

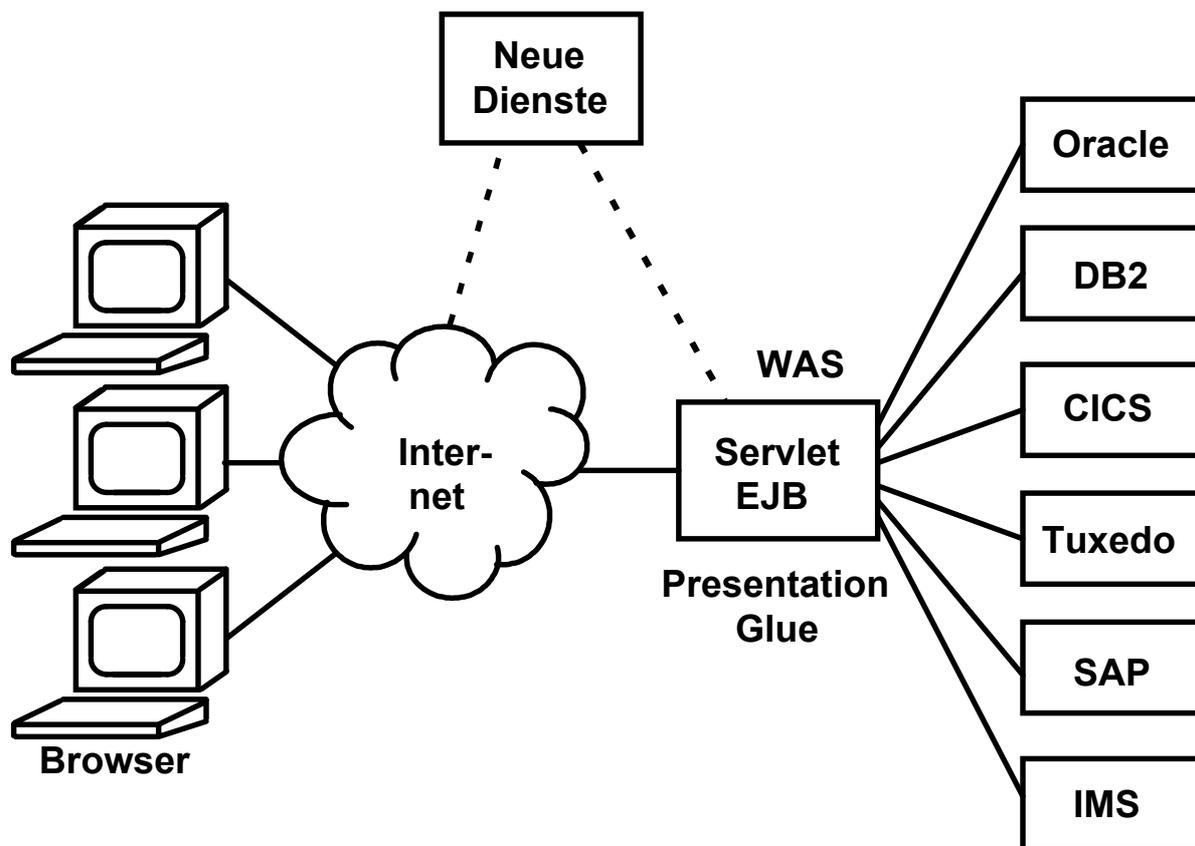
# **Internet Anwendungen unter OS/390**

**Dr. rer. nat. Paul Herrmannn  
Prof. Dr.-Ing. Udo Kebschull  
Prof. Dr.-Ing. Wilhelm G. Spruth**

**WS 2004/2005**

**Teil 7**

**z/OS Internet Integration**



## Client/Server Aufgabenstellung

- Browser orientierter Web Zugang
- Datenhaltung in existierenden Datenbanken
- Dominierender Anteil der Business Logik in existierenden Transaktionsprogrammen und/oder Stored Procedures
- Neue Software (z.B. EJBs) stellen Querverbindungen zwischen existierenden Komponenten her (Glue)
- System Management - TCO

# **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**
- **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 available as a library and 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.

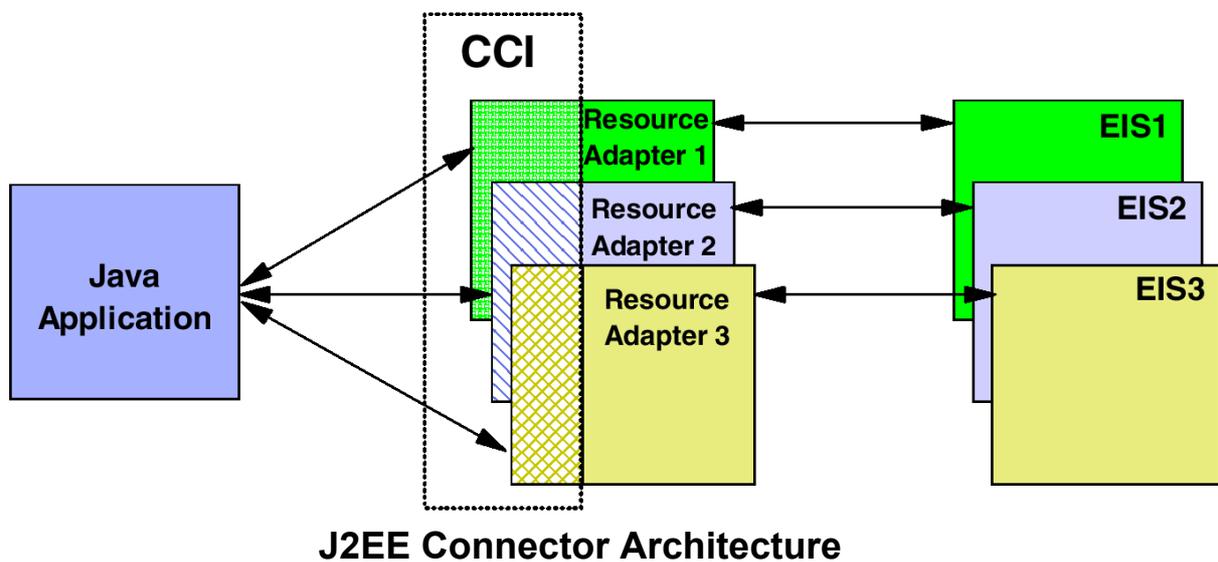
**Für CICS existieren:**

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

**CTG enthält ECI RA, EPI RA und CCI, sowie weitere Zugriffsmechanismen ausserhalb der JCA**

# JCA Common Client Interface CCI

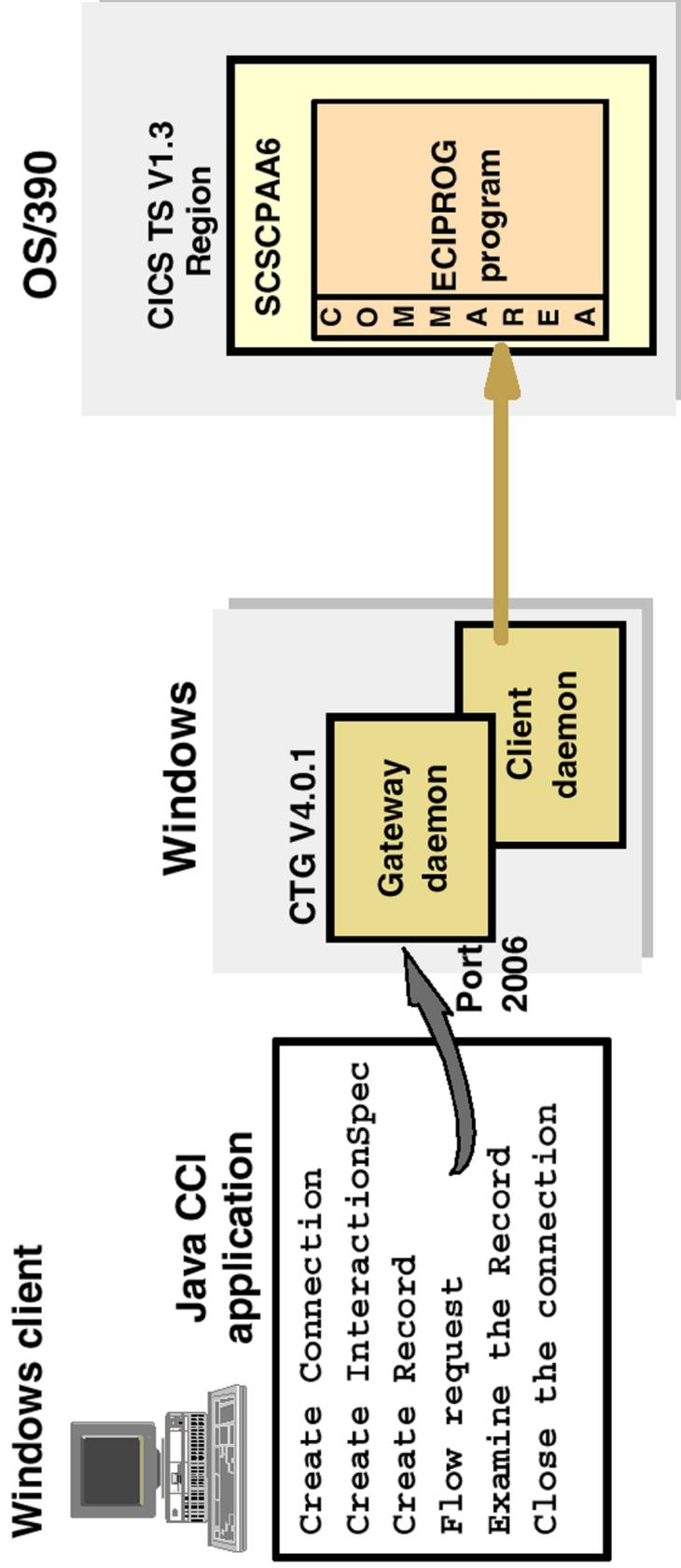
Die Common Client Interface (CCI) definiert eine Standard Client API. Die CCI ist eine EIS unabhängige API. Anwendungen können mit einer identischen API unterschiedliche Resource Adapter verwenden. Ähnlich JDBC Interface



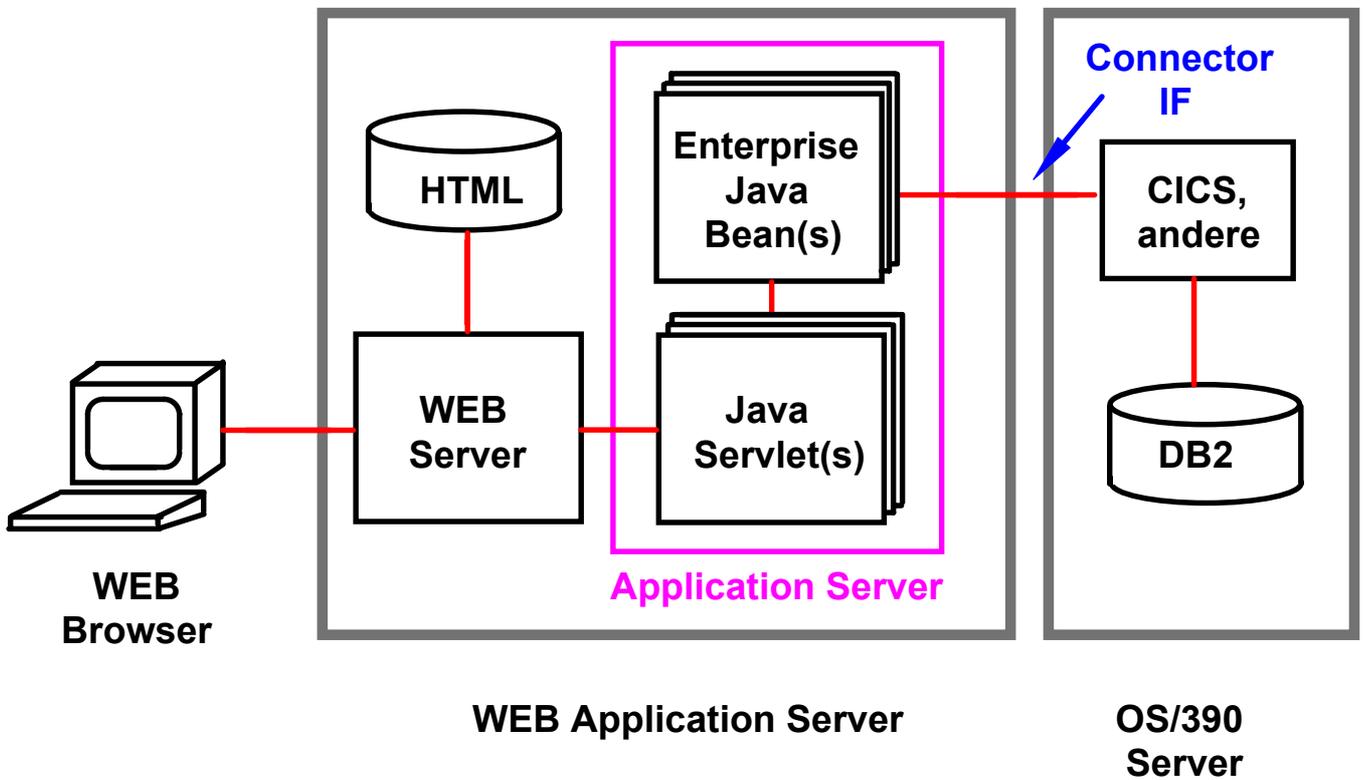
CCI      Common Client Interface  
EIS      Enterprise Information System

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.



**CTG Scenario für eine ECI CCI Anwendung**



## Web Application Server als Presentation und Integration Server

### WebSphere Funktionen unter OS/390

- Parallel Sysplex
- Workload Manager
- RACF
- Crypto
- Common Connector Framework
- Virtuelle Server

**Nutzung von Java für  
Business-Logik Backend  
Anwendungen**

**Alternative zu CICS – Cobol  
J2EE Standard**

**Wahrung der ACID Eigenschaften:  
Atomicity, Consistency, Isolation,  
Durability**

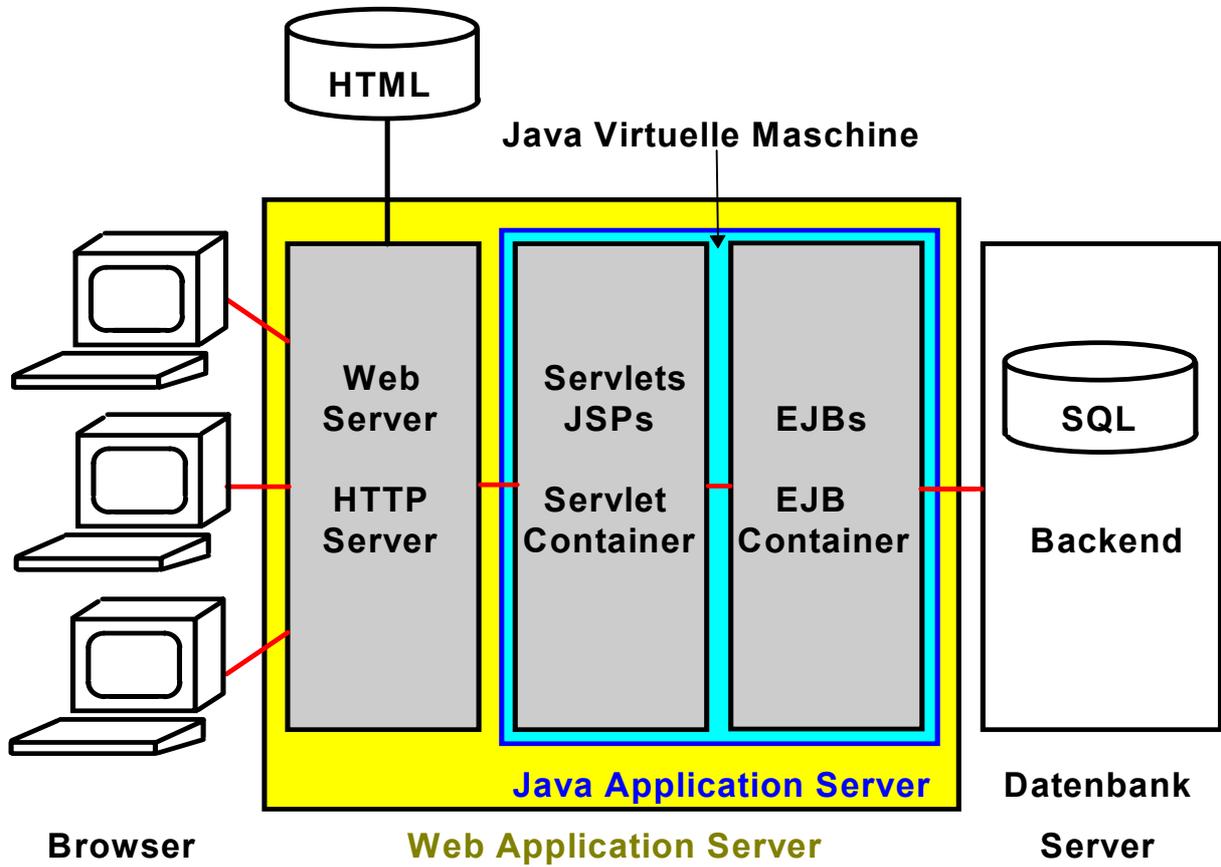
**Java Transaktionsverarbeitung  
unter dem z/OS Betriebssystem**

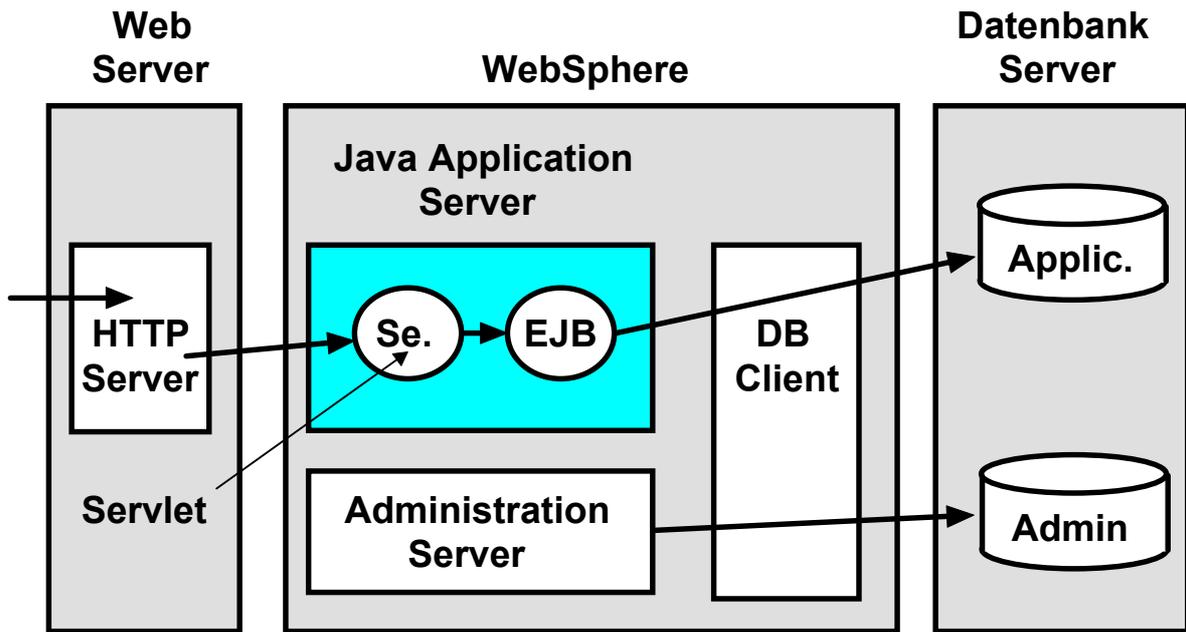
**Persistent Reuseable  
Java Virtual Machine**

**PRJVM**

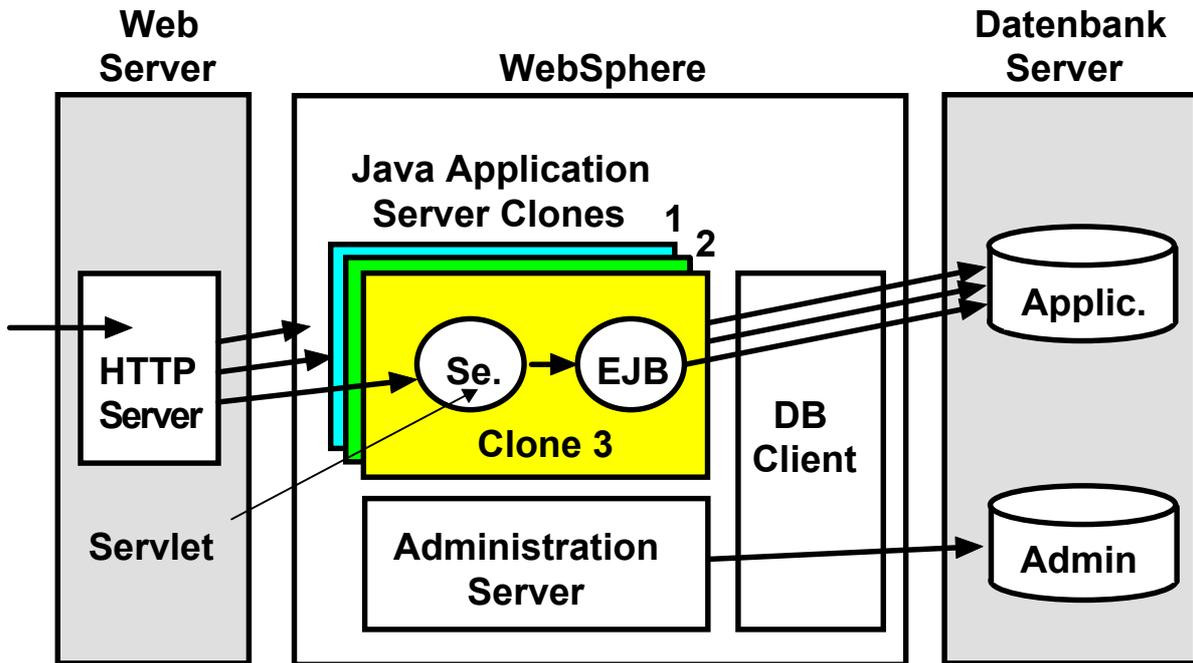
## **Software-Projekt für Finanzämter gescheitert**

**Das millionenteure Software-Projekt Fiscus, das den 650 Finanzämtern spätestens ab 2006 einheitliche Programme bringen sollte, steht nach einem Bericht des Focus endgültig vor dem Aus. Die Finanzminister von Bund und Ländern wollten die 1991 begonnene gemeinsame Software-Entwicklung am kommenden Freitag in einer Sondersitzung für gescheitert erklären. Die Programmiersuche haben den Steuerzahler nach internen Schätzungen bislang zwischen 250 und 900 Millionen Euro gekostet.**

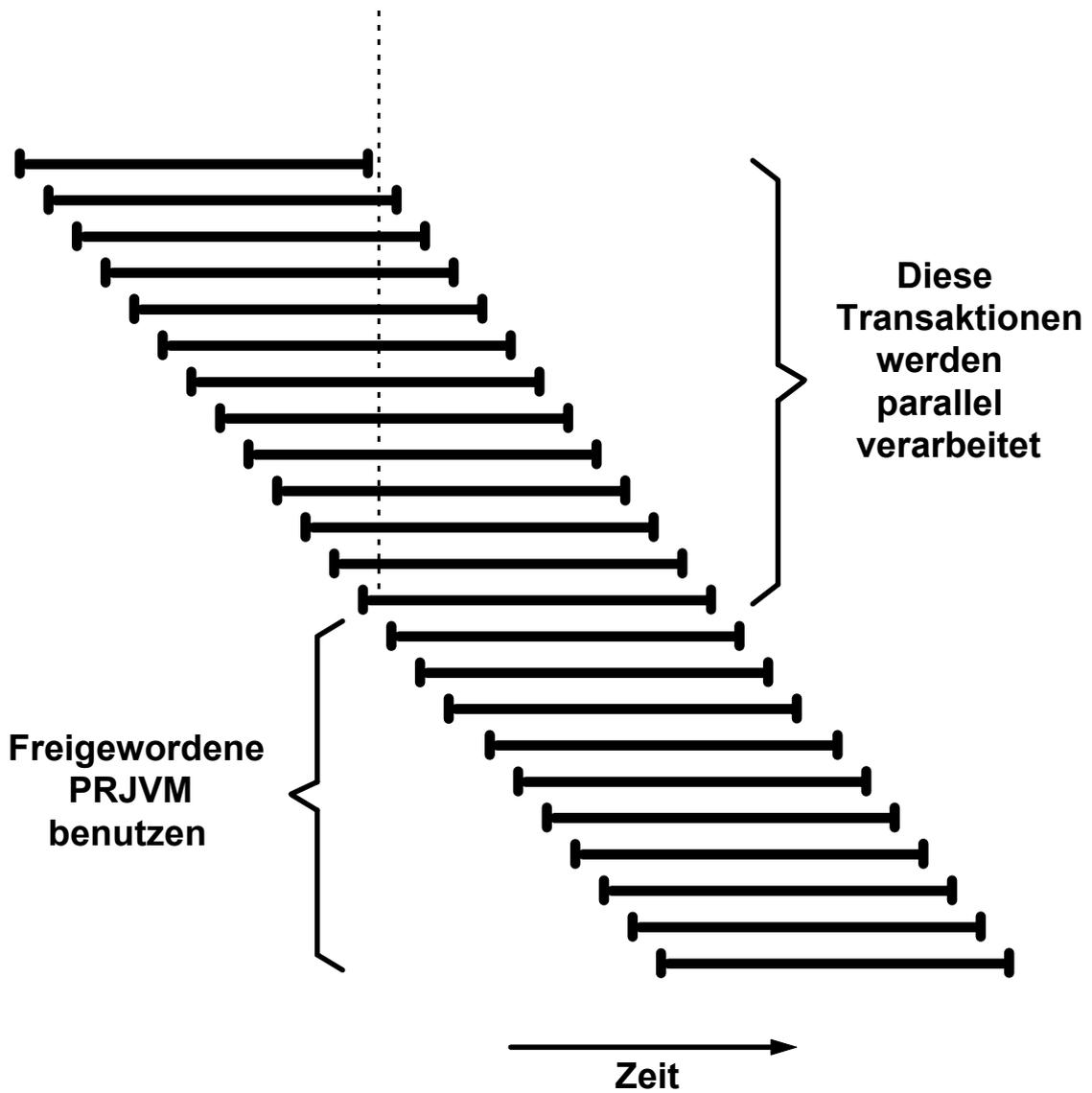




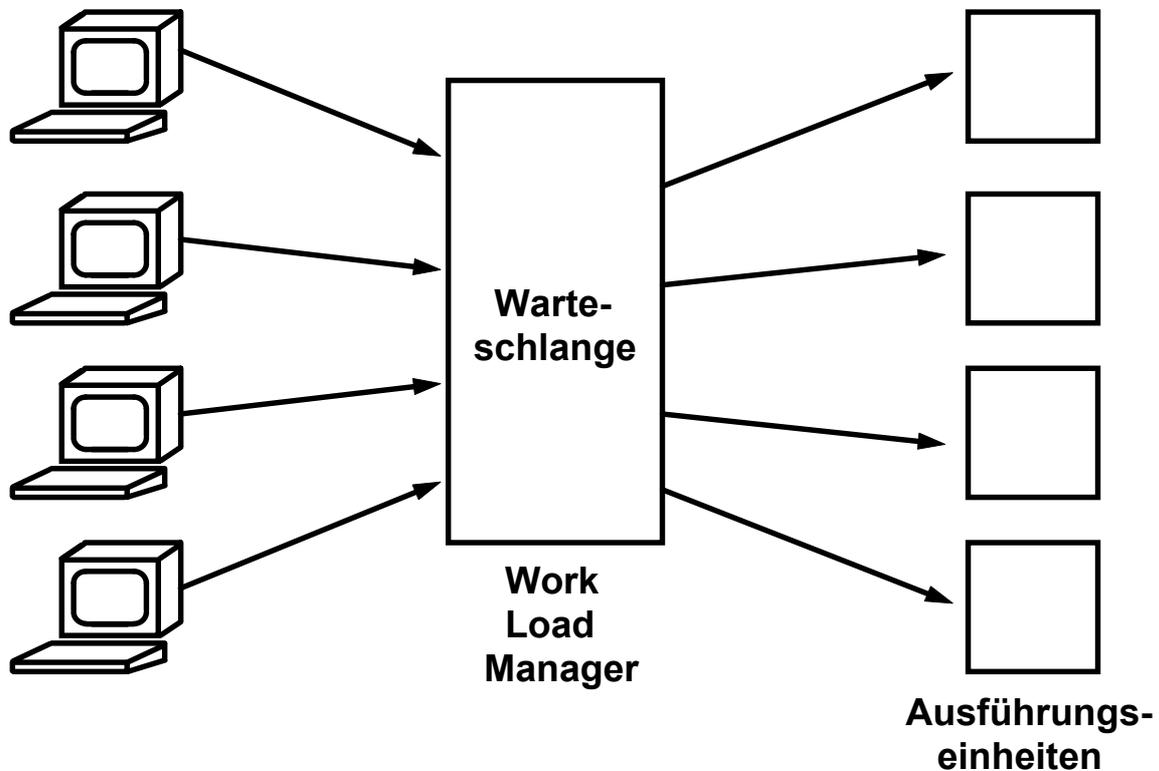
**Web Application Server**



**Mehrfache Clones des Java Application Servers,  
jeder mit einer eigenen Java Virtuellen Maschine**



## Parallele Verarbeitung von Transaktionen



## Lastverteilung

Eintreffende Transaktionen werden zunächst in einer Warteschlange zwischengespeichert, und dann auf eine von mehreren Ausführungseinheiten verteilt.

Ausführungseinheiten können sein:

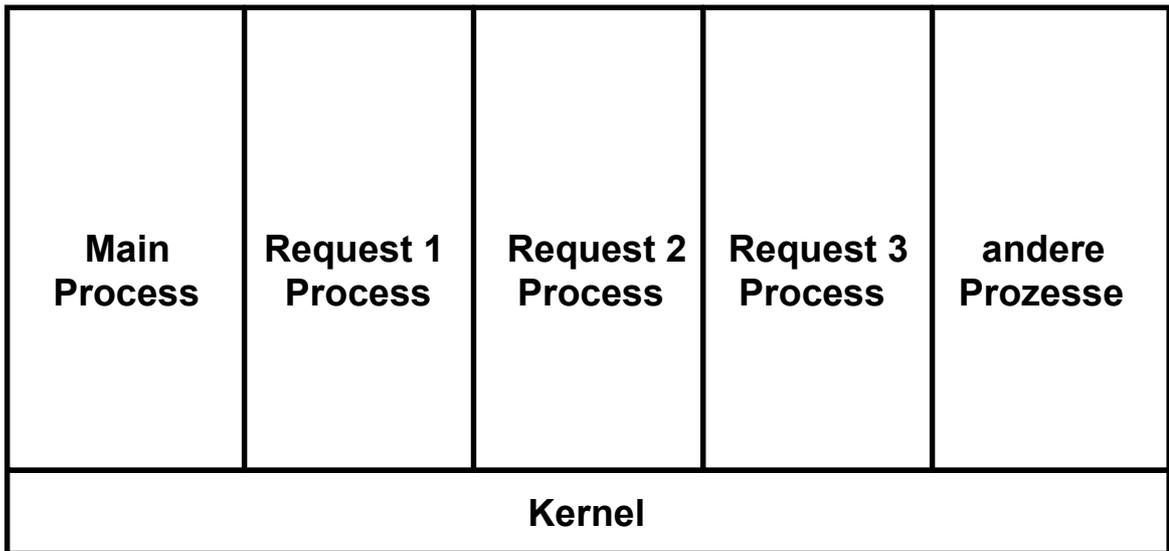
- individuelle Rechner, 1 Prozess/Rechner (unwahrscheinlich)
- individuelle Prozesse
- individuelle Threads innerhalb eines Prozesses

**Keine Performanz ohne  
Multiprogrammierung**

**Nutzung von Java Threads  
innerhalb einer Java Virtuellen  
Maschine**

**ACID Bedingungen einhalten**

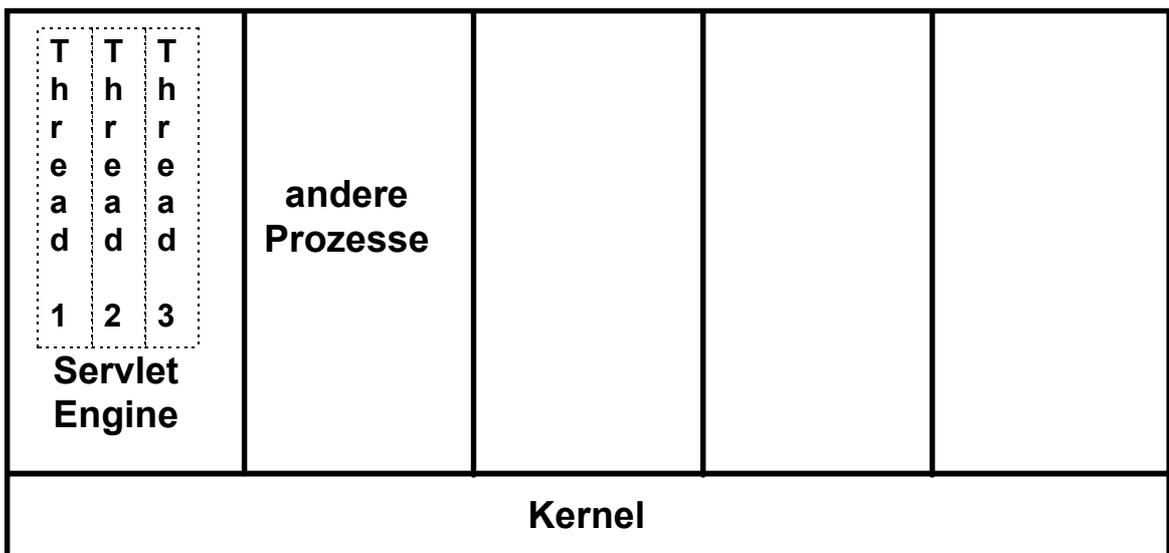
FF..FF



00..00

## Ansatz mit mehreren Prozessen

FF..FF



00..00

## Thread Ansatz

Ein Hochleistungs-Transaktionssystem verarbeitet in jedem augenblick Hunderte oder Tausende von Transaktionen gleichzeitig und parallel. Zwei Alternativen:

- 1 Prozess pro Transaktion (Beispiel SAP R/3)
- 1 Thread pro Transaktion (Beispiel CICS unter z/OS)

# Nutzung der Java Threads

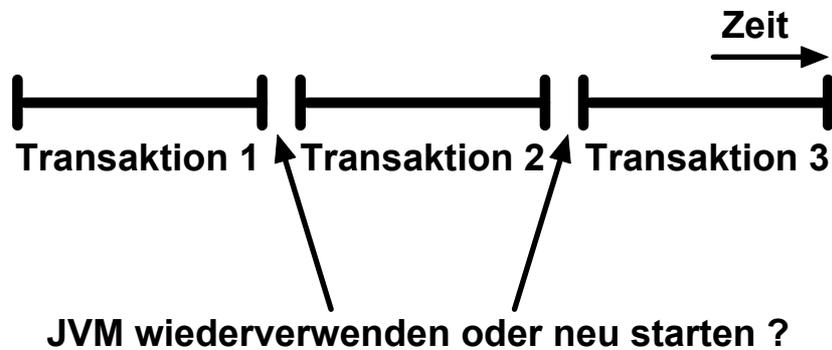
**The existing application isolation mechanisms, such as class loaders, do not guarantee that two arbitrary applications executing in the same instance of the JVM will not interfere with one another. Such interference can occur in many places. For instance, mutable parts of classes can leak object references and can allow one application to prevent the others from invoking certain methods. The internalized strings introduce shared, easy to capture monitors. Sharing event and finalization queues and their associated handling threads can block or hinder the execution of some application. Monopolizing of computational resources, such as heap memory, by one application can starve the others.**

*Grzegorz Czajkowski, Laurent Daynès:  
Multitasking without Compromise: a Virtual Machine Evolution.  
<http://research.sun.com/projects/barcelona/papers/oopsla01.pdf>*

**Java gives the virtuoso thread programmer considerable freedom, but it also presents many pitfalls for less experienced programmers, who can create complex programs that fail in baffling ways.**

*Bo Sandén: Coping with Java Threads. IEEE Computer, Vol. 37, Nr. 4, April 2004, p. 20.*

# Ausführung einer Folge von Transaktionen



Eine Transaktion die Ausführung der Folge-Transaktion beeinflussen, indem sie den Zustand (State) der JVM ändert. Beispiele für eventuelle sicherheitskritische Reste der vorhergehenden Transaktion

- überschriebene statische Variablen
- Starten von Threads
- geladene native Bibliotheken.

**Klassischer Ansatz:** Für jede Transaktion wird eine neue JVM gestartet und nach Abschluss der Transaktion wieder beendet.

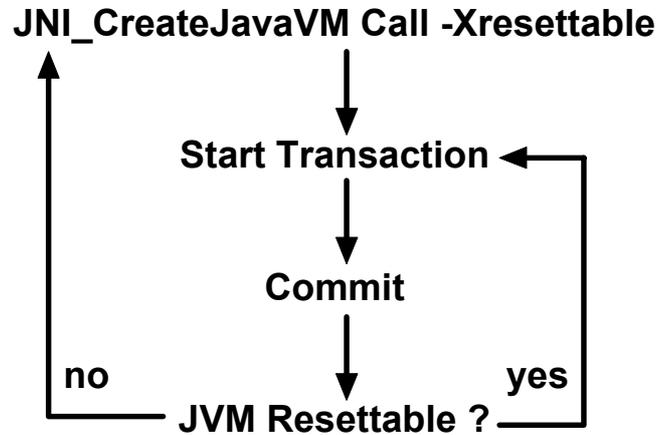
Das Hoch- und Herunterfahren einer JVM hat jedoch einen erheblichen Zeitaufwand zur Folge. Bei jeder Initialisierung einer JVM werden

- 60 System Klassen geladen,
- 700 Array-Objekte und
- 1000 non-Array-Objekte allokiert und angelegt.

Pfadlänge bis zu 100 Millionen Maschinenbefehle, ca. 1 Sekunde Verarbeitungsdauer.

**Lösung:** JVM mit einer zusätzlichen Reset Methode ausstatten.

**Persistent Reusable Java Virtual Machine - PRJVM**



Es ist möglich, dass die Ausführung einer Transaktion die PRJVM in einen Zustand *unresetable* versetzt. Dies kann auf einem asozialen, jedoch legalen Verhalten der Tansaktion beruhen.

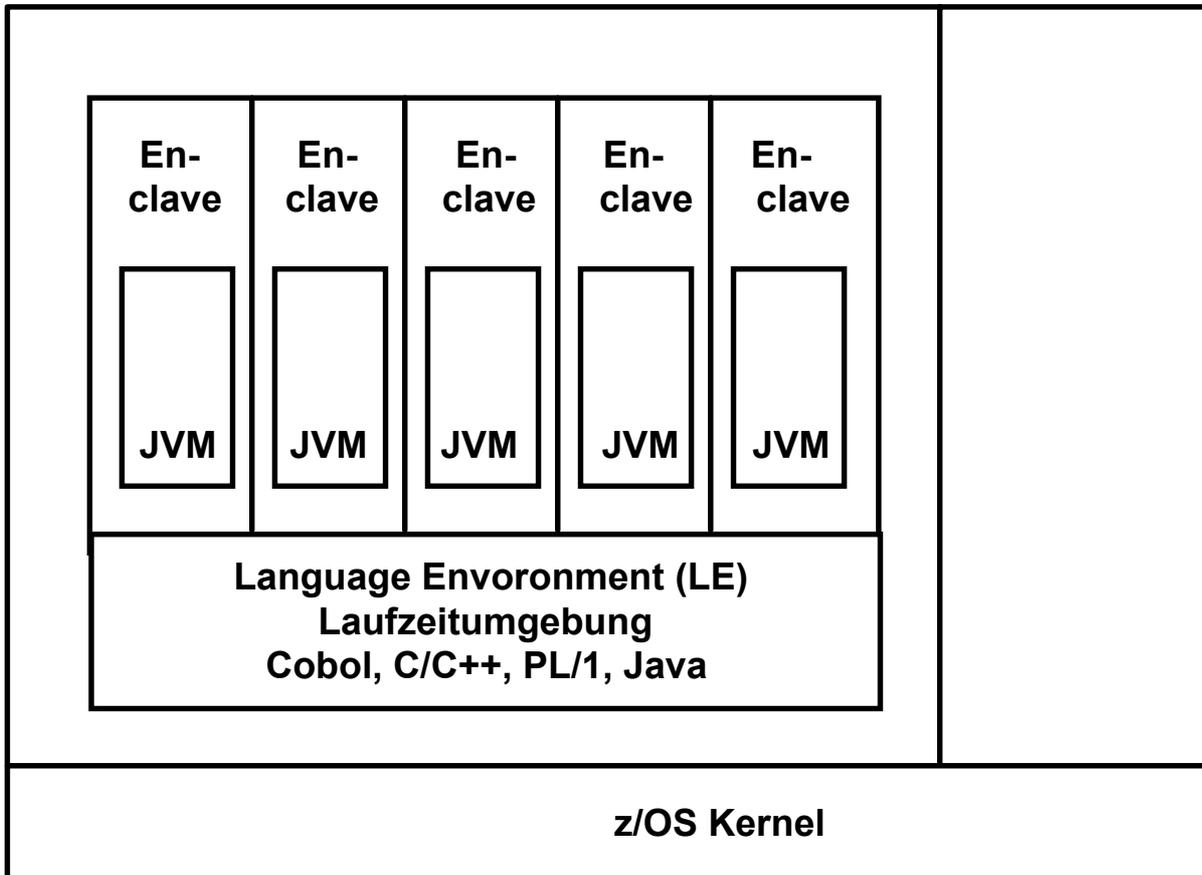
### Beispiele

- übrig gebliebene geöffnete Files,
- noch existierende „user threads“
- Referenzen in den Transient Heap durch lokale Variablen.

Wenn eine dieser Bedingungen auftritt, erzeugt der Aufruf `ResetJavaVM` den Rückgabewert `false`

Virtueller Adressenraum # 1

# 2



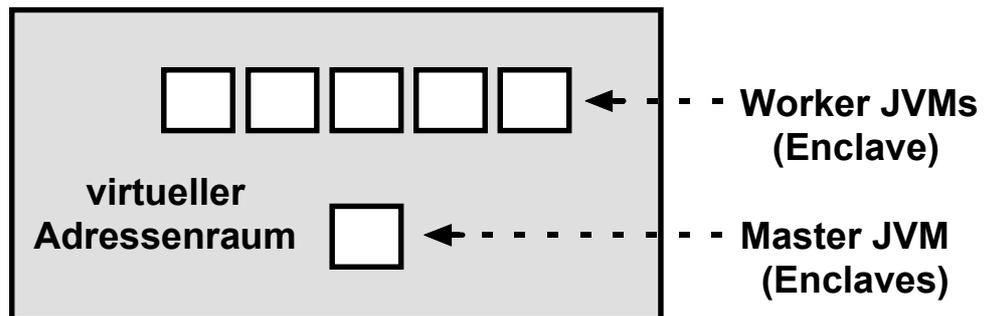
## **z/OS Enclaves**

**Das *Language Environment (LE)* war ursprünglich eine von verschiedenen Compilern (Cobol, PL/1, C/C++) gemeinsam genutzte Sammlung von Laufzeit-Bibliotheken.**

**Enclaves sind LE Run-Time Units, die eine multithreaded Verarbeitung ermöglichen.**

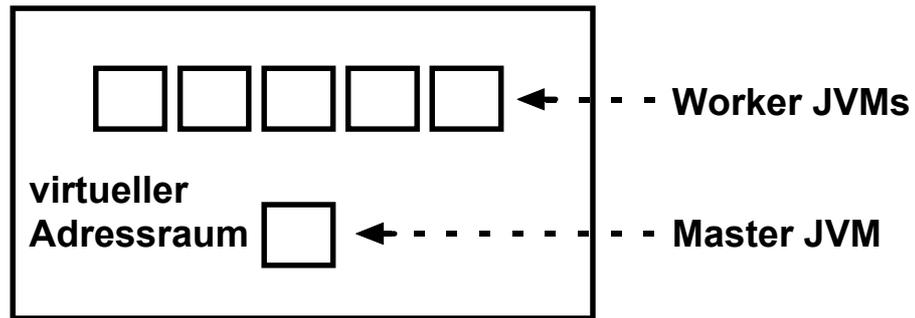
**Beim Einsatz von Java kann man jeweils eine JVM innerhalb einer Enclave unterbringen und damit mehrere JVMs (z.B. 60 .. 100) innerhalb eines einzelnen virtuellen Adressraums ausführen.**

## JVM Set



Die erste PRJVM übernimmt dabei die Rolle der *Master PRJVM* und kontrolliert den JVM Set:

- Sie stellt den System Heap zur Verfügung, welcher von allen *Worker PRJVMs* benutzt wird.
- Sie richtet die *Class Loader* Umgebung ein, die zum Laden der Systemklassen und *Shareable* Anwendungsklassen benötigt wird.

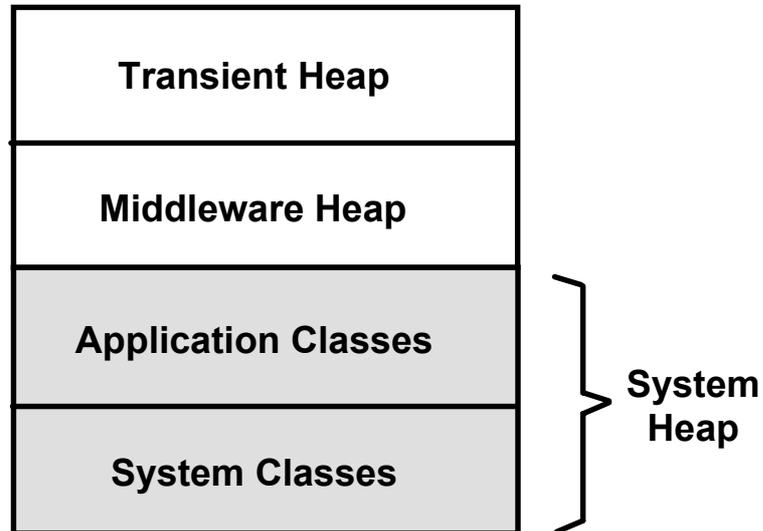


## LE Enclaves

Für z/OS und OS/390 Sprachen wie Cobol, PL/1, C/C++ und Fortran existiert eine als *LE* (Language Environment) bezeichnete Einrichtung. Dies ist zunächst eine Laufzeitumgebung von einheitlichen Bibliotheken, die von den genannten Sprachcompilern gemeinsam benutzt werden kann.

Ein Teil von LE sind die Run-Units oder Enclaves. Enclaves sind Thread-ähnliche Einheiten, die jeweils unter einem eigenen Prozessleitblock (Process Control Block, Task Control Block, TCB) gemeinsam in einem virtuellen Adressenraum laufen. Sie benutzen hierfür einen Speicherschutzschlüssel-(storage protection key), normalerweise Schlüssel Nr. 8. Der Hardware Speicherschutz arbeitet unabhängig und zusätzlich zu dem üblichen Speicherschutz über Segment- und Seitentafeln und ist eine Einrichtung der z/Series Hardware Architektur, die auf anderen Plattformen nicht verfügbar ist (siehe [ HERR ] , S. 14).

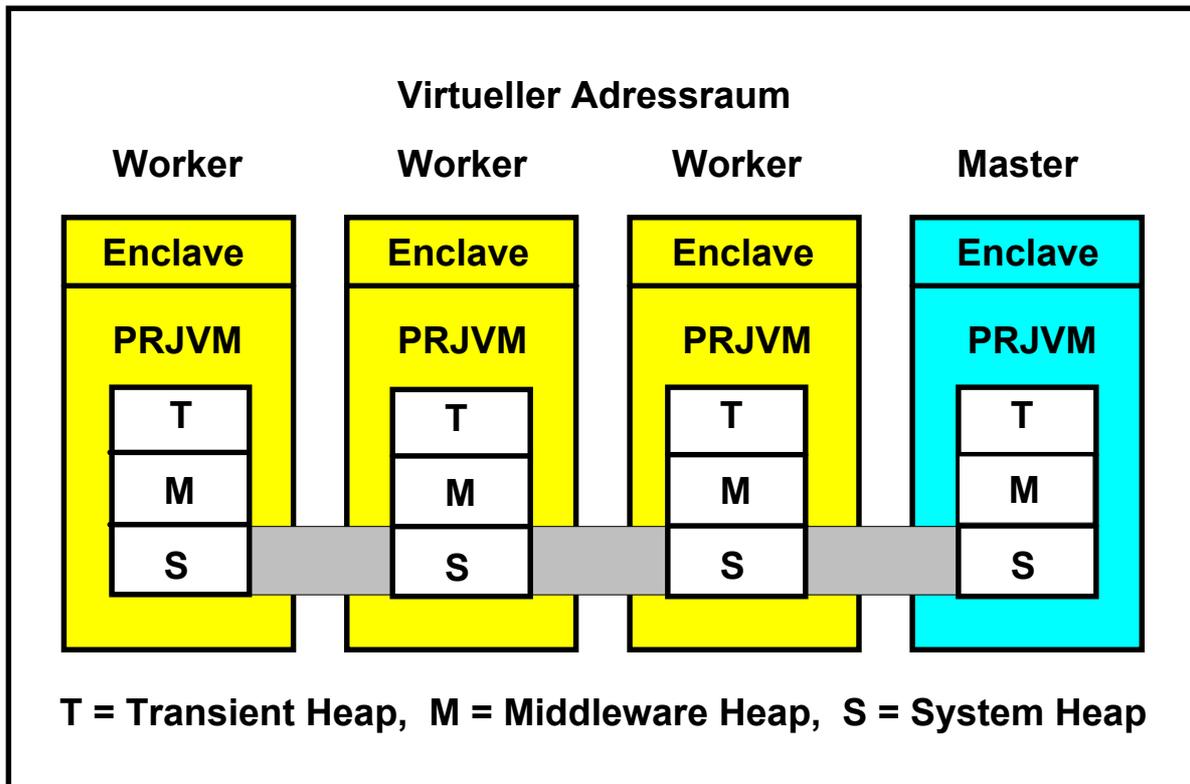
LE Enclaves können eingesetzt werden um unterschiedliche Anwendungen innerhalb des gleichen virtuellen Adressenraums voneinander zu isolieren. Sie werden u.a. in transaktionalen Subsystemen wie CICS, IMS und DB2 benutzt. Spezifisch ist es hiermit möglich, mehrere JVMs innerhalb eines Adressraums laufen zu lassen.



Der *Transient Heap* enthält Objekte, die von den Anwendungsklassen erzeugt werden.

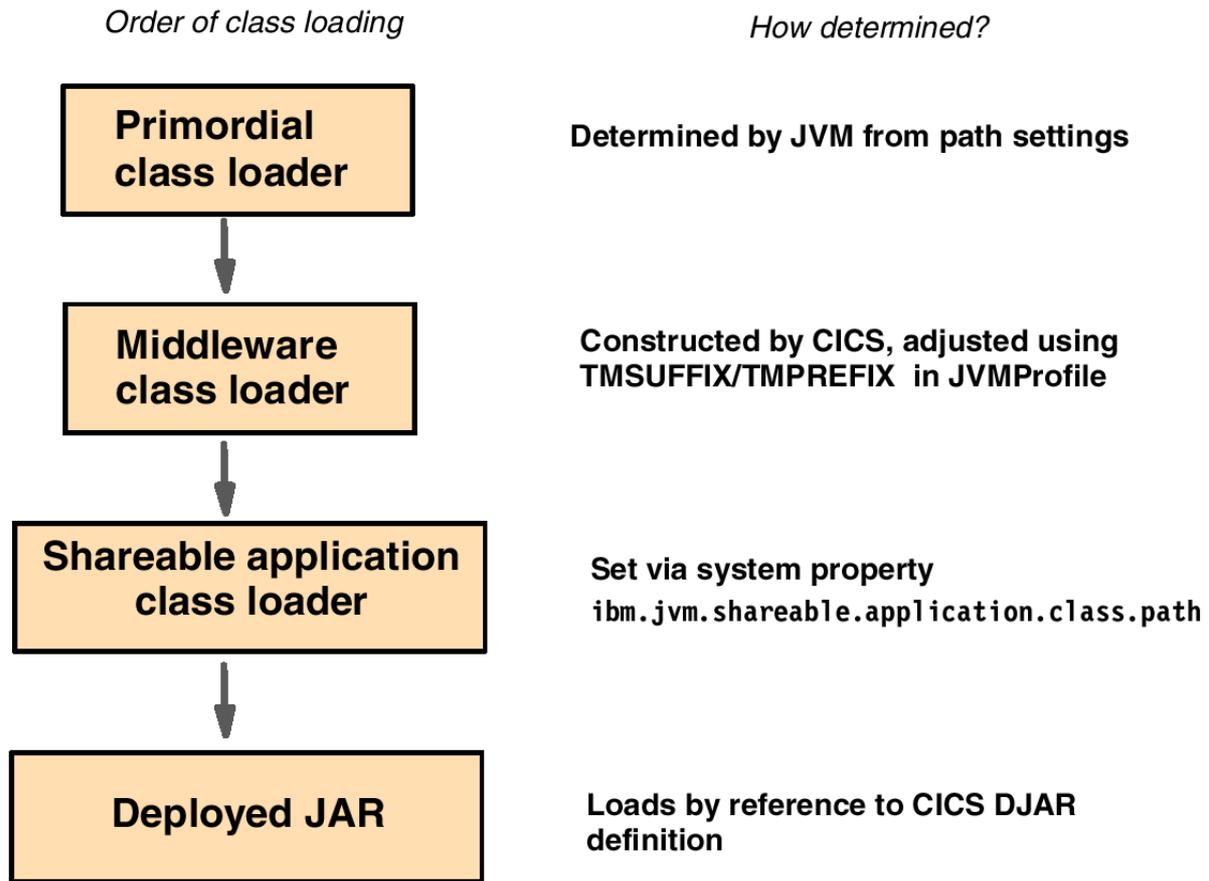
Der *Middleware Heap* enthält Objekte, die von den Middleware Klassen erzeugt werden. *Middleware enthält* Funktionen wie CICS Transaction Gateway Klassen, SQLJ/JDBC Klassen und JNDI Klassen.

Der *System Heap* besteht aus dem Main System Heap, der die System Klassen enthält, und dem Shareable Application Class Heap, der Klassen aus der eingesetzten JAR File enthält.



## Gemeinsame Nutzung des System Heaps und von JIT Klassen

**System- und Anwendungsklassen sind nur einmal vorhanden und können von allen PRJVMs gemeinsam genutzt werden**



## Order of class loading for the persistent reusable JVM

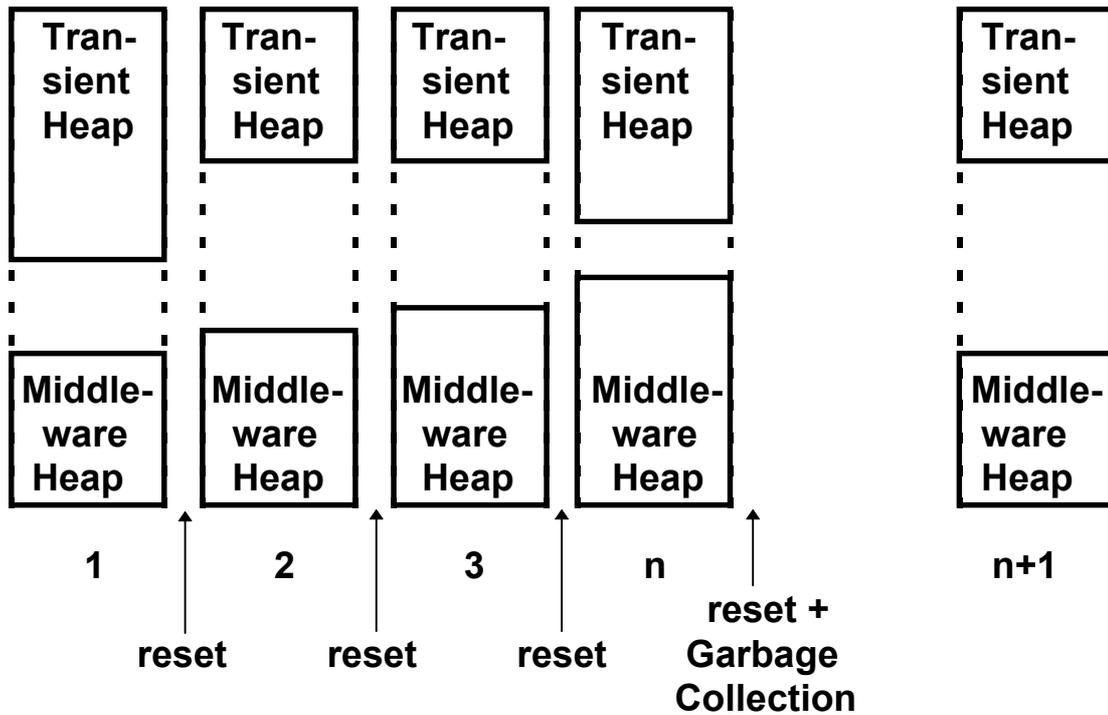
Serial reuse of a JVM is enabled by dividing the classes contained in the JVM into distinct parts as follows.

The *JVM* or *primordial code*, consisting of the system classes and standard Java extension classes, which provide the base services in the JVM.

\_ The *middleware*, which provides services that access resources. These include the JCICS interfaces classes, the CCF and CTG classes, the JDBC classes, and the JNDI classes. The trusted middleware class path is built automatically by CICS from the paths specified on the CICS\_DIRECTORY and JAVA\_HOME parameters. Additional directories or JAR files can be added by using the TMPREFIX and TMSUFFIX parameters. Generally speaking, only classes supplied by IBM, or your chosen middleware vendor, should be placed on the trusted middleware class path, using the TMSUFFIX parameter.

\_ The *shareable application* code, consisting of the enterprise beans themselves and designated shareable application code. The shareable application classpath is defined using the system property `ibm.jvm.shareable.application.class.path` and should be used for any utility classes. Classes defined in here take precedence over classes from the deployed JAR file.

The enterprise bean classes are also loaded into the application class system heap. They are located by reference to the DJAR definition defined in CICS.



**Garbage Collection erfolgt lediglich nach 100 Resets. Bedeutender Leistungsgewinn.**

JNI\_CreateJavaVM Call -Xresettable  
Create Master JVM  
Create Worker JVMs  
Start Transaktionen

while Verarbeitung laufender Transaktionen  
warten auf neue Transaktion

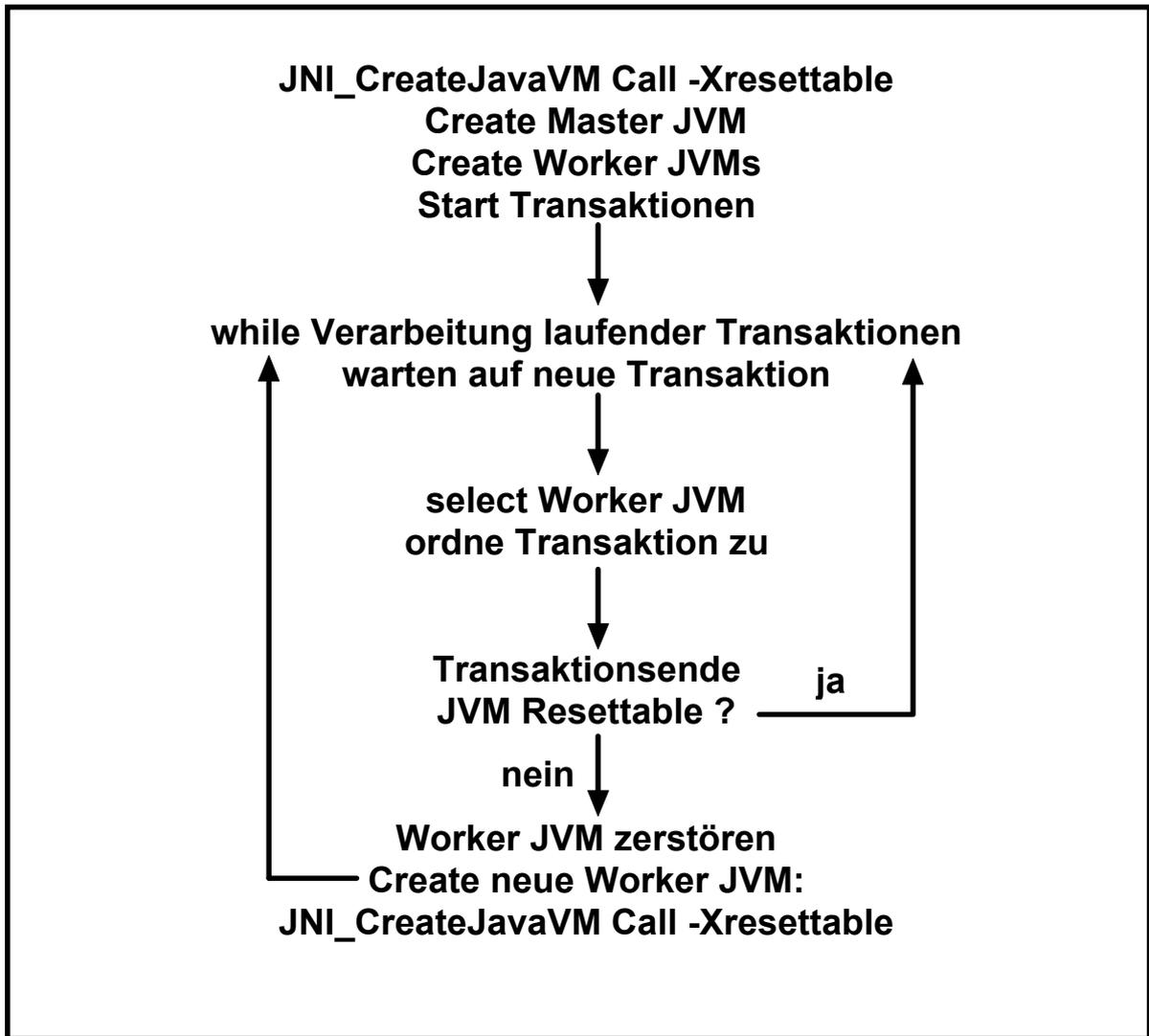
select Worker JVM  
ordne Transaktion zu

Transaktionsende  
JVM Resettable ?

ja

nein

Worker JVM zerstören  
Create neue Worker JVM:  
JNI\_CreateJavaVM Call -Xresettable



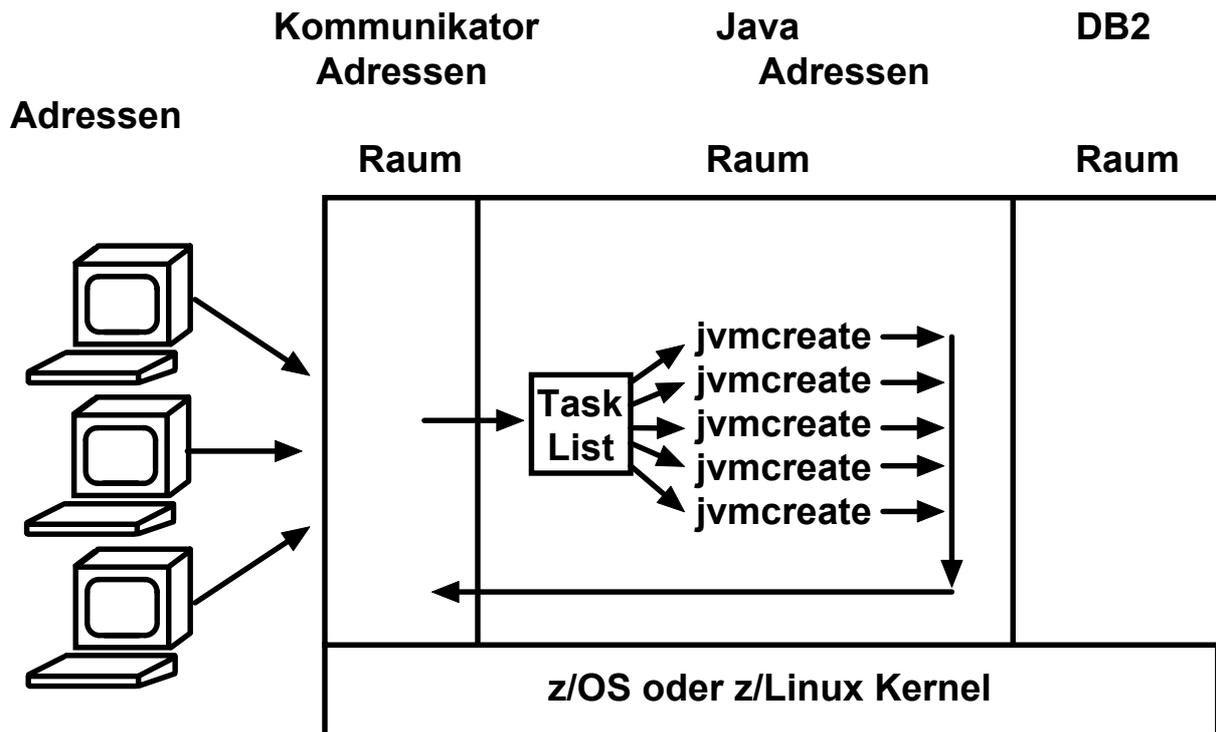
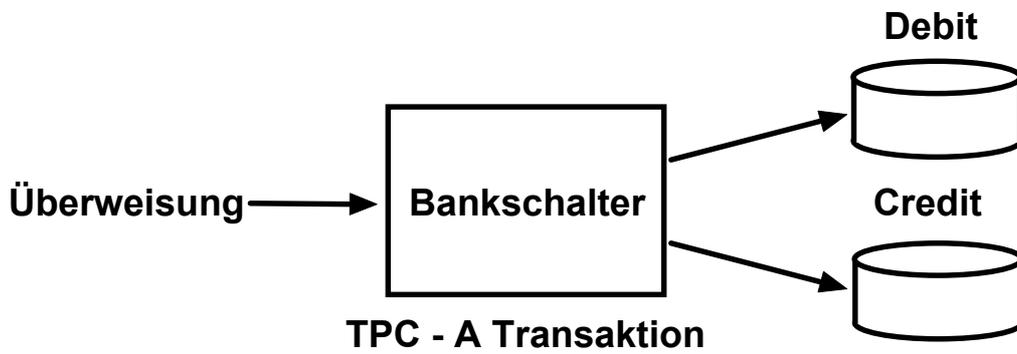
# **Architekturanalyse der Java Virtual Machine unter z/OS und Linux**

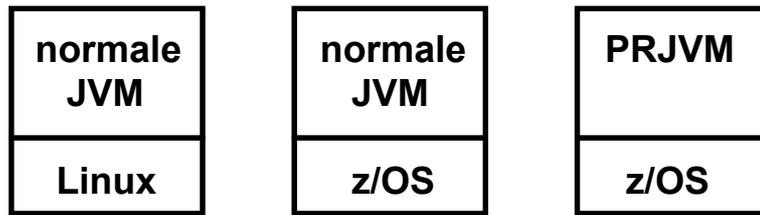
**Diplomarbeit  
der Fakultät für Informations- und  
Kognitionswissenschaften  
an der Universität Tübingen**

**vorgelegt von**

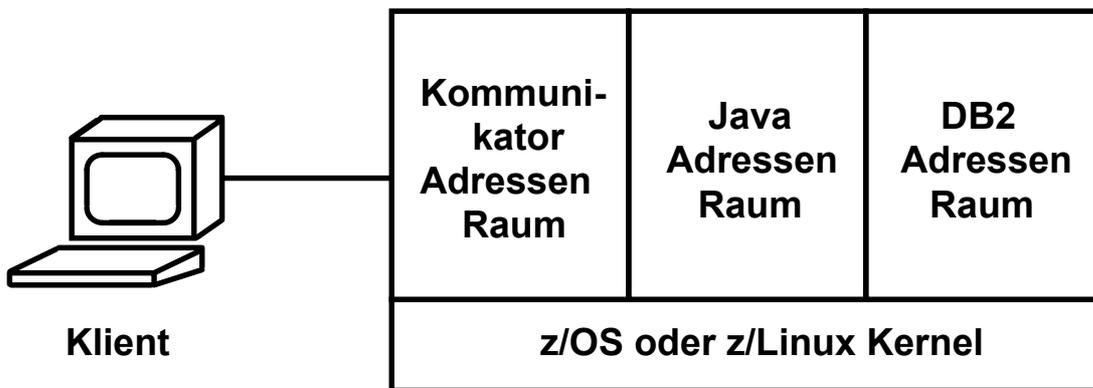
**Marc Beyerle  
Matrikelnummer: 1830910**

**Betreuer:  
Prof. Dr.-Ing. Wilhelm G. Spruth  
Dr. Bernhard Dierberger**

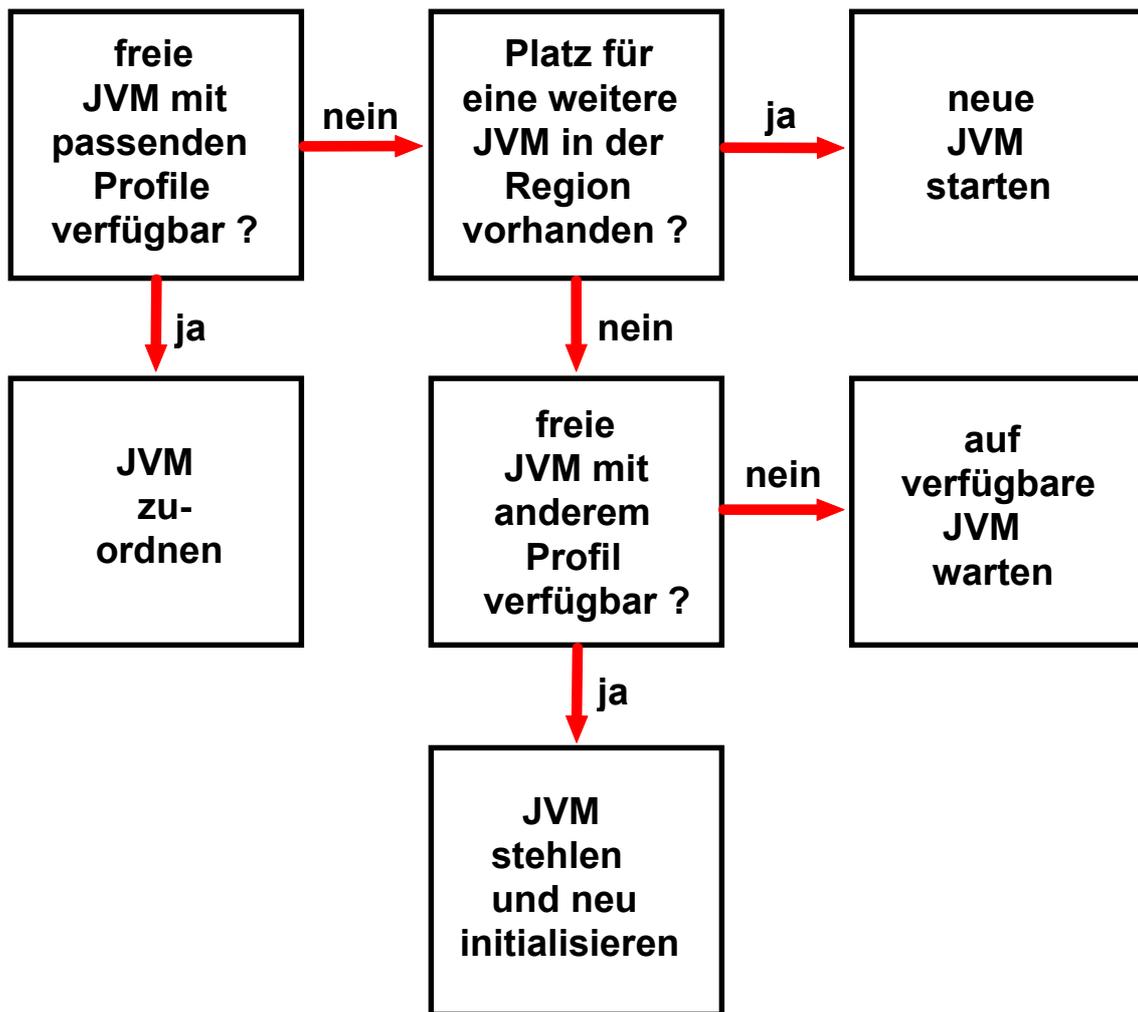




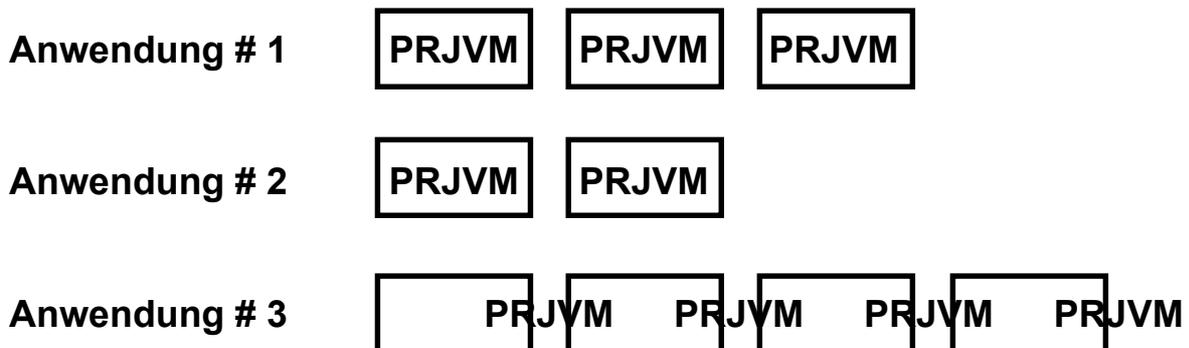
### Testumgebung



Normale JVM zLinux Tx/s	normale JVM z/OS Tx/s	Verhältnis zLinux zu z/OS	PRJVM z/OS Tx/s	Verhältnis PRJVM zu normale JVM
1,223	0,796	1,54	261,04	327,94



Unterschiedliche Transaktionstypen unterscheiden sich durch unterschiedliche Anwendungsklassen. Nach Möglichkeit eine freie Worker PRJVM mit den richtigen Anwendungsklassen auswählen. Wenn das nicht möglich ist, Anwendungsklassen in eine verfügbare Worker JVM nachladen.

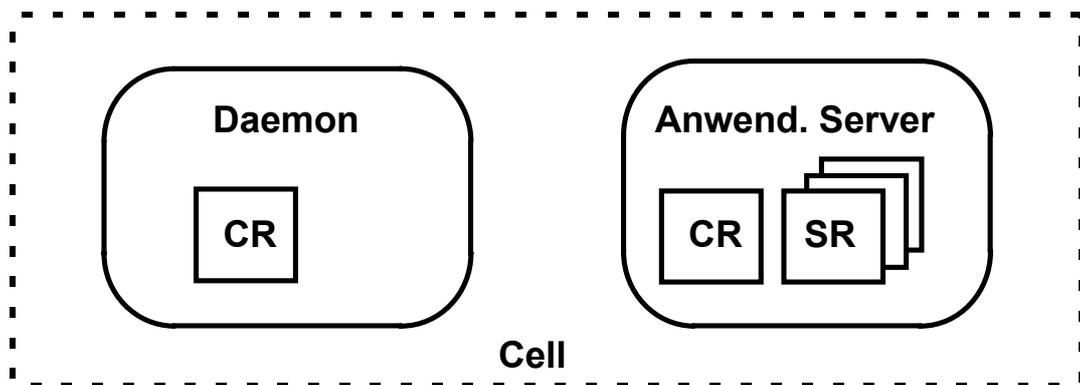


# WebSphere

WebSphere auf Unix, Linux und Windows Plattformen ist als ein Einzel-Prozess Server implementiert. Alle Anwendungen und der Systemcode laufen im gleichen virtuellen Adressenraum.

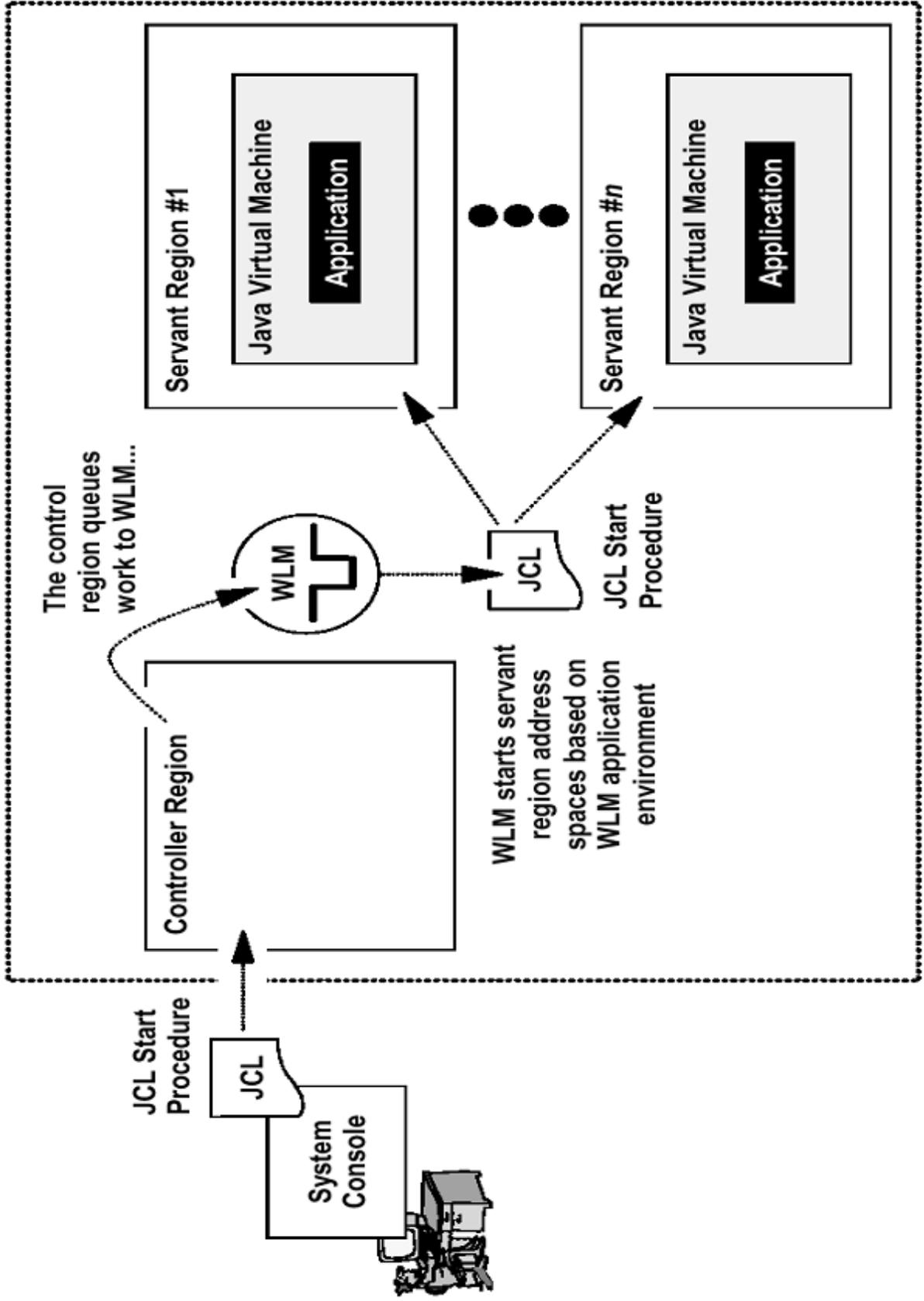
z/OS WebSphere entspricht einer multi-Prozess Architektur. Anwendungen sind gegeneinander isoliert

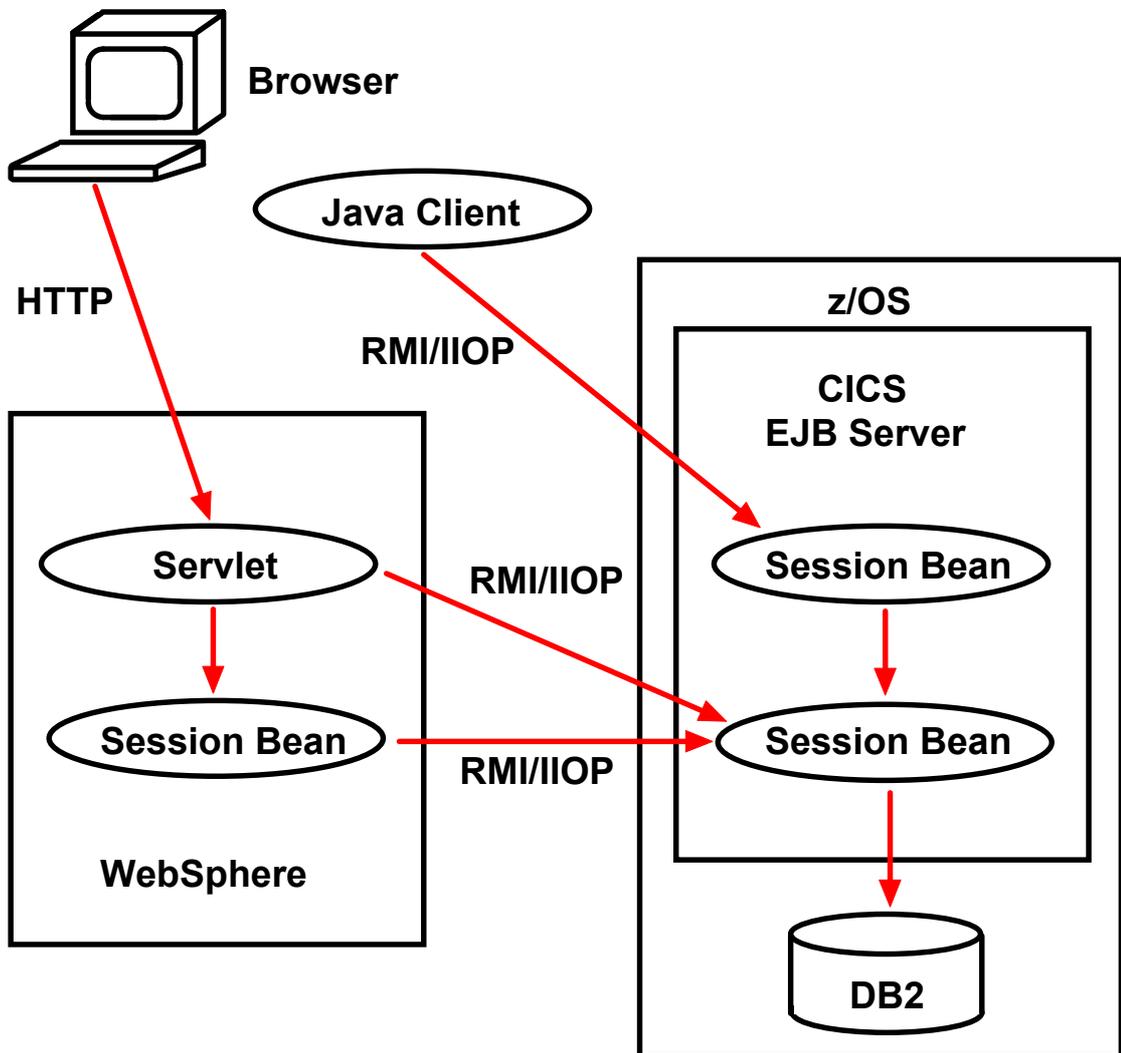
## z/OS WebSphere Struktur



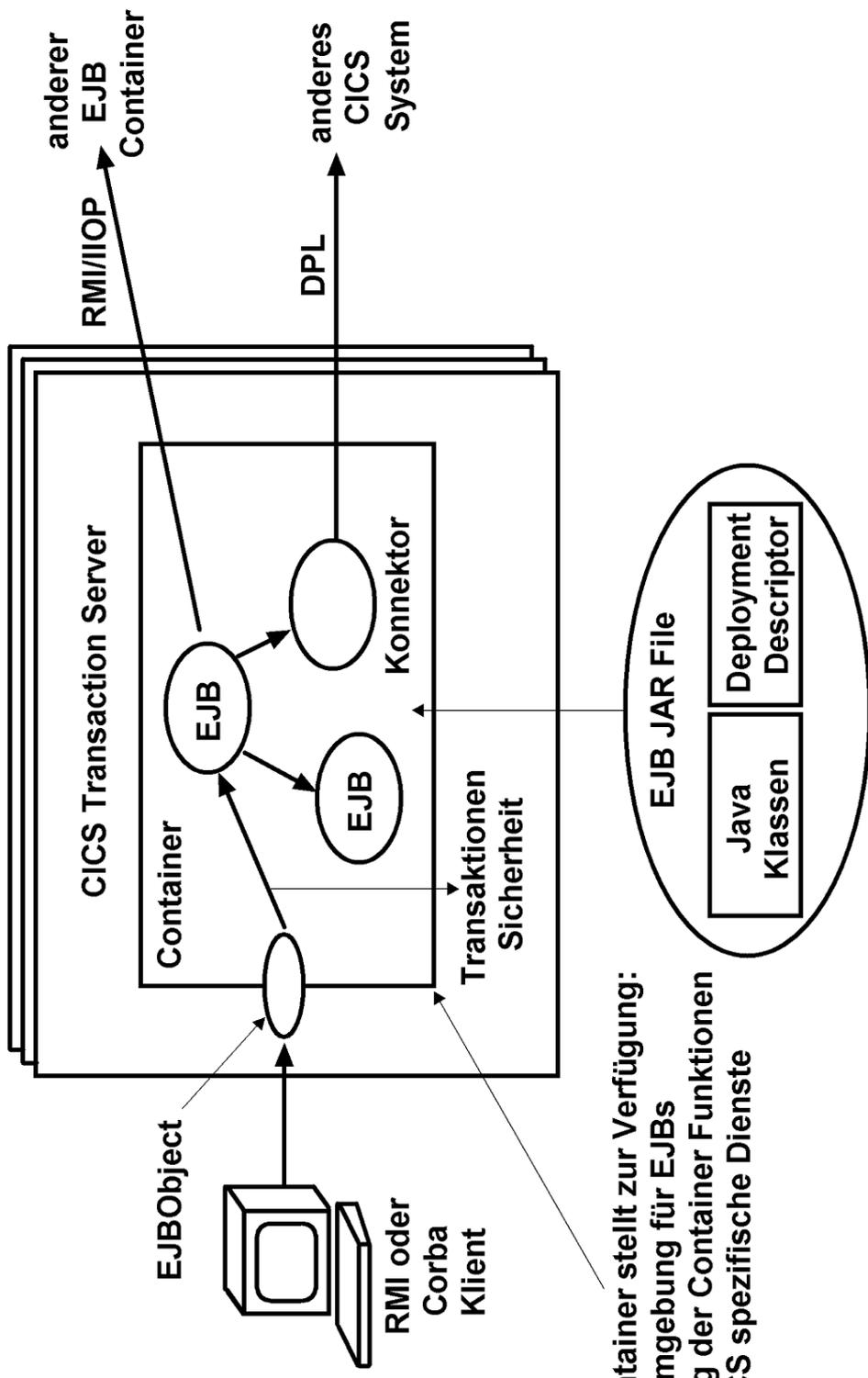
**Daemon empfängt IORs**

**CR = Controller, SR = Servant**



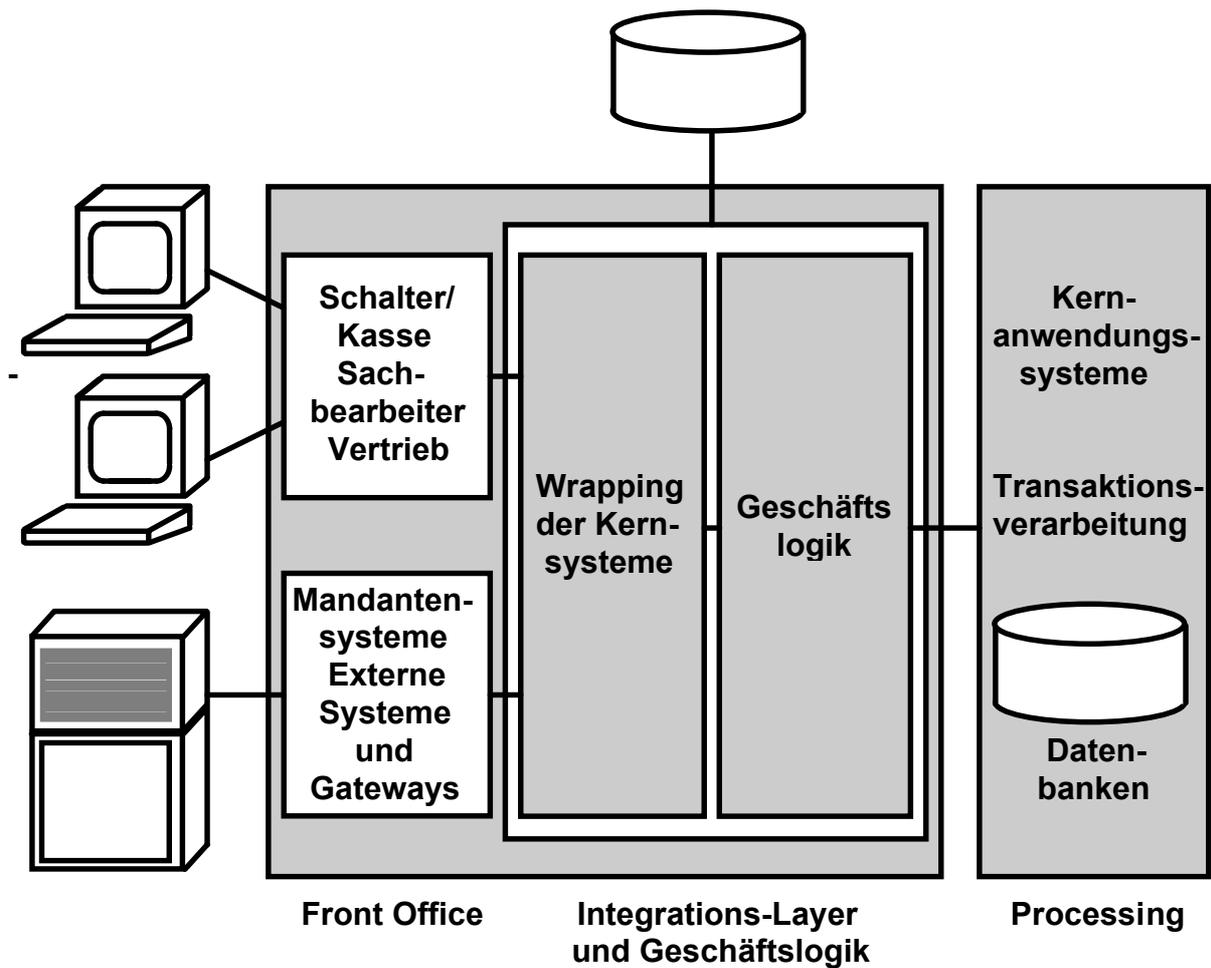


## Benutzung von Enterprise Java Beans in einer CICS Anwendung



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

## CICS Nutzung der PRJVM



**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.**

## **Transaktionsmaschine - 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 Subworkflowsteuerung  
Diplomarbeit, Fakultät für Informations- und Kognitionswissenschaften,  
Universität Tübingen  
<http://www-ti.informatik.uni-tuebingen.de/~spruth/DiplArb/index.html>**

**Martin Schlodder: Datenzugriffsdienst für eine Transaktionsmaschine  
Diplomarbeit, Fakultät für Informations- und Kognitionswissenschaften,  
Universität Tübingen  
<http://www-ti.informatik.uni-tuebingen.de/~spruth/DiplArb/index.html>**

# **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**

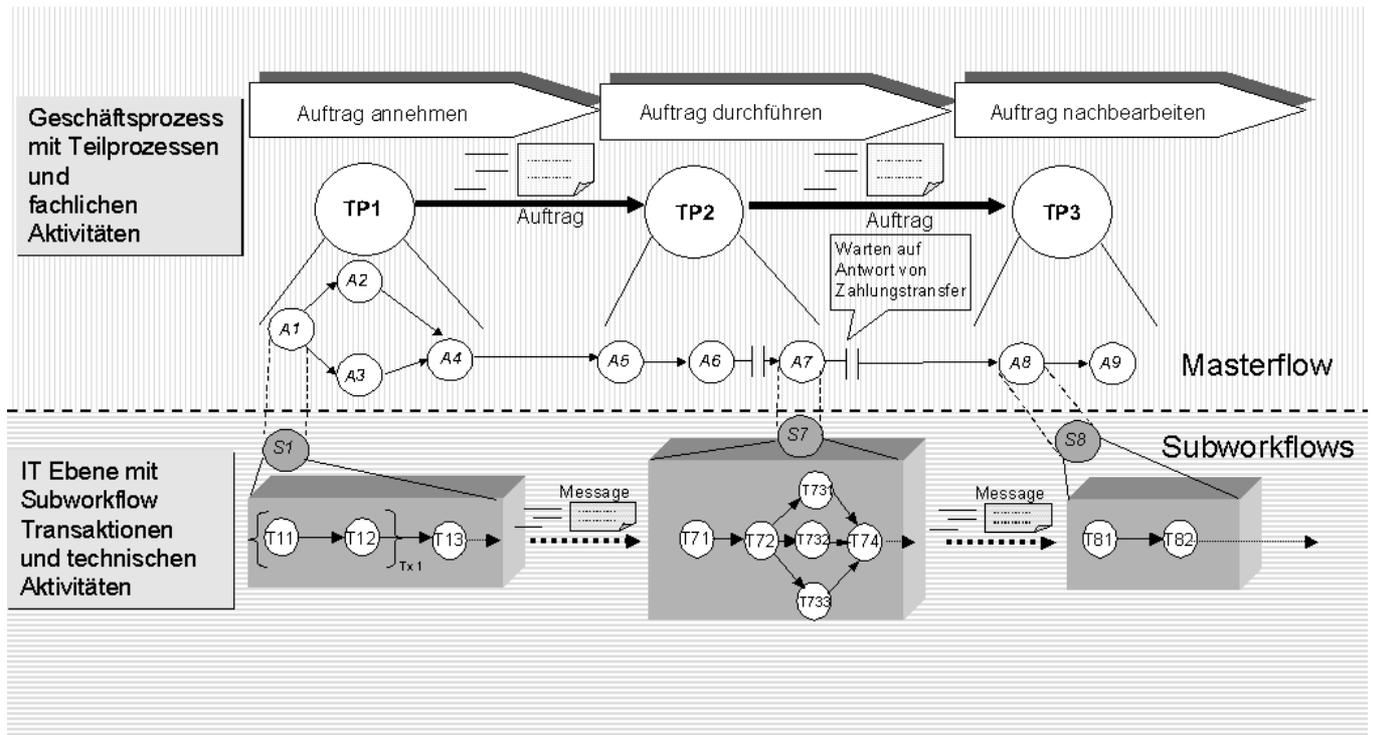
**Hinzufügen, Verändern und Ersetzen von Banken oder Versicherungsprodukten im laufenden Betrieb**

## **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**

## **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?**

**Was ist die Aufgabe des SAP (Support Assist Processor)**

**Kann eine E/A Operation während ihres Ablaufs über unterschiedliche Pfade (Kanäle) mit der CPU verbunden sein ) ?**

**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 der Goal 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?**

**Hat SNA einen ähnlichen Funktionsumfang wie TCP/IP ?**

**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 ?**

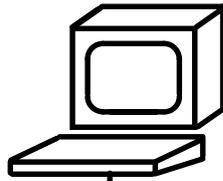
**Wodurch unterscheidet sich Multiprogramming mit getrennten Prozessen von der Nutzung von Threads ? Was sind Vor- und Nachteile ?**

**Gibt es Schwierigkeiten mit der Nutzung von Java Threads bei der Aufrechterhaltung der ACID Eigenschaften von Transaktionen ?**

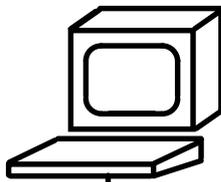
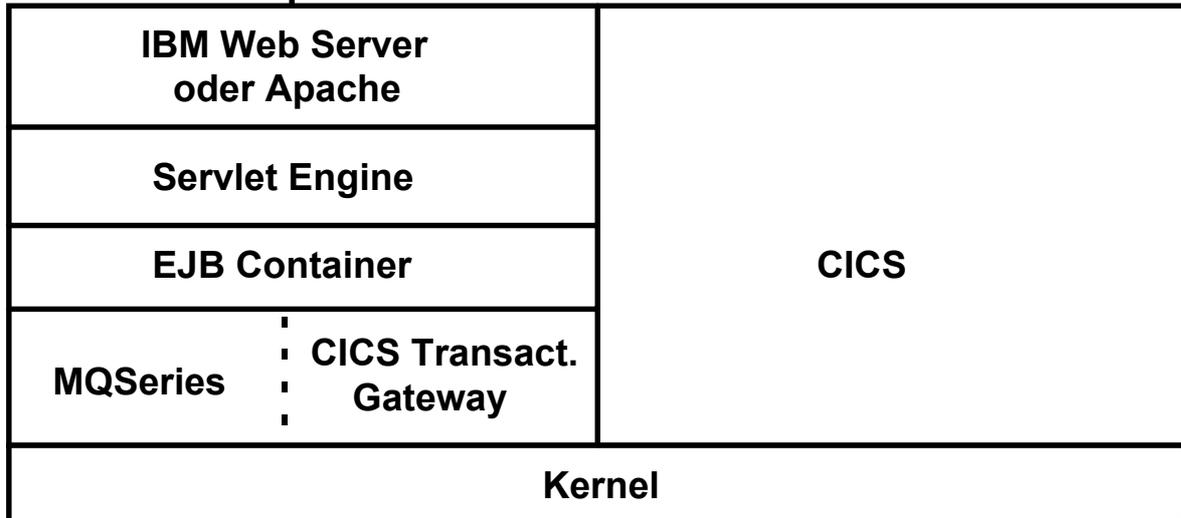
**Was ist die Funktion der Persistent Reusable Virtual Machine ? Welches Problem wird damit gelöst ?**

**Was sind Language Environment (LE) und Enclaves ?**

**Was ist ein Business Prozess?**

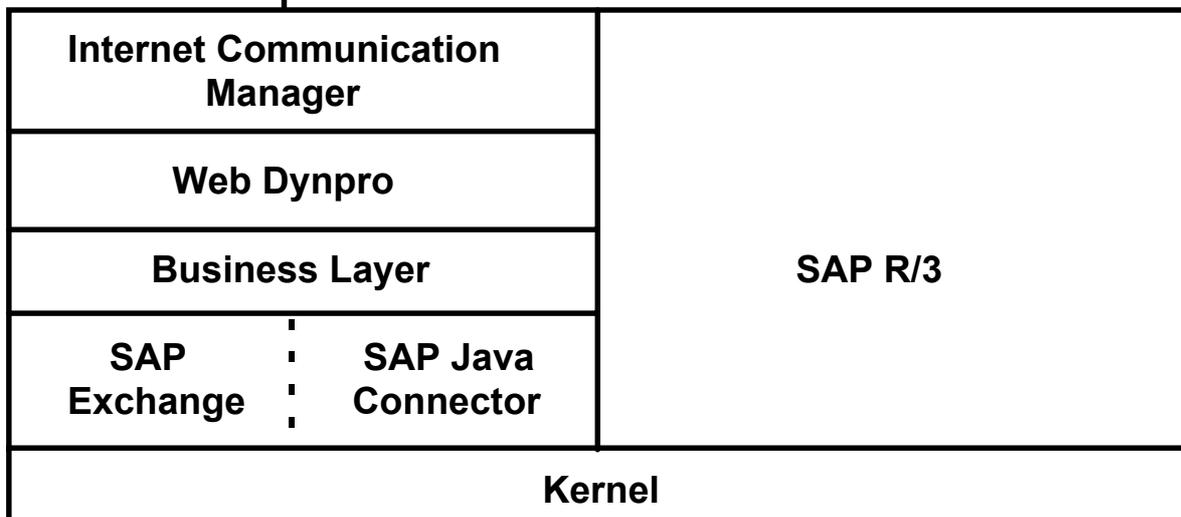


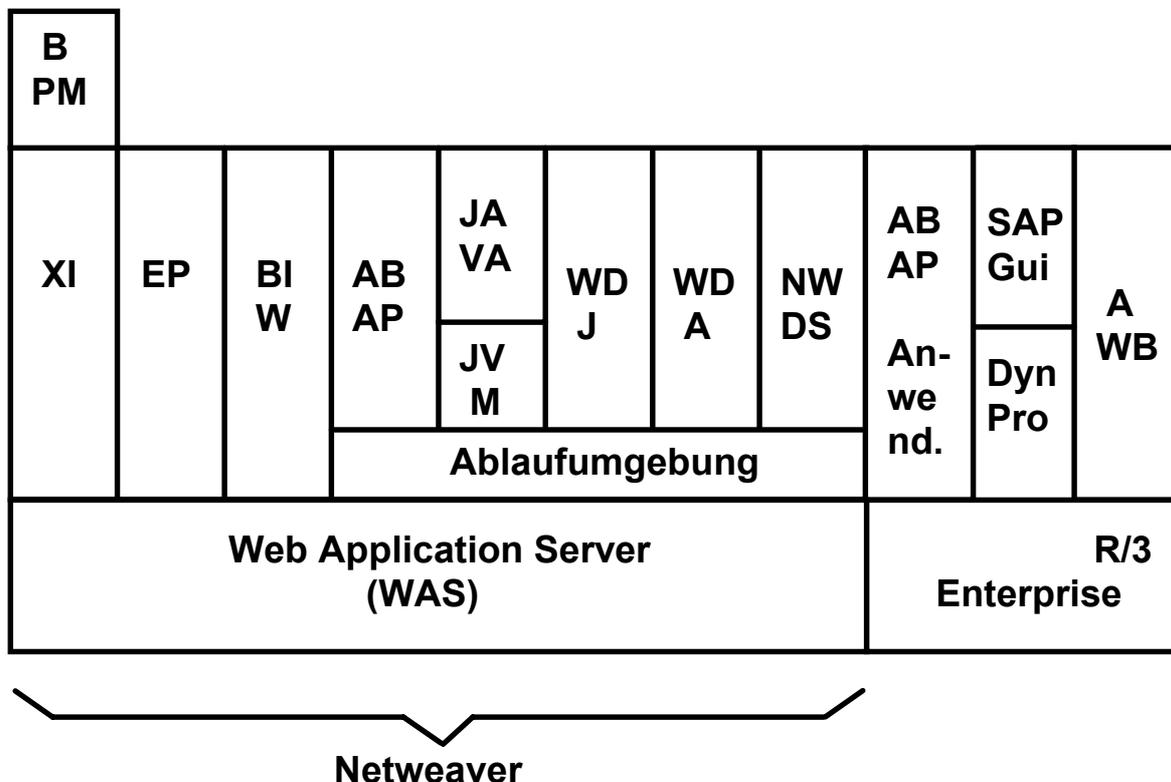
IBM WebSphere



SAP Web Applic. Server

**Web Application Server  
als Transaktionsmonitor  
Front End**





## Struktur eines modernen SAP R/3 Systems

<b>BPM</b>	<b>Business Process Management (benutzt BPEL)</b>
<b>XI</b>	<b>Exchange Infrastruktur</b>
<b>EP</b>	<b>Enterprise Portal</b>
<b>BIW</b>	<b>Business Information Warehouse</b>
<b>JVM</b>	<b>Java Virtual Machine</b>
<b>WDJ</b>	<b>Web Dynpro for Java</b>
<b>WDA</b>	<b>Web Dynpro for ABAP</b>
<b>NWDS</b>	<b>Netweaver Development Studio</b>
<b>AWB</b>	<b>ABAP Workbench</b>

## **SAP Web Application Server Architecture**

**SAP Web Application Server is the e-business infrastructure for mySAP.com solutions, SAP R/3 Enterprise, SAP xApps, and any J2EE-based applications.**

### **Internet Communication Manager**

**The ICM dispatches HTTP requests to the presentation layer and sends Web services requests to the integration engine.**

**Mit dem Internet Communication Manager (kurz ICM) verfügt das System über einen eigenen Webserver, der die Internet-Protokolle http, https, smtp, SOAP und XML unterstützt.**

### **Presentation Layer**

**Web Dynpro is a runtime environment plus a set of development tools based on the Model View Controller (MVC) paradigm.**

**The Web Dynpro technology in the Web Application Server provides a professional programming model with a development and runtime environment for building Web-based application interfaces. The Web Dynpro technology is based on Java Server Pages (JSP) and a tag library containing, among other things, ready-made user interface elements (such as push buttons and input fields).**

**It enables virtually device-independent development for browsers, PDAs, and WAP-enabled cell phones**

**Das Konzept „Web-Dynpro“ ersetzt das mittlerweile 20 Jahre alte Dynpro-Konzept des SAP-Systems. Die Präsentationsschicht der Applikationen wird separiert und basiert auf Java/J2EE-JSP Technologie. Somit werden Anwendungen die auf Java/J2EE und auf ABAP basieren, mit einer gemeinsamen Präsentationsschicht versorgt und die Business-Logik wird getrennt.**

**The description of the application is kept in a metadata repository and is used to generate the runtime code for one of the following runtime environments: Java, ABAP, or .NET.**

**Für interaktive Webanwendungen steht die Web-Dynpro-Runtime zur Verfügung, die für die Erstellung der browserbasierten Oberfläche zuständig ist. verwendet Programmiermodell „Java Server Pages“**

## **Business Layer**

**The business layer processes the requests from the ICM and dynamically generates responses.**

**Die Business-Logik wird entweder mit den bereits vorhandenen SAP Business Objects und den Methoden dieser Objekte (BAPIs) oder mit Enterprise Java-Beans (EJB) realisiert**

**Enterprise Java Beans which link the presentation layer and business-critical data in the J2EE programming model are responsible for the application logic to process data and generate results.**

## **SAP JAVA Connector**

**Die Kommunikation zwischen den Java- und den ABAP-Komponenten wird mit dem SAP Java Connector realisiert. Der Connector stellt dem Java-Entwickler eine Klassenbibliothek zur Verfügung, um mittels des Protokolls Fast-RFC ABAP-Komponenten einzubinden. Auch die Kommunikation ABAP->Java ist damit möglich.**

**Die Business-Komponenten greifen gemeinsam auf die Datenbank des Servers zu. Mittels der J2EE-Datenbankschnittstelle JDBC können weiter Datenbanken problemlos eingebunden werden. So sind auch Transaktionen mit verteilten Datenbanken möglich.**

## **Integration Layer**

**Zur Integration von Fremdsystemen wird der Web Application Server um die Komponente „Exchange-Infrastruktur“ erweitert. SAP Web Application Server can be connected to the SAP Exchange Infrastructure using its Integration Engine. Dem Kunden steht damit ein vollständiges Werkzeug zur Integration von unternehmensinternen und unternehmensübergreifenden Applikationen („EAI-Tool“) zur Verfügung. Die Kommunikation basiert auf wie XML-basierten Protokollen wie SOAP oder xCBL.**

## **Persistence**

**SAP Web Application Server's persistence capabilities include optimized data access through OpenSQL for ABAP and OpenSQL for Java**

## **IDE's**

**SAP has offered the robust ABAP Workbench as an integrated development environment (IDE) that is proven. The same is now extended to the IDE for Java**

## **Connectivity**

**The WAS supports J2EE Connector Architecture as well as Microsoft .NET connectivity. It is open to the CORBA, COM+, FTP and SMTP protocols to connect to non-SAP systems. WAS supports XML, SOAP, WSDL, UDDI and WSCI. The BAPI Interface of SAP, for example, is available as Web Services through WSDL.**