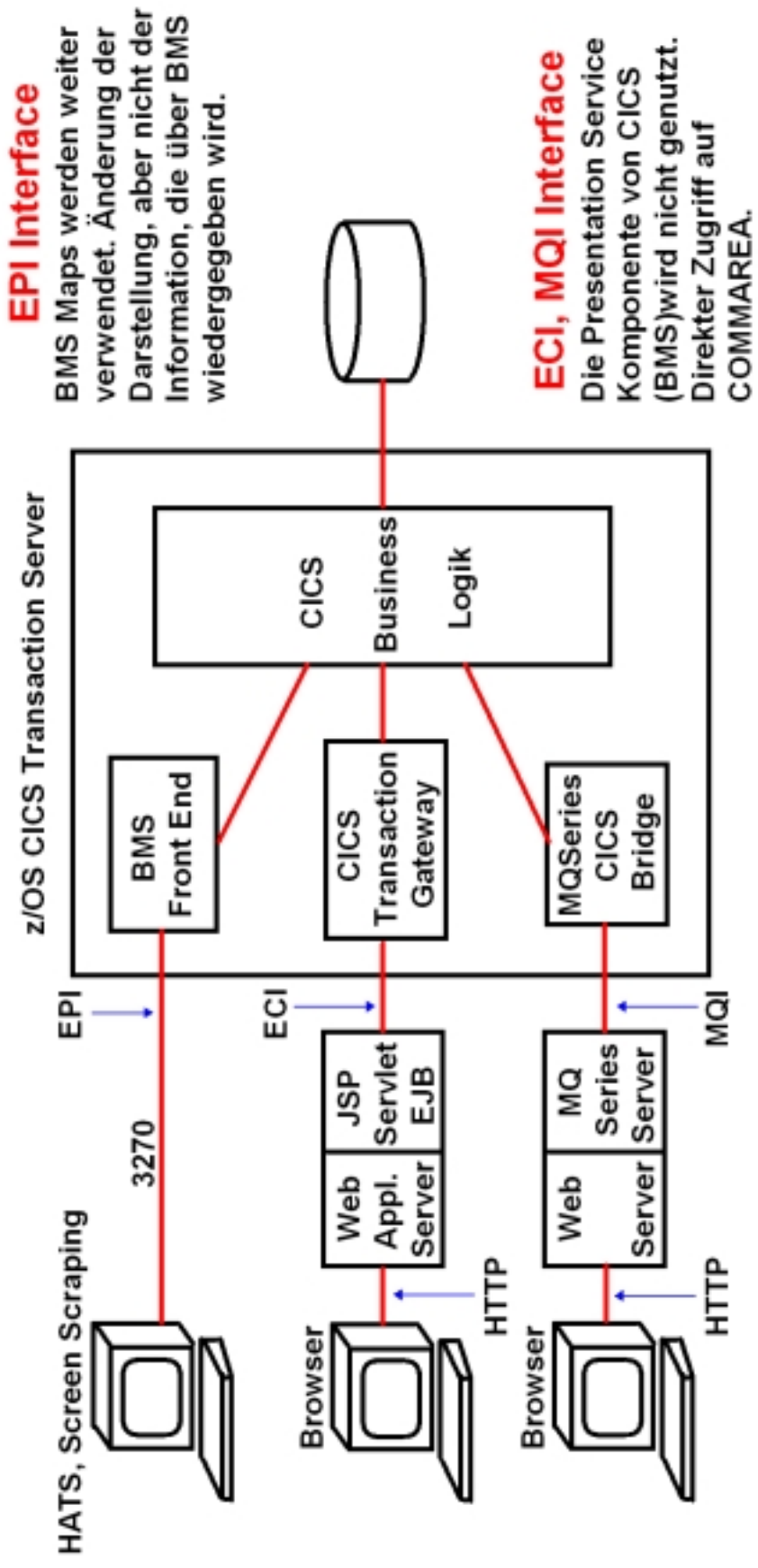
CCI und Resource Adapter sind als Java Klassen implementiert. Sie können von einer Java Anwendung alleinstehend benutzt werden (*non-managed environment*).

Üblich ist es, CCI und Resource Adapter als Elemente eines Web Application Servers einzusetzen (managed environment). Hierbei kann der Web Application Server das Management von Verbindungen, Transaktionen und Sicherheit direkt übernehmen. Die CCI Entwicklung kann in ein Entwicklungswerkzeug wie Eclipse integriert werden.

J2EE Connector Architecture Overview







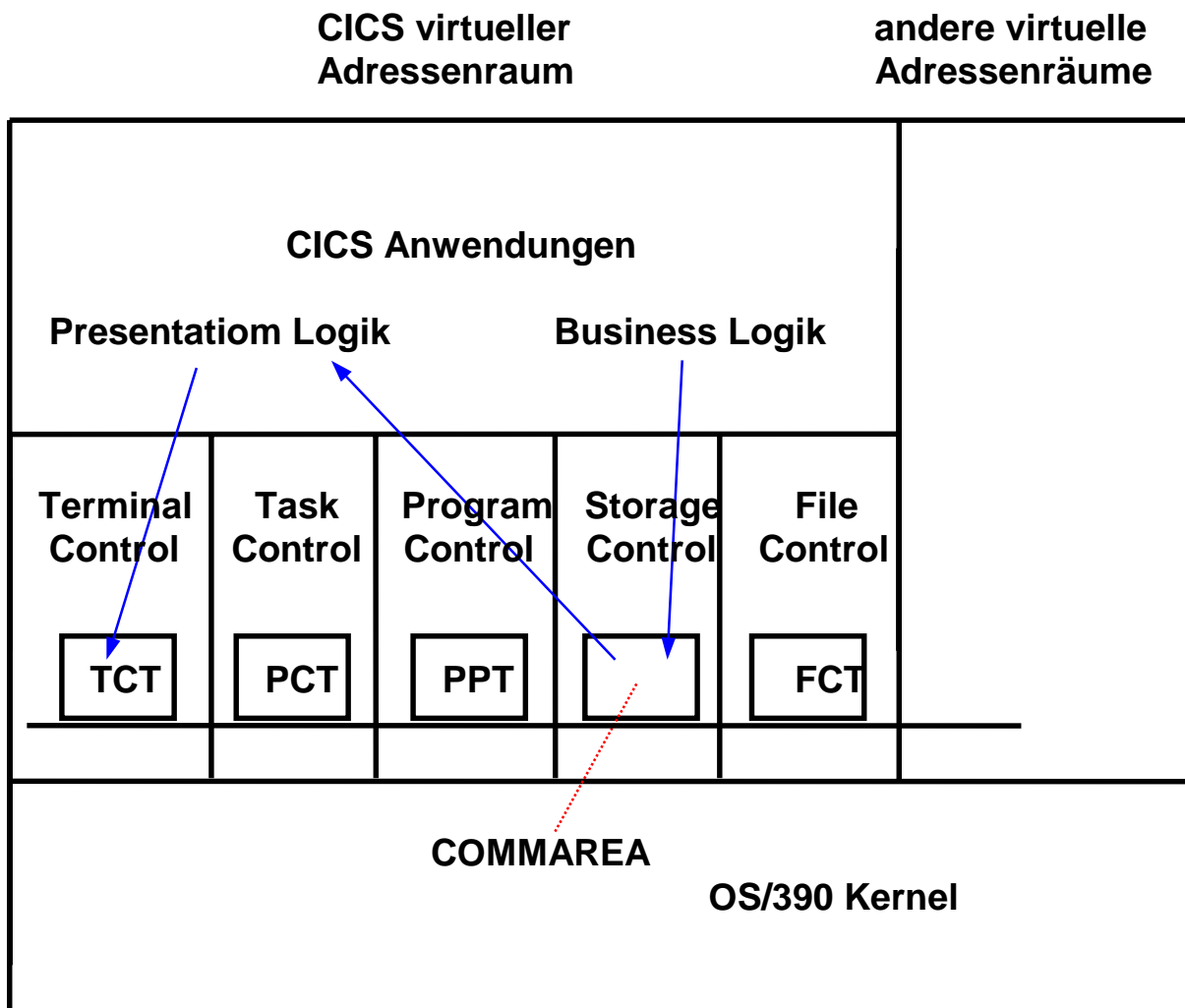
EPI Interface

BMS Maps werden weiter verwendet. Änderung der Darstellung, aber nicht der Information, die über BMS wiedergegeben wird.

ECI, MQI Interface

Die Presentation Service Komponente von CICS (BMS) wird nicht genutzt. Direkter Zugriff auf COMMAREA.

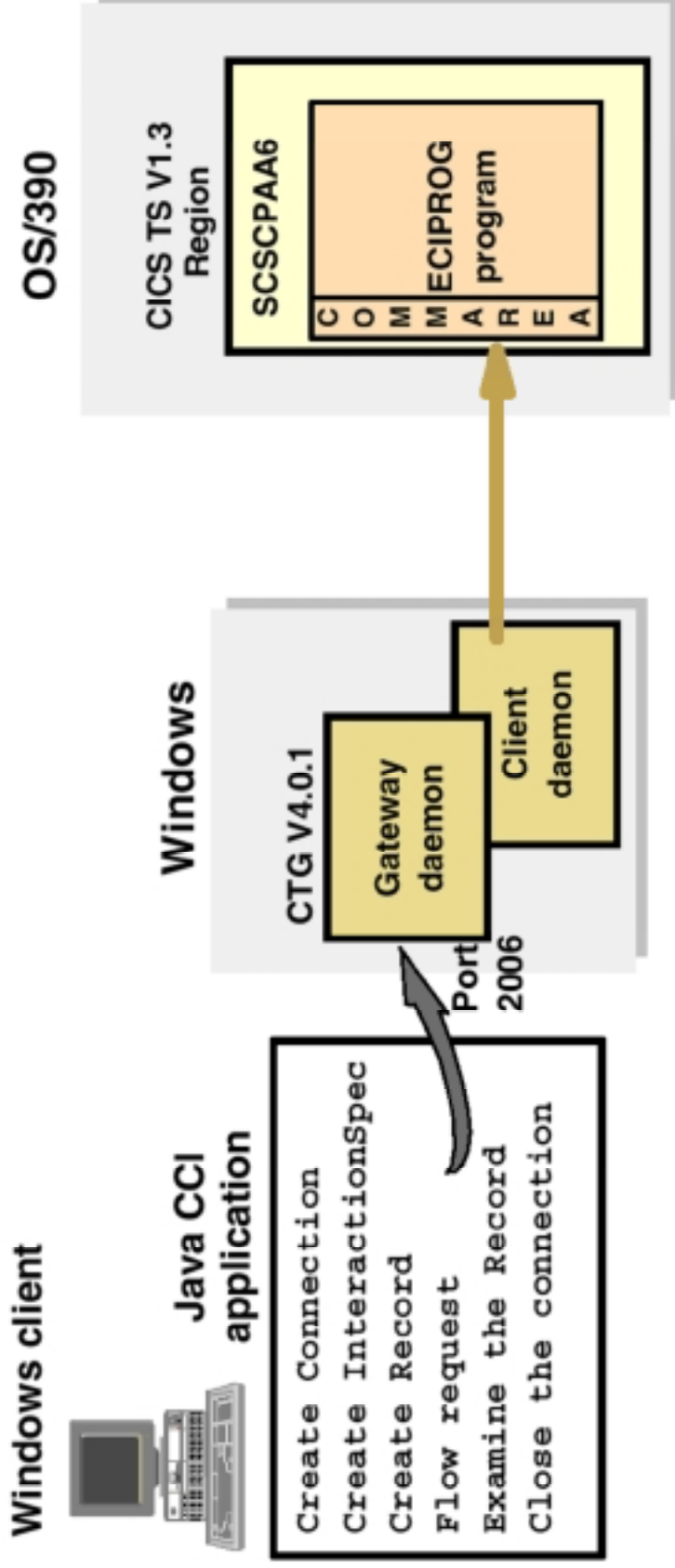
CICS Klienten Anbindung



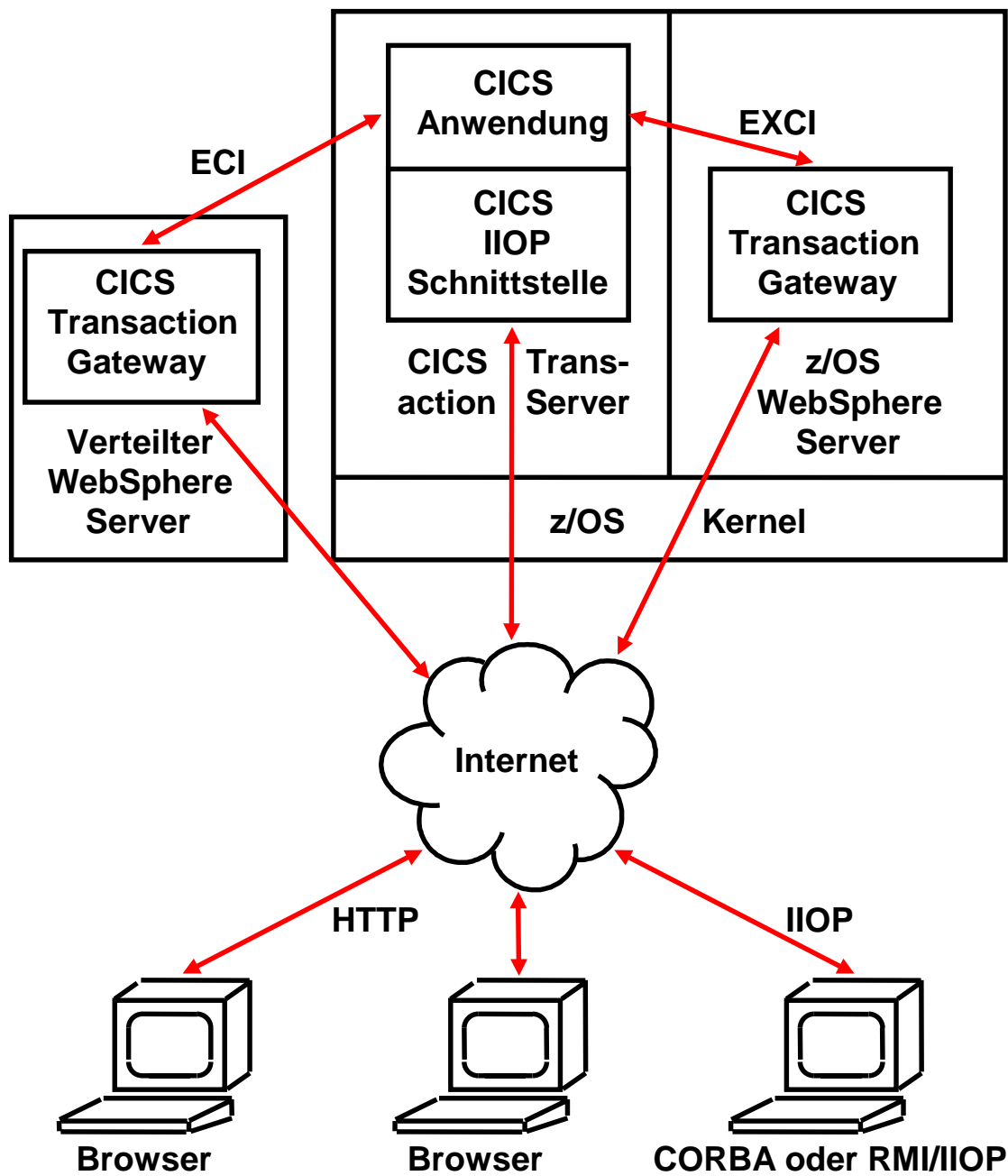
COMMAREA Communication Area

COMMAREA kann benutzt werden um Ein/Ausgabedaten zu übergeben:

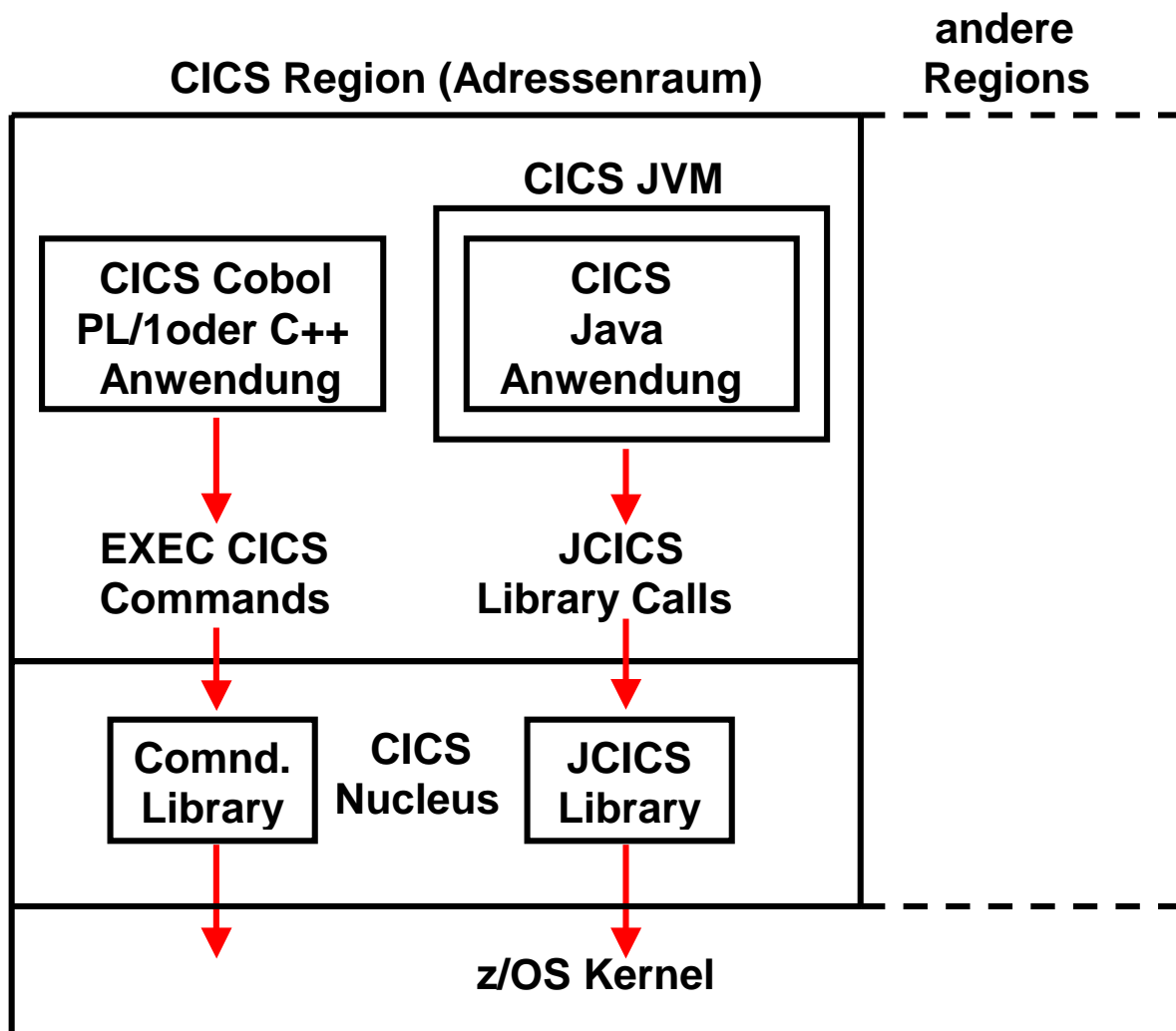
- An ein CICS Anwendungsprogramm im gleichen Adressenraum
- An ein CICS Anwendungsprogramm in einem anderen Adressenraum auf dem gleichen physikalischen Rechner,
- An ein CICS Anwendungsprogramm auf einem getrennten physikalischen Rechner unter Benutzung von EXEC CICS LINK (...)
- An ein nicht unter CICS laufendes Programm, z.B. ein GUI Prozess unter Verwendung der EPI Schnittstelle



CTG Scenario für eine ECI CCI Anwendung



Alternative CICS Zugriffe über das Internet

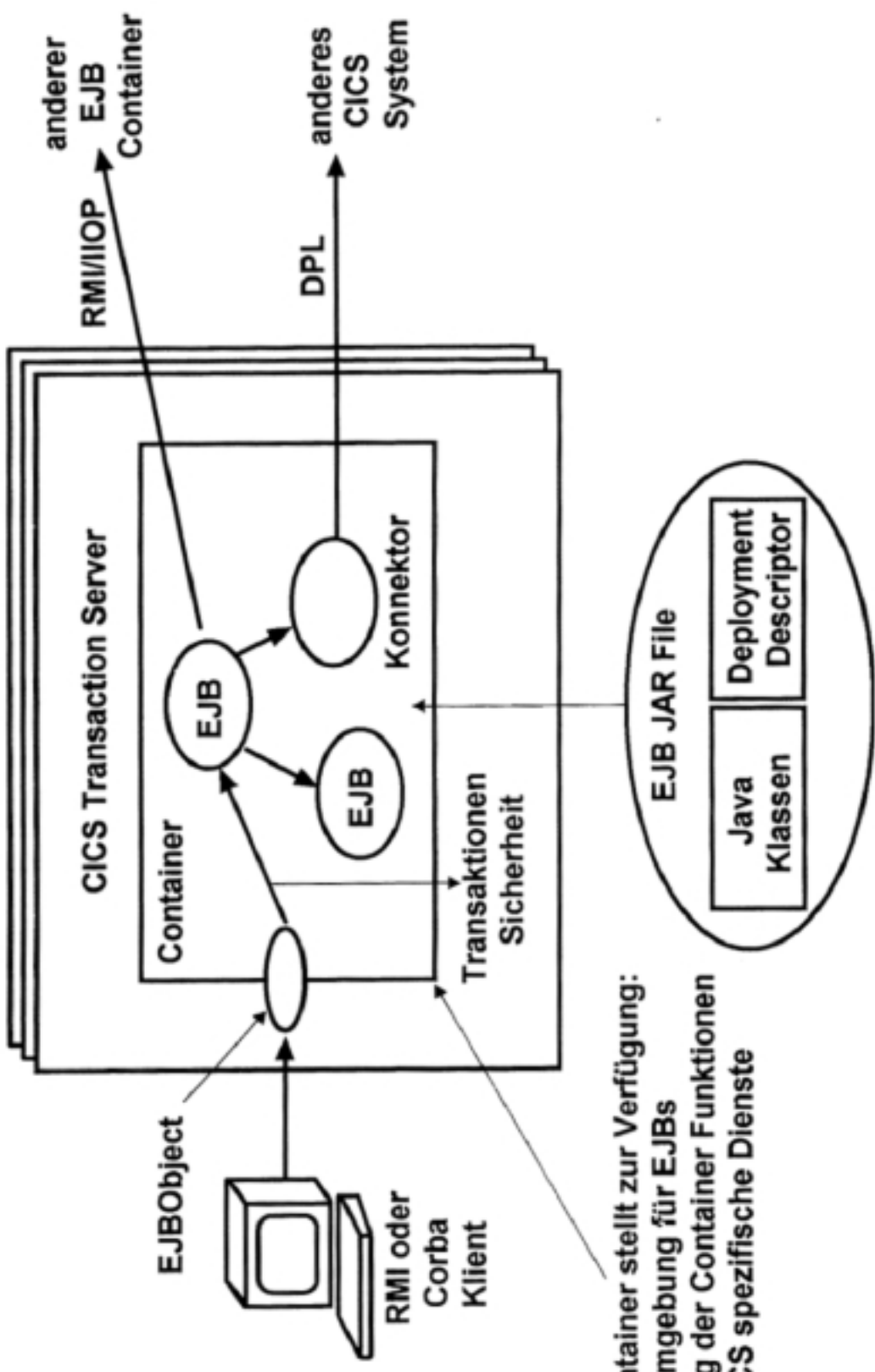


JCICS Java Anwendungsprogrammierung

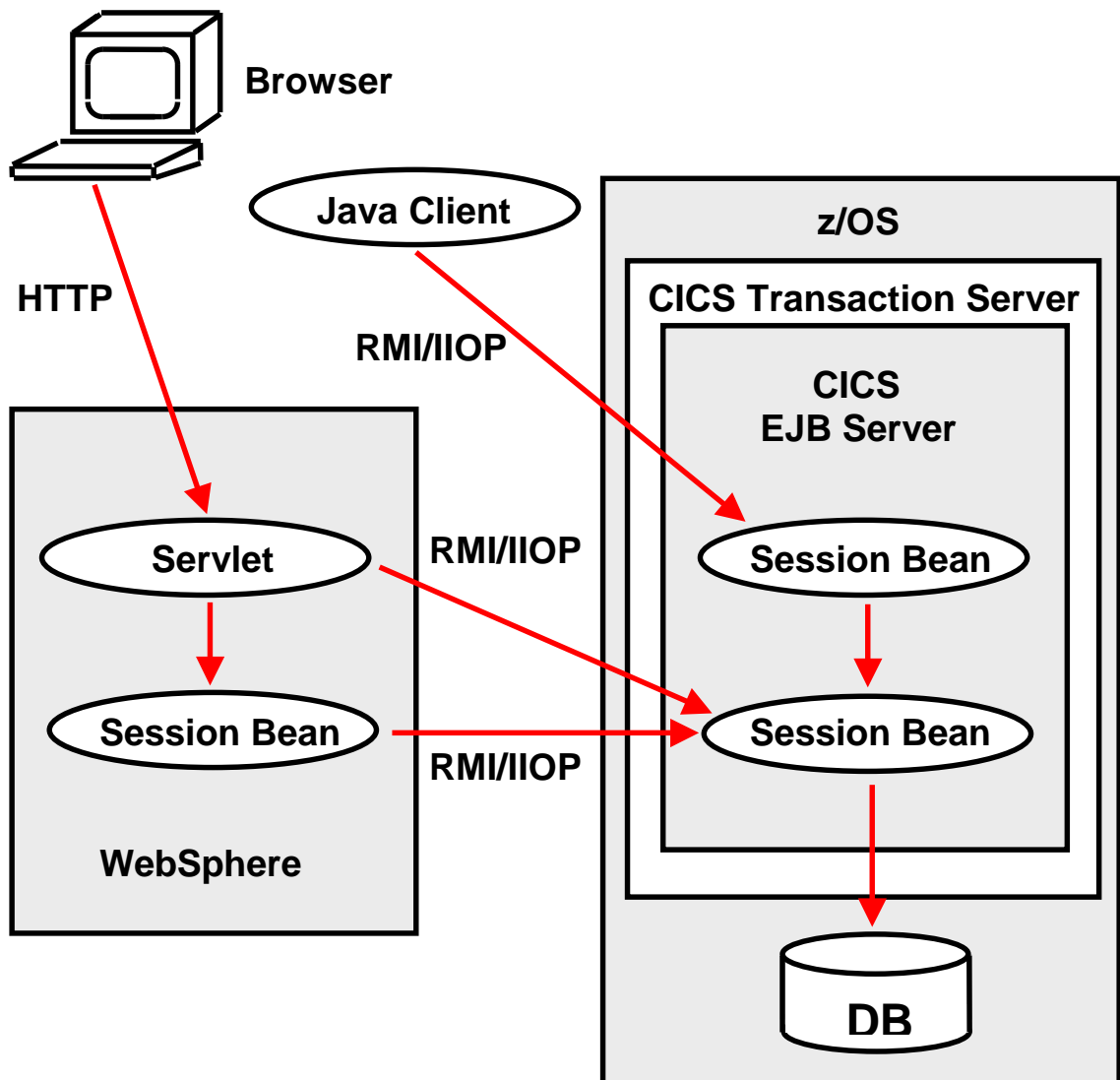
Java wird wie eine andere Programmiersprache eingesetzt. Die JCICS Library Calls ersetzen 1:1 die EXEC CICS Kommandos.

Keine Enterprise Java Beans

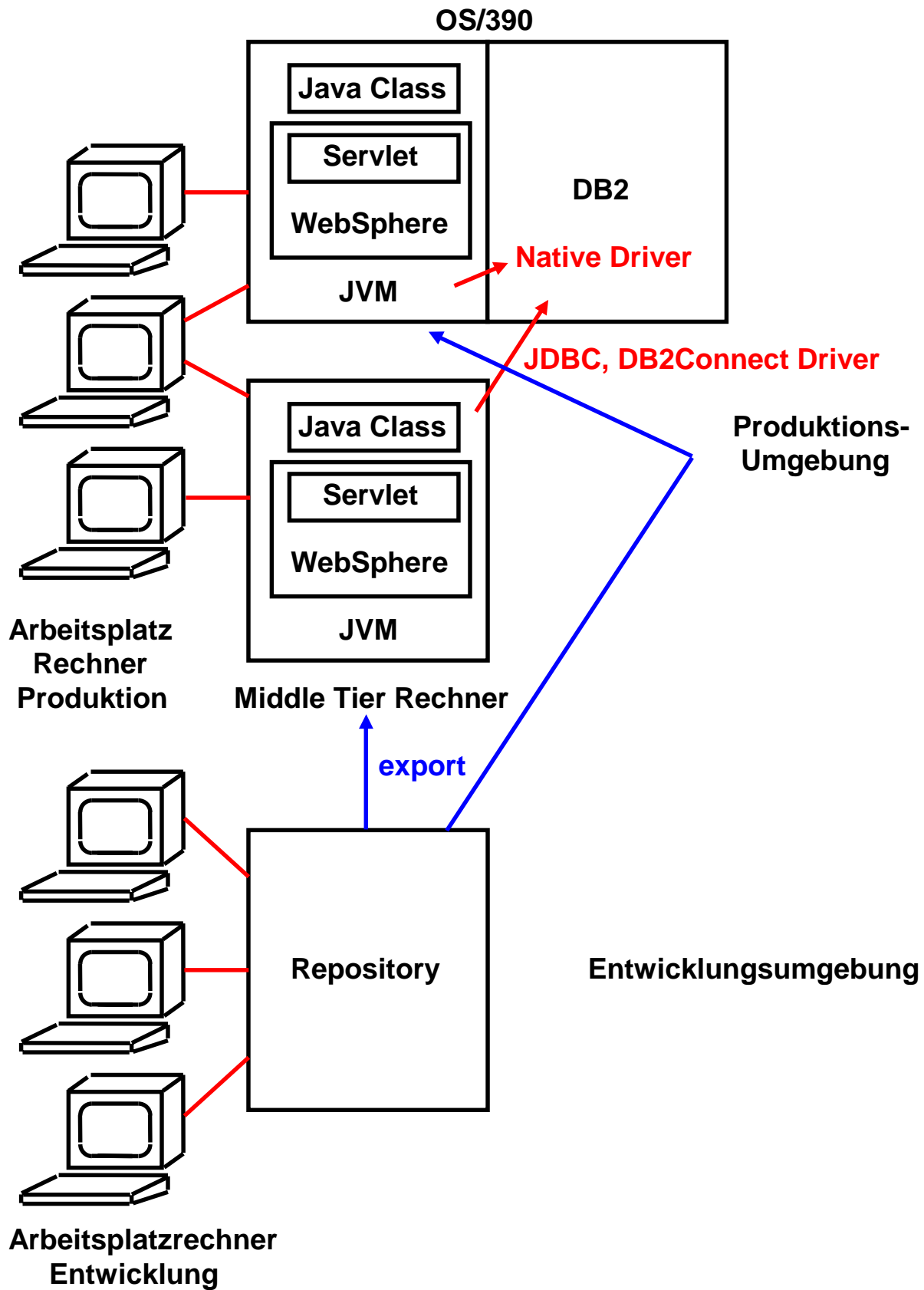
Java Application Development for CICS
<http://publib-b.boulder.ibm.com/abstracts/sg245275.html?Open>



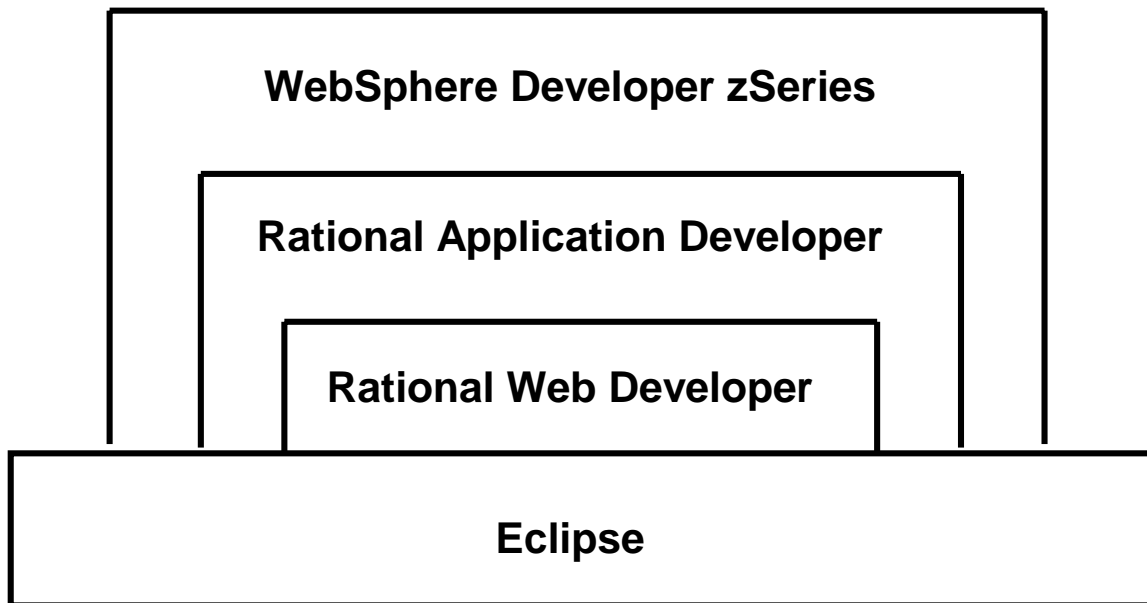
CICS Container stellt zur Verfügung:
 Laufzeitumgebung für EJBs
 Abbildung der Container Funktionen
 auf CICS spezifische Dienste



Benutzung von Enterprise Java Beans in einer CICS Anwendung



Trennung zwischen Entwicklung und Produktion



WDz Plug-in for Eclipse

Rational Web Developer

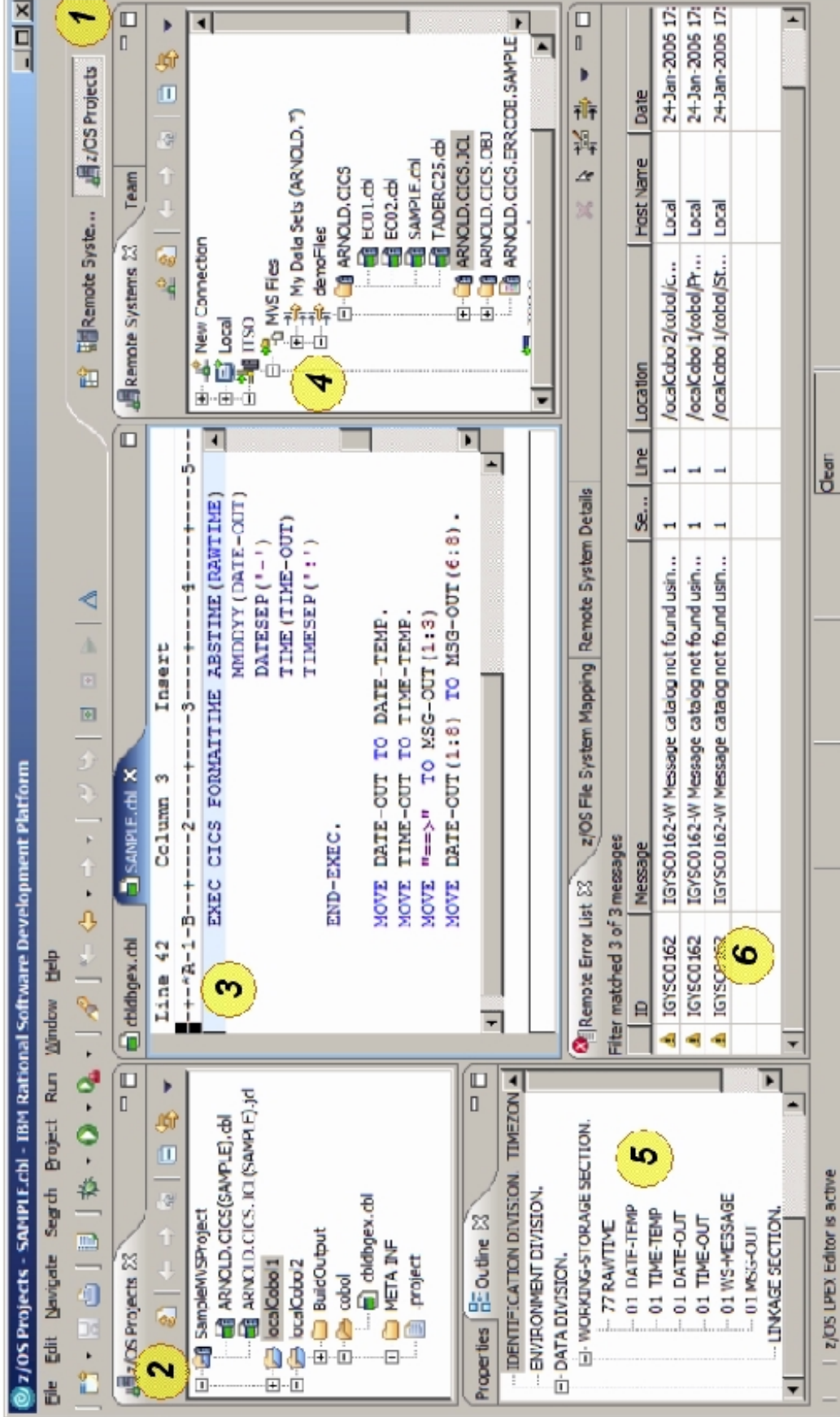
- **Web Development**
- **Web Services Development**
- **Rich Client Development**

Rational Application Developer

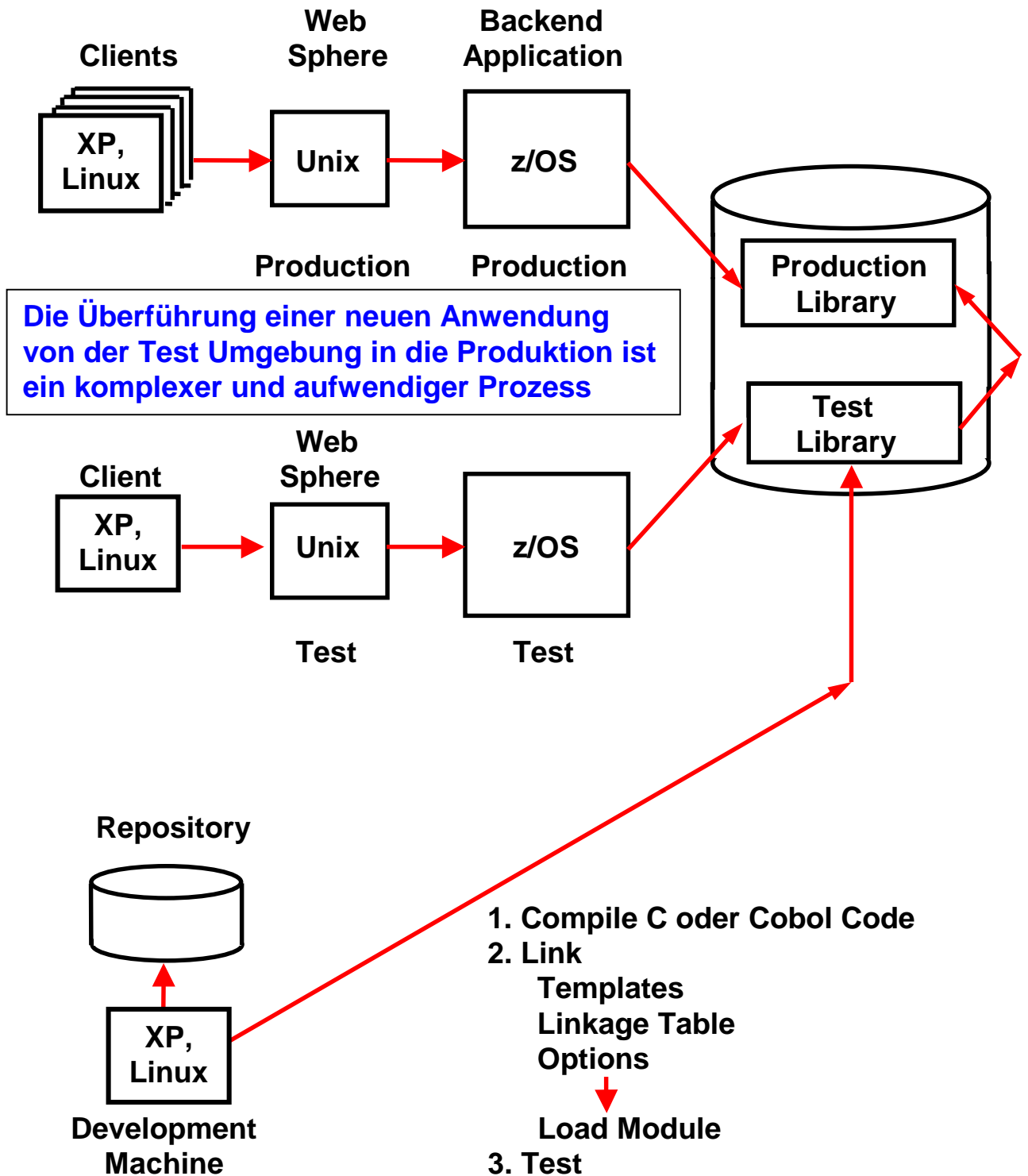
- **J2EE und EJB Development**
- **Component Testing**
- **Code Review und Runtime Analysis**
- **UML Visual Editors**

WebSphere Developer for zSeries (WDz)

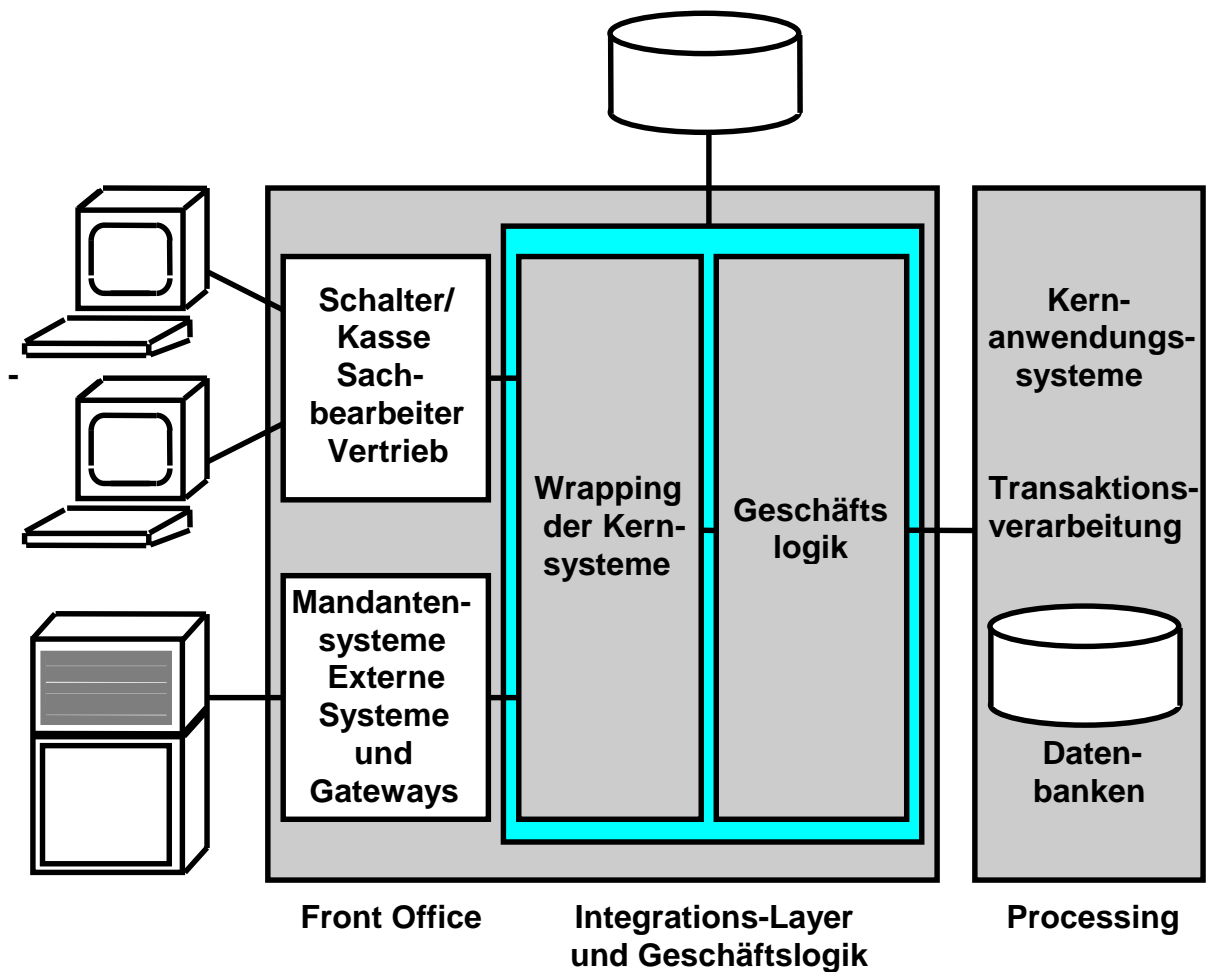
- **z/OS Application Development**
- **XML Services for z/OS**
- **BMS Map Editor**
- **CICS Application Development, Cobol, PL/1 and Java**
- **Stored Procedure Development, Cobol, PL/1 and Java**



1.Short-cut Icons zu anderen Perspektiven, 2.z/OS Projects view, 3. Editor, 4. Remote System Explorer view, 5.Outline view, 6. Tasks view



Überführung von der Entwicklung in die Produktion



Anwendungsarchitektur in Front-Office (mit Präsentationslogik) und Integrationsebene mit der Geschäftslogik sowie Processing mit der Transaktionsverarbeitung, wird verwendet um bestehende Kernanwendungssysteme durch den Integration-Layer zu wrappen

Geschäftsprozess

Beispiel: Überweisung der Betriebsrente an 100 000 ehemalige Daimler Mitarbeiter.

Komplexer Prozess: Viele Empfängerbanken, manche im Ausland, unterschiedliche Währung, unterschiedliche Steuerabzüge, Renten Erhöhungen und Kürzungen, Lohnpfändungen,

Viele einzelne Aktivitäten, die als Transaktionen ausgeführt werden (short running transactions).

Ersatz der manuellen Ablaufsteuerung durch einen Workflow Prozess.

STP-(Straight -Through-Processing-)Fähigkeit.

Literatur:

**Joachim Franz, Wilhelm G. Spruth:
Reengineering von Kernanwendungssystemen auf Großrechnern.
Informatik Spektrum, Band 26, Nr. 2, April 2003, S. 83-93.
<http://www.informatik.uni-leipzig.de/cs/pers/publish.html>**

Thomas Hornung: Entwurf und Implementierung einer transaktionalen

Geschäftsprozess - Anforderungen

Performance

Bis zu 5000 Transaktionen pro Sekunde

Mandantenfähigkeit

Anwendungstechnische Unterstützung für eigenständige, juristisch unabhängige Unternehmen. Erfordert z.B. getrennte physische Datenhaltung; eventuell Verschlüsselung von Transaktionsdaten; Ablaufverfolgung der Aufträge und Abrechnung der erbrachten Leistungen pro Mandant

Realtimefähigkeit

Direkte Bearbeitung der Transaktion , und Weiterleitung z.B. an Disposition und Buchung

Unterbrechungsfreier Betrieb

Unterbrechungsfreier Betrieb

Komponentenbasiertes System

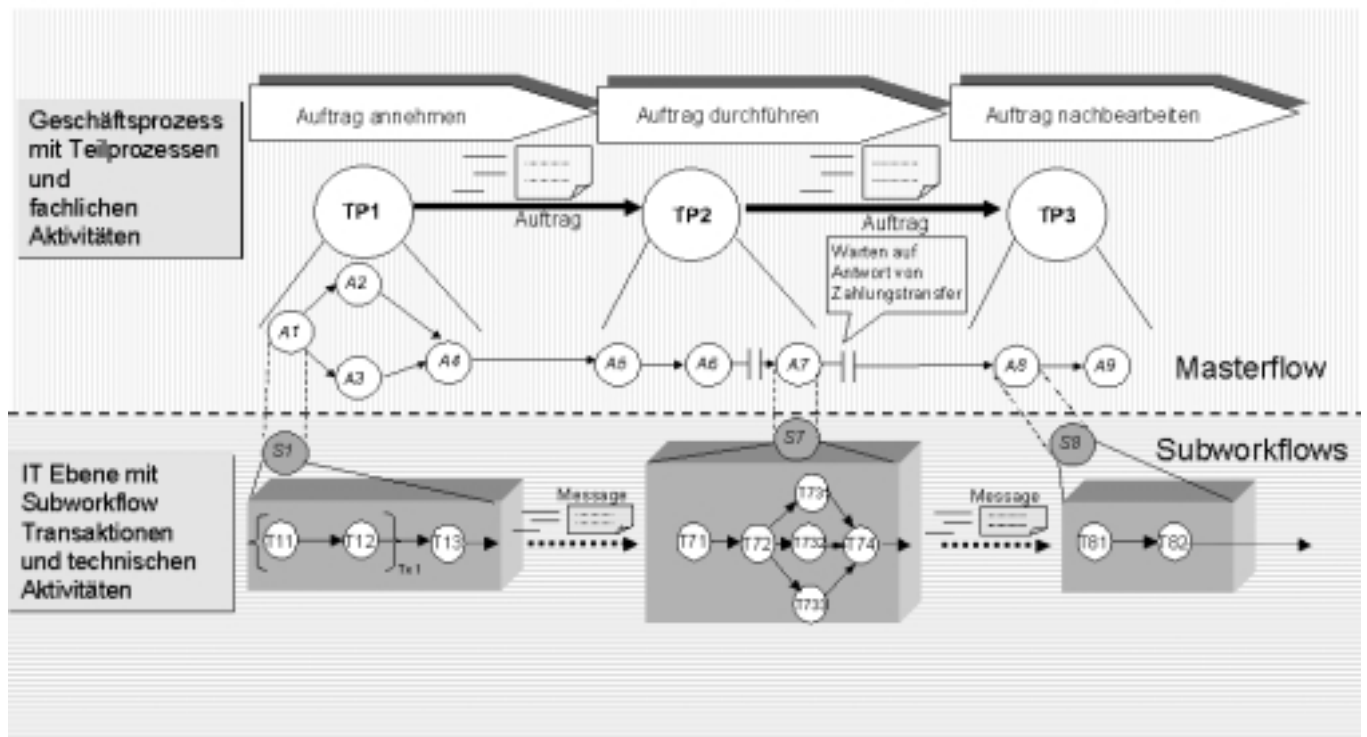
Hinzufügen, Verändern und Ersetzen von Banken oder Versicherungsprodukten im laufenden Betrieb

Standardsoftware

Anbindung bzw. Integration von Vendor software

Prozesssteuerung

SLA Steuerung, Transaktionssteuerung, Verarbeitungssteuerung



Transaktionsmaschine

Joachim Franz, Wilhelm G. Spruth:
 Reengineering von Kernanwendungssystemen auf Großrechnern.
 Informatik Spektrum, Band 26, April 2003

Aufteilung in Master Workflow und Subworkflow

Fachliche Aktivitäten

- A1: Auftrag fachlich prüfen**
- A2: Disposition für Auftrag prüfen**
- A3: Empfänger Banken bestimmen**
- A4: Ausführungszeiten bestimmen**
- A5: Zahlungspositionen, Konten, Bank, Land zusammenfassen**
- A6: Leitwege bestimmen**
- A7: Zahlungstransfer durchführen**
- A8: Zahlung fiskalisch verbuchen**
- A9: Auftrag archivieren und abschließen**

Technische Aktivitäten

Subworkflow 1

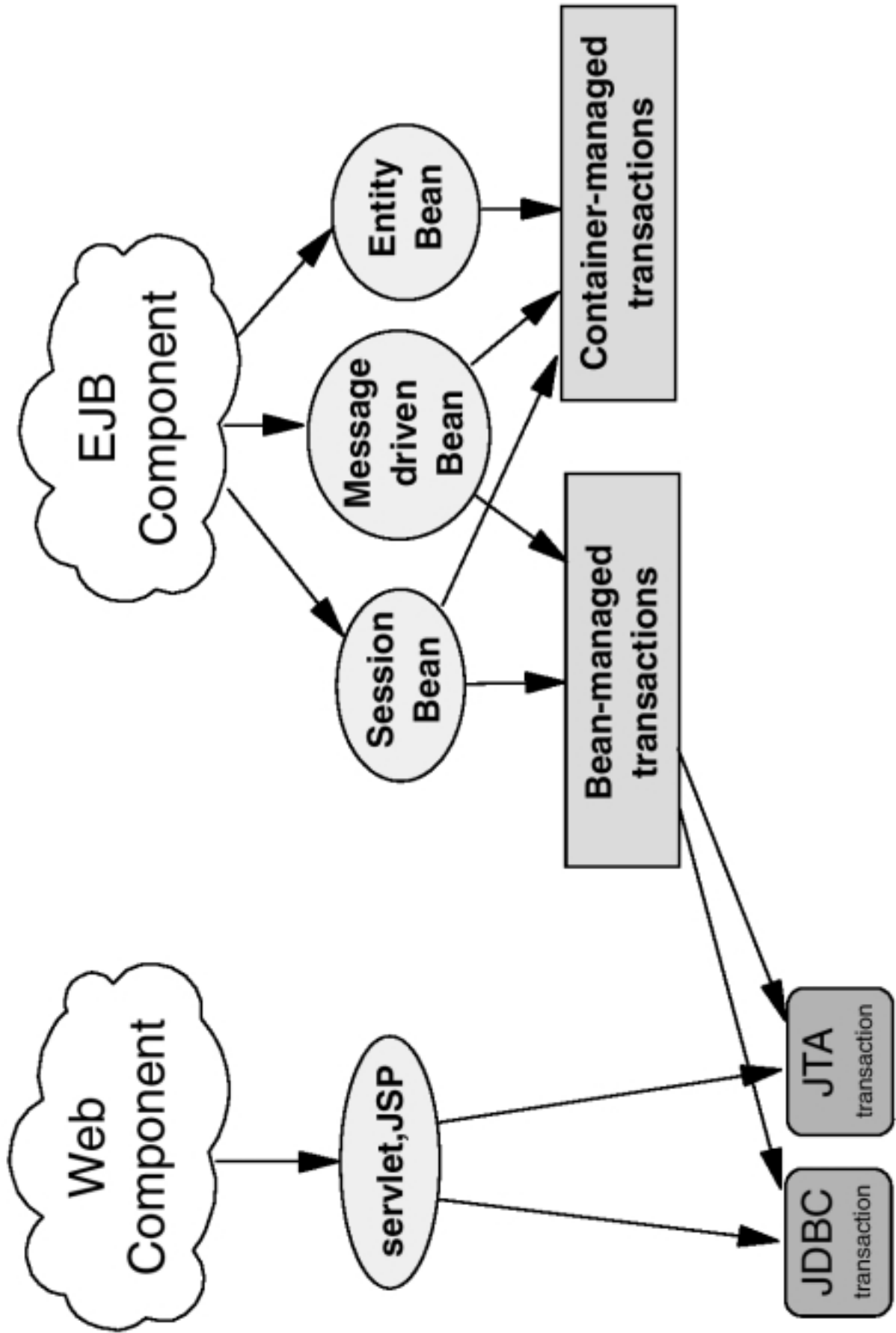
- T11: Auftragsprüfung auf Plausibilität**
- T12: Auftragsprüfung fachlich**
- T13: Status und Bestätigungsmeldung an Auftraggeber**

Subworkflow 7

- T71: Leitwegliste auf Vollständigkeit prüfen**
- T72: Zahlungstransfer vorbereiten**
 - T731: Führe Transfer für Land x1 (z.B. USA) und Bank y1 aus**
 - T732: Führe Transfer für Land x2 und Bank y2 aus**
 - T732: Führe Transfer für Land x3 und Bank y3 aus**
- T74: Buchungsaufträge erstellen**

Subworkflow 8

- T81: Ergebnisauswertung des Zahlungsverkehrs**
- T82 Positionen fiskalisch verbuchen**



SOA

Service Oriented Architecture

Business Prozesse flexibel miteinander verbinden.

Hierfür sind Webservices

- **SOAP**
- **WSDL**
- **UDDI**

als Kommunikationsbausteine vorgesehen.

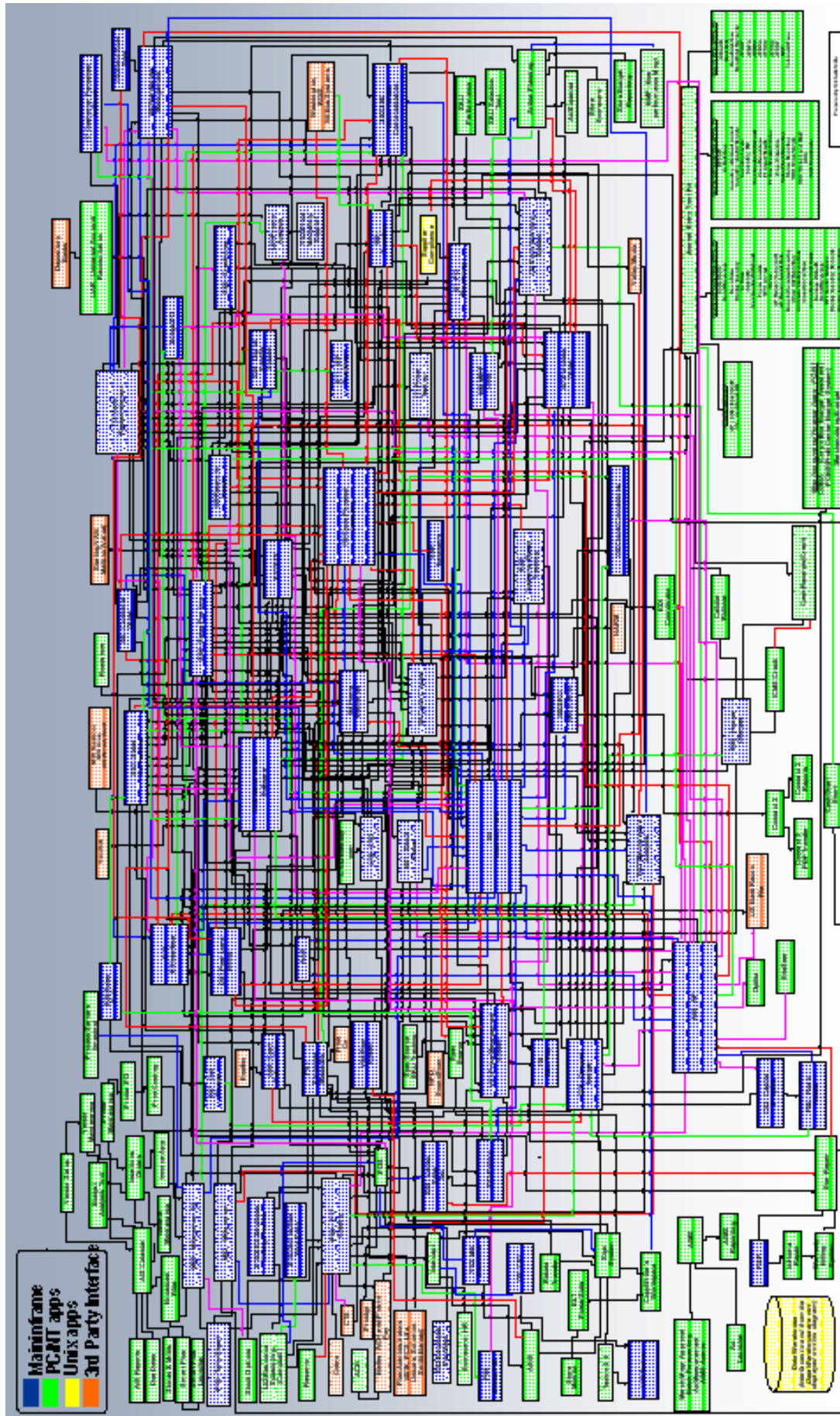
Existierende Anwendungen (Cobol und in Zukunft Java) als Bausteine rekonfigurieren.

Business Prozesse flexibel und mit geringem Aufwand an sich ständig ändernde Anforderungen anpassen.

Legacy-Systeme

Legacy-System ist ein Begriff zur Bezeichnung einer "historisch gewachsenen Alt-Anwendung". In größeren Unternehmen sind das häufig Individualentwicklungen, die auf Großrechnern unter den z/OS und OS/390 Betriebssystemen ablaufen. In etwa 80 % aller Fälle sind dies transaktionale Anwendungen, die unter Transaktionsmonitoren wie CICS und IMS ablaufen.

Diese unternehmenskritischen Kernanwendungen sind meist über Jahrzehnte entstanden und bilden das Erbe mehrerer Generationen von Softwareentwicklern. Es wird geschätzt, dass etwa 10 Millionen Mannjahre in die Entwicklung von unternehmenskritischen OS/390-Anwendungen unter CICS investiert wurden. Das bedeutet eine Investition von etwa einer Billion US-Dollar in OS/390-Anwendungssoftware unter CICS. Die meisten dieser Programme sind in Cobol geschrieben. Die existierende Menge an Cobol-Programmen besteht aus etwa 180 Milliarden Code-Zeilen mit einer jährlichen Zuwachsrate von 5 Milliarden Code-Zeilen.



Application Architecture for a Consumer Electronics Company

Diplomarbeiten Berufspraktikum Projektarbeit

Dr. Paul Herrmann, Prof. Wilhelm Spruth

An der Universität Leipzig sind Themen im Umkreis z/OS - Internet Integration zu bearbeiten.

Diplomarbeitsthemen, Praktika in der Industrie (nach Absprache)

- **im EJB - OS/390 Umfeld**
- **im Linux - S/390 Umfeld**
- **im RACF WebSphere Umfeld**
- **Im Websphere Enterprise Developer (WSED) Umfeld**

Themenstellung nach Absprache

Nach Voranmeldung sind wir ggf. in der Lage, Praktika und Diplomarbeiten in der Industrie zu vermitteln. Derzeitig laufen mehrere Arbeiten in den IBM Entwicklungslaboratorien Böblingen.

Im Rahmen eines größeren Industrieforschungsprojektes zwischen Prof. Bogdan, Prof. Spruth und den IBM Entwicklungslaboratorien Böblingen auf dem Gebiet des z/OS Work Load Managers sind kürzlich mehrere Diplomarbeiten erfolgreich abgeschlossen worden.

Beispiele für Prüfungsfragen

Nennen sie einige der Faktoren, die für einen Großrechner in der wirtschaft wichtig sind.

Die zSeries Rechner haben eine ganze Reihe von zukunftsorientierten Hardware- und Software Eigenschaften, die auf anderen Rechnern heute (noch) nicht zu finden sind. Nennen sie einige Beispiele.

Was ist EBCDIC.

Benennen Sie einige Unterschiede in denen sich die S/390 Hardware Architektur von anderen Rechnern unterscheidet.

Was wissen Sie über das S/390 I/O Subsystem? Was ist der Unterschied zwischen einem S/390 Kanal und einem SCSI Anschluss?

Beschreiben Sie in jeweils 1 kurzen Satz was sich hinter den folgenden OS/390 Subsystemen verbirgt: JES, TSO, USS, CICS, IMS, DB2, RACF,

Manche Dateien sind unstrukturiert. Andere bestehen aus Datensätzen. Wie sieht das bei Windows, Unix und z/OS aus ?

Welche Aufgabe bewältigt DFSMS (Data Facility Storage Management System)

Was ist der Unterschied zwischen OS/390 Unix System Services und S/390 Linux ?

Was sind die Eigenschaften von Virtuellen Maschinen (PR/SM)? Welche Rolle spielen sensitive Maschinenbefehle ? Was sind Host- und Gast-Status ?

Werden realer Hauptspeicher, Ein-/Ausgabegeräte und Kanäle den einzelnen virtuellen Maschinen fest zugeordnet ?

Was ist eine LPAR (Logical Partition) ?

Was ist ein Sysplex? Können die Knoten eines Sysplex auch SMP's sein?

Welche Aufgaben hat eine Sysplex Coupling Facility (CF)?

Wie werden Locks durch eine CF verwaltet?

Welche Funktion hat derGoal oriented Work Load Manager

Welches sind die wichtigsten Bestandteile eines Transaktionsmonitors wie CICS

Was ist COMMAREA ?

Diskutieren Sie die Aufgabe von Presentation Services in einem Transaktionsverarbeitungssystem. Welche Aufgabe haben Maps (Screens)?

Können CICS Screens durch Java Server Pages implementiert werden?

**Was ist Message Based Queuing bzw Message oriented Middleware ?
Erläutern sie kurz die Funktion von MQseries .**

Was ist ein Web Application Server? Wodurch unterscheidet er sich von einem Web Server wie z.B. Apache?

Was sind Konnectoren? Können sie mit EJB´s implementiert werden ? Was ist die Java Connection Architektur (JCA) ?

Werden auf einem Rechner mit einer einzigen CPU mehrere Transaktionen gleichzeitig verarbeitet ? Warum ?

Wodurch unterscheidet sich Multiprogramming mit getrennten Prozessen von der Nutzung von Threads ? Was sind Vor- und Nachteile ?

Was ist ein Business Prozess?