UNIVERSITÄT LEIPZIG

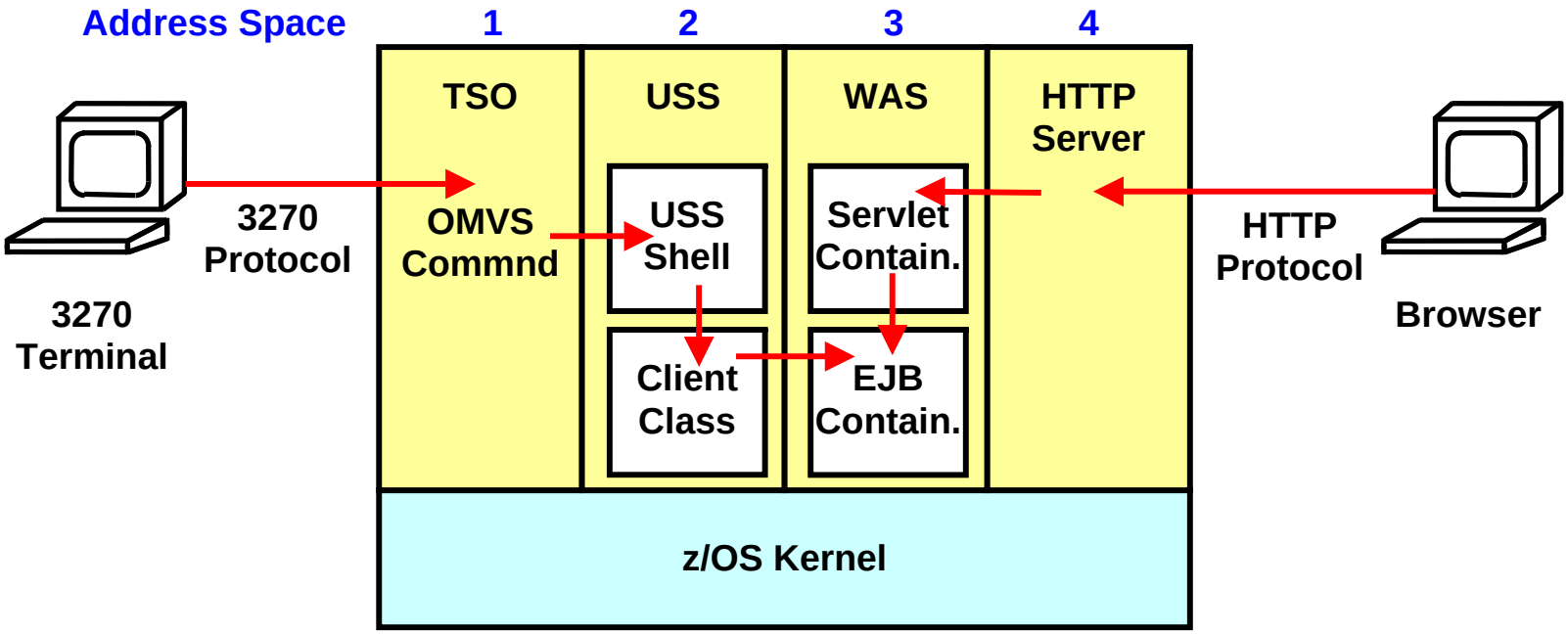
Mainframe Internet Integration

Prof. Dr. Martin Bogdan
Prof. Dr.-Ing. Wilhelm G. Spruth

SS2013

WebSphere Application Server Teil 4

Leistungsverhalten

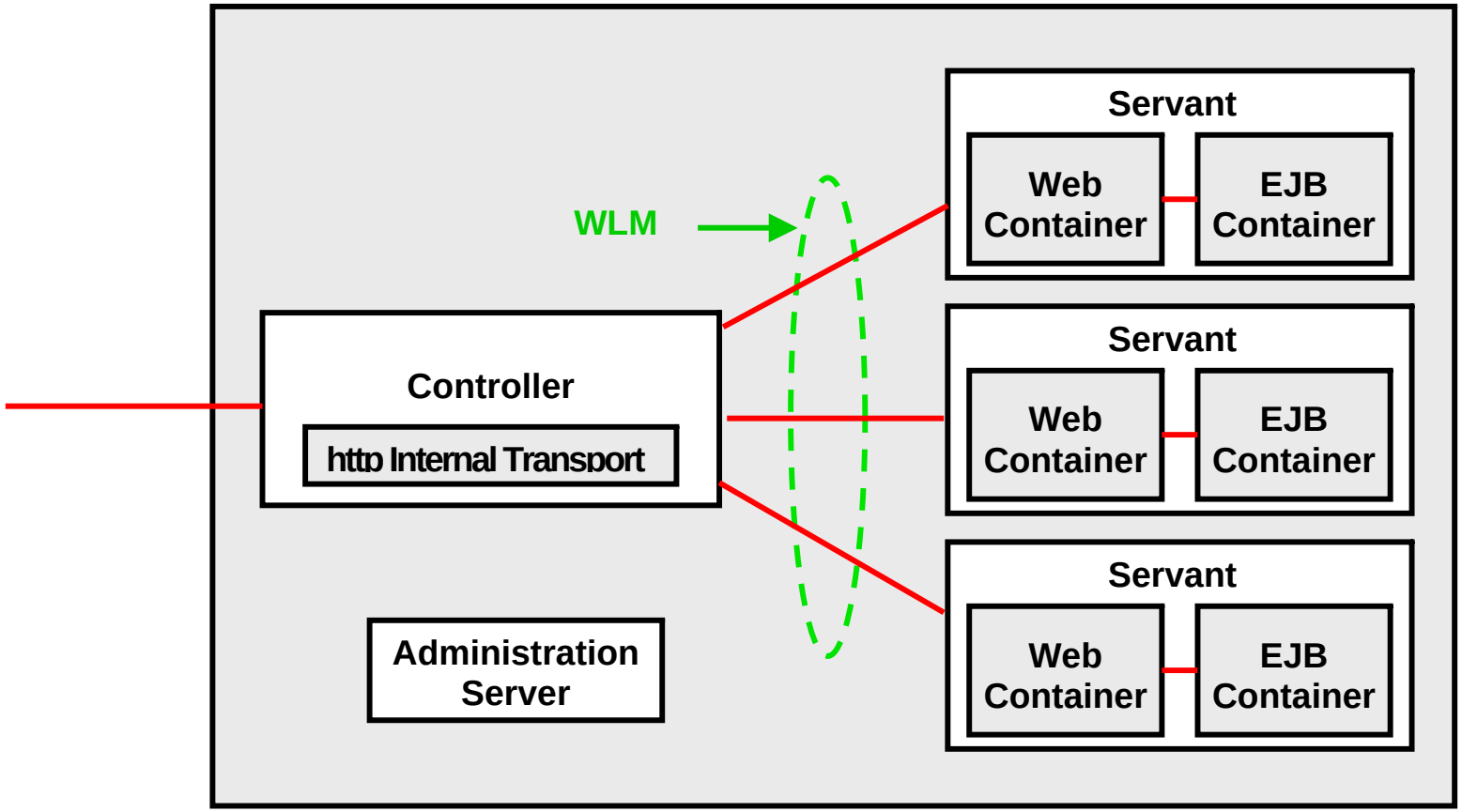


Arten von EJB Klienten

Der Zugriff auf eine EJB (JEE application) erfordert Code, der einen Klienten für diese EJB darstellt. Eine Möglichkeit besteht darin, mittels eines 3270 Terminals und TSO über das OMVS Command eine USS Session aufzurufen. In dieser Session wird eine Java Client Class mittels eines Unix System Services Shell Kommandos aufgerufen. Der Klient ruft über eine RMI/IIOP Verbindung eine EJB auf, die unter dem WebSphere Applicarion Server läuft.

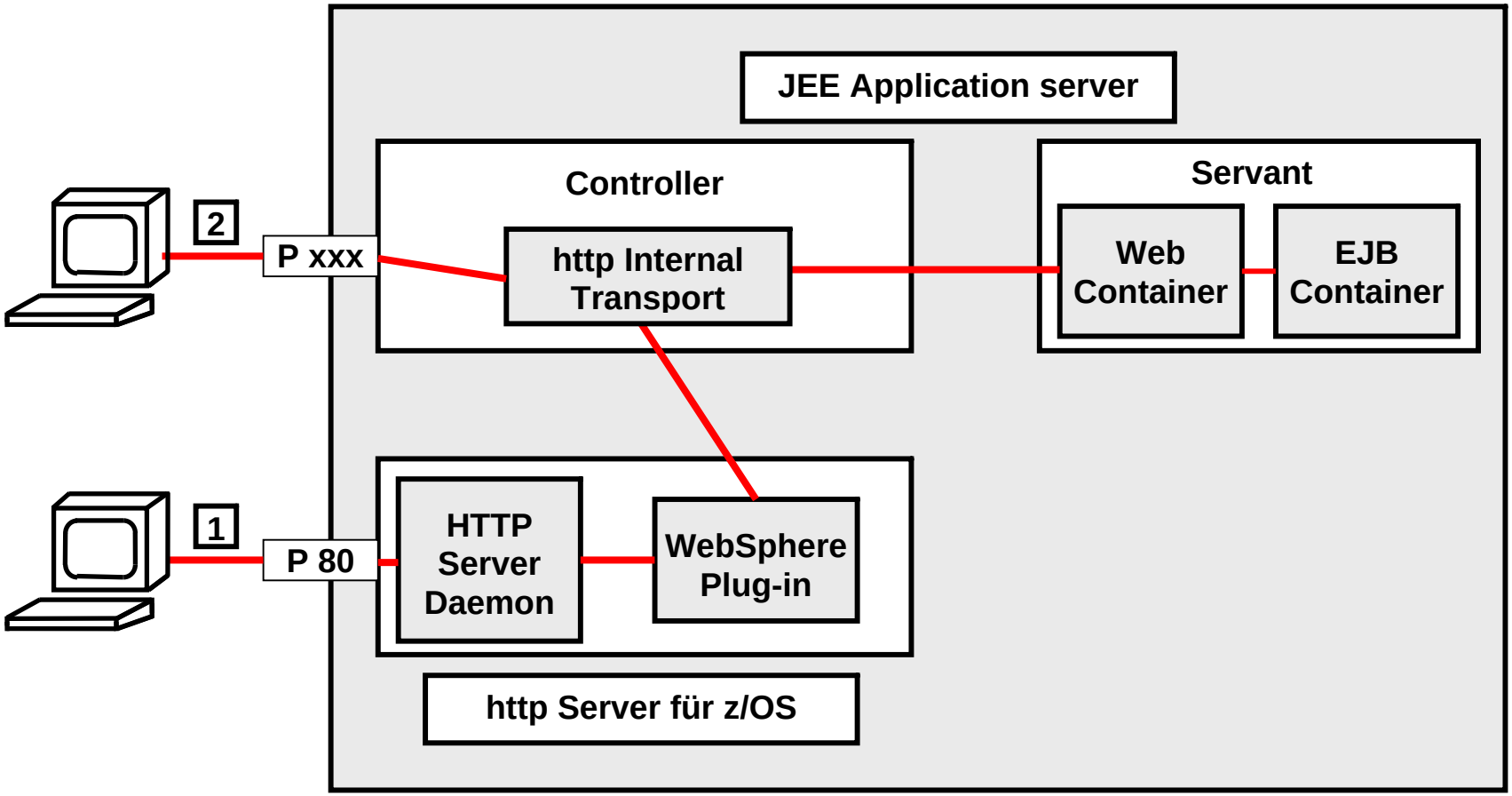
Der Vorteil dieses Vorgehens besteht darin, dass der Klient relativ leicht zu installieren ist, ohne dass man sich über Netzwerkprobleme Gedanken machen muss. Für das Austesten einer neuen Anwendung mag das günstig sein.

Viel üblicher ist es jedoch, mittels eines Browsers auf den z/OS http Server und ein Servlet, und über diesen eine EJB aufzurufen.



Dargestellt ist nochmals die z/OS Version des WebSphere Web Application Servers. Der Controller übernimmt die Aufgabe des Interactive Network Dispatchers für eine Gruppe von Servants. Letztere stellen die eigentlichen Web Application Server dar. Controller und Servants laufen in getrennten Adressenräumen und besitzen jeder eine eigene JVM.

Der Google Interactive Network Dispatcher hat sehr viel weniger Funktionalität als der WAS Controller Interactive Network Dispatcher.



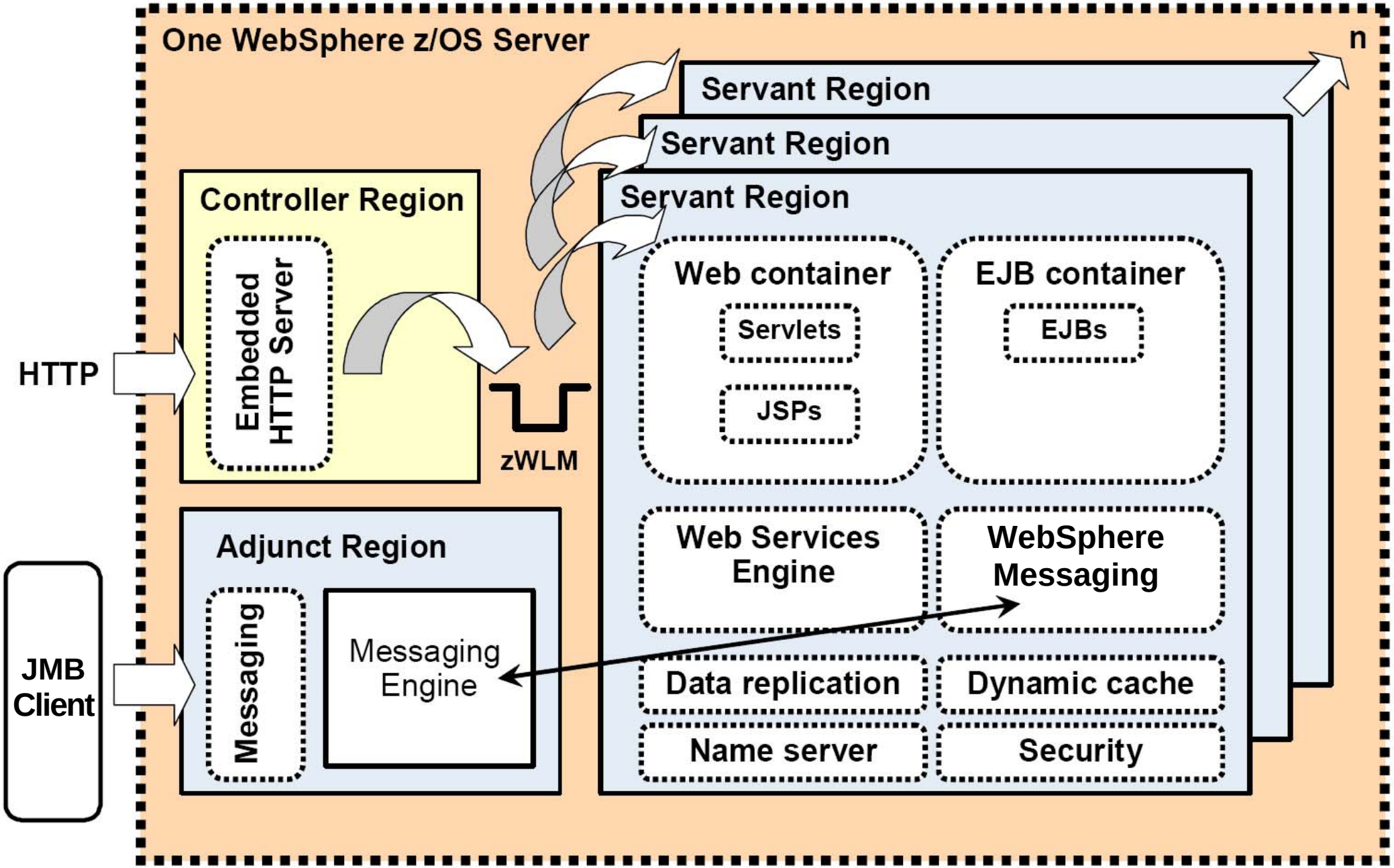
Angenommen, ein Benutzer möchte mittels seines Browsers eine WebSphere Anwendung (spezifisch ein Servlet) aufrufen, das unter z/OS läuft. Hierfür sind zwei Software Komponenten verfügbar:

1. **Die erste Komponente kann z.B. der z/OS http Server sein, der Nachrichten auf Port 80 entgegennimmt. Er benutzt das WebSphere Plug-in, um die http Nachricht an den WAS Controller weiterzureichen.**
2. **In einer z/OS Controller-Servant Konfiguration enthält der Controller aber eine alternative Komponente, den „http Internal Transport“. Dieser ist in der Lage, auf einem beliebigen Port Nachrichten direkt entgegenzunehmen, und unmittelbar an ein spezifisches Servlet in einen Servant weiterzuleiten.**

HTTP Internal Transport

WebSphere für z/OS unterstützt zwei verschiedene Arten von Komponenten für den Empfang von Nachrichten. Die erste (1) ist der IBM HTTP-Server, der auf Apache-Basis arbeitet, den Port 80 als Standard verwendet und http Nachrichten empfängt und sendet. Der HTTP-Server wartet auf eingehende HTTP-Anforderungen von Remote-Clients im Netzwerk. Sie aktivieren die WebSphere HTTP Plug-In-Komponente in dem HTTP-Server-Adressraum. Das Plug-in kann konfiguriert werden, HTTP-Requests vom HTTP-Server zu filtern, und Nachrichten an einen von einem von mehren möglichen JEE-Anwendungsservern weiterzureichen

Die zweite Art von Protokoll Catcher (2) in WebSphere für z/OS,wird als HTTP Internal Transport bezeichnet. Diese Komponente läuft in dem "Controller" (CR) Adressraum eines WebSphere für z/OS Application Servers. Die Internal Transport-Komponente hört ausgewählten Ports auf dem TCP / IP-Stack auf eintreffende Nachrichten ab. Im Prinzip können beliebige Protokolle eingesetzt werden. Wenn eine Nachricht empfangen wird, prüft der Internal Transport, dass die Nachricht in diesem Server ausführbar ist. Vorausgesetzt die Nachricht ist gültig, leitet der Internal Transport die Anforderung zur Ausführung an den Web-Container weiter. Der Web Container und der EJB-Container laufen in "Servant" (SR) Adressräumen.


One WebSphere z/OS Server
n
Servant Region
Servant Region
Servant Region
Controller Region
Embedded HTTP Server
HTTP
zWLM
Web container
Servlets
JSPs
EJB container
EJBs
Web Services Engine
WebSphere Messaging
Data replication
Dynamic cache
Name server
Security
Adjunct Region
Messaging
Messaging Engine
JMB Client

WebSphere Application Server Overview

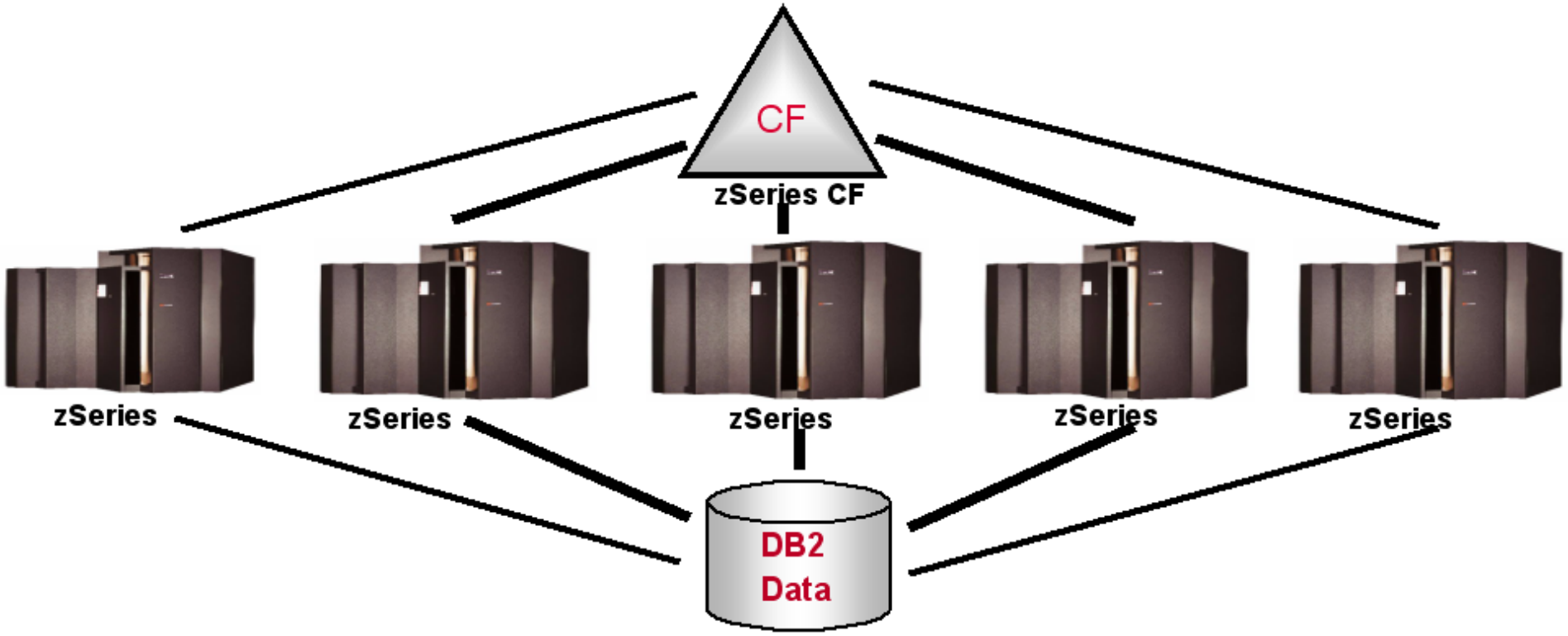
Die obige Abbildung zeigt Details eines z/OS WebSphere Servers.

Der Controller Region und die einzelnen Servants laufen in getrennten Adressenräumen (Regions). Die Controller Region enthält einen eigenen http Server, der HTTP Requests direkt entgegen nehmen kann, ggf. über definierte Port Nummern, um sie an die http Internal Transport Komponente der Controller Region weiterzureichen. Daneben kann natürlich auch ein externer http Server eingesetzt werden. Neben dem IBM HTTP Server kommen dafür z.B. der Microsoft Internet Information Services (IIS), Lotus Domino Enterprise Server und Oracle iPlanet Web Server in Frage.

Weiterhin existiert eine getrennte Region (Adjunct Region) für die Verarbeitung von Java Message Beans. JMBs werden durch einen JMB Klienten aufgerufen, der als ein beliebiges Java Programm (POJO, Plain Old Java Program) implementiert werden kann. JMBs selbst laufen in dem EJB Container einer Servant Region. Jede Servant Region enthält eine Messaging Komponente, an welche die Messaging Engine der Adjunct Region JMB Client Requests weiterreichen kann.

Die Servant Region implementiert die eigentliche WAS Funktionalität. Neben dem Servlet (Web-) Container und EJB Vontainer enthält sie weitere Funktionen für Data Replication, Security, Dynamic Caching und Naming Service.

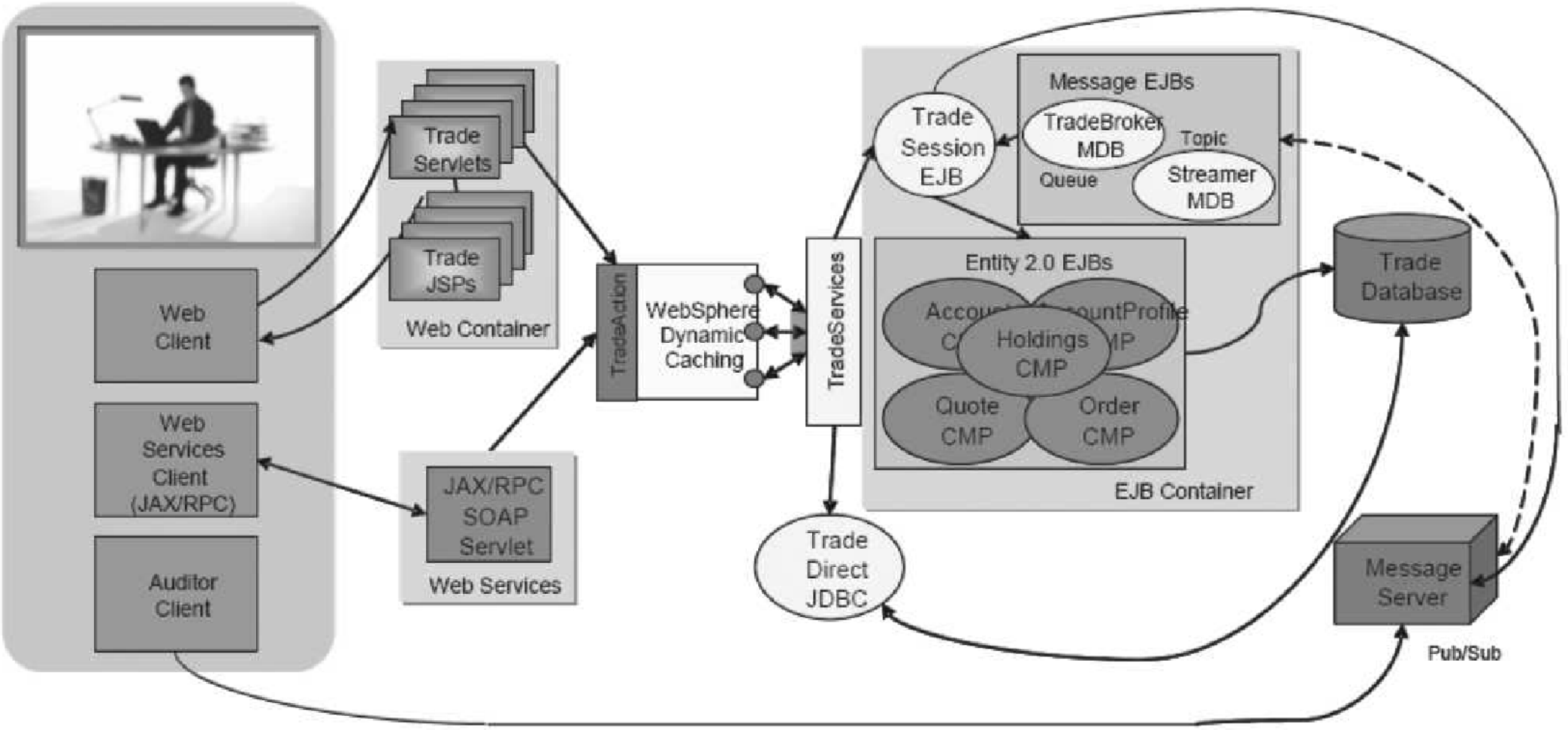
Zusätzliche Funktionen existieren für Web Services (siehe Thema 10 dieser Vorlesung).



WebSphere Skalierbarkeit unter z/OS

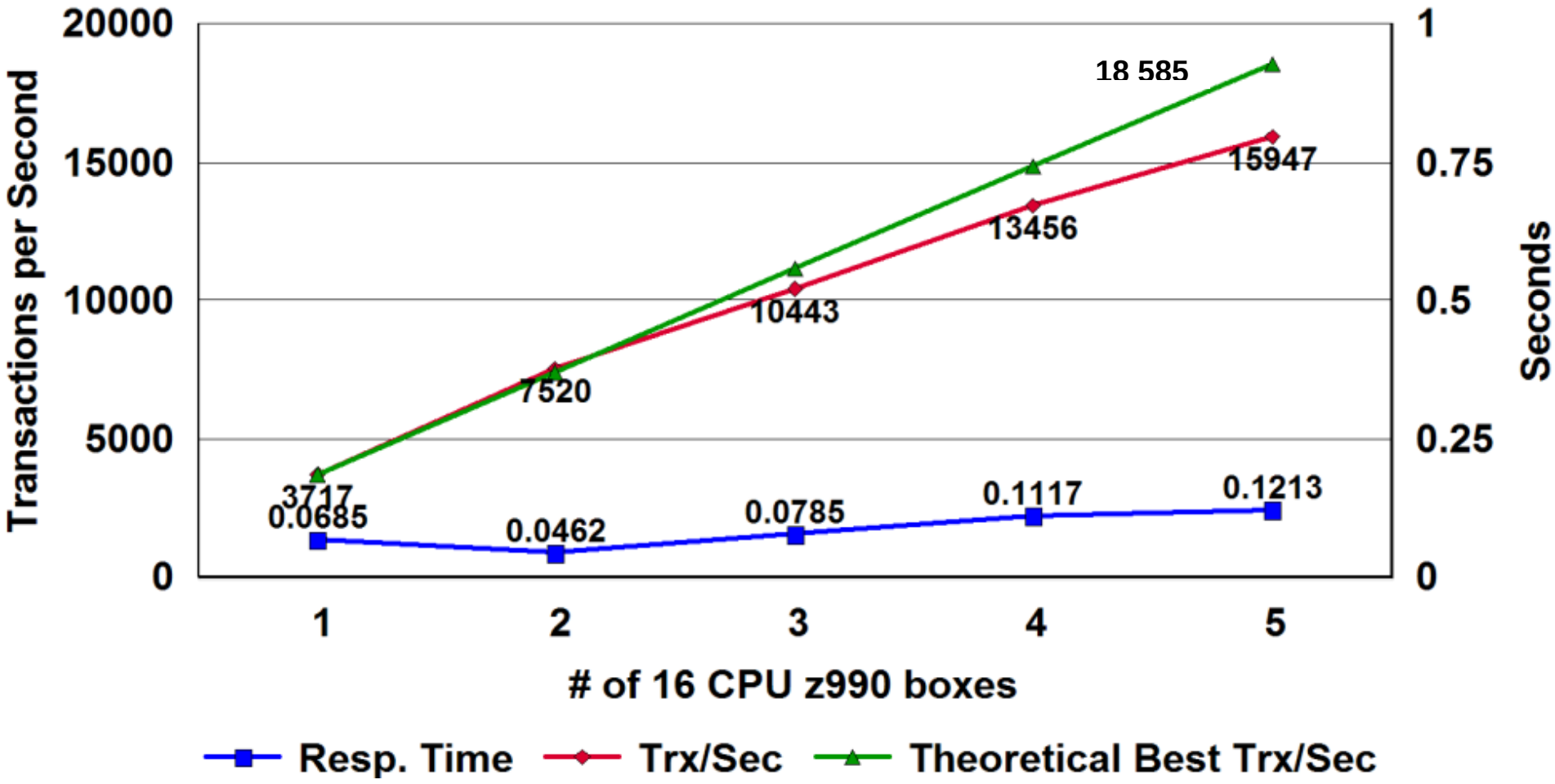
Skalierbarket ist ein Problem mit den meisten Unix, Linux and Windows Implementierungen. Die folgenden Abbildungen geben die Ergebnisse eines WebSpere Skalierbarkeit Testes unter z/OS wieder

Der Test wurde 2006 mit fünf z990 Systememen, mit je 16 CPUs durchgeführt. Eine weitere z990 mit 4 CPUs diente als Coupling Facility.



Trader Benchmark

z/OS benutzt für viele Tests ein Standard Client/Server Benchmark, die “Trader Application“. Dieses Benchmark wurde über die Jahre immer wieder verbessert. Dargestellt ist die Topologie des Trader V6.1 Benchmarks.



WebSphere Scalability für das Trader Benchmark, fünf Systeme

Ein einzelnes System mit 16 CPUs ist in der Lage, 3 717 Trader Transactions/s durchzuführen. Das theoretische Maximum für 5 Systeme mit 80 CPUs ist 5 x 3717 Tx/s = 18 585 Tansactions/s. Der tatsächlich gemessene Durchsatz ist 15 947 ttransactions/s,oder 15 947 / 18 585 = 86 % des theoretischen Maximums.

ftp://ftp.software.ibm.com/software/zseries/pdf/WASforzOSv6PerformanceReport.pdf, oder
http://jedi.informatik.uni-leipzig.de/de/VorlesMirror/ii/Vorles/Perform01.pdf