Open Source Application Server auf dem Mainframe

Fabian Czicholl, Technical Sales WebSphere on System z

Dennis Behm, Technical Sales WebSphere on System z

Agenda

- Einführung in den Mainframe
- Java unter z/OS
- Open Source Application Server auf dem Mainframe
- Nutzen und Anwendungsmöglichkeiten
  - Warum Open Source auf dem Mainframe?
- Vergleich Open Source AS z/OS <--> High End AS z/OS

Die „Büchse“

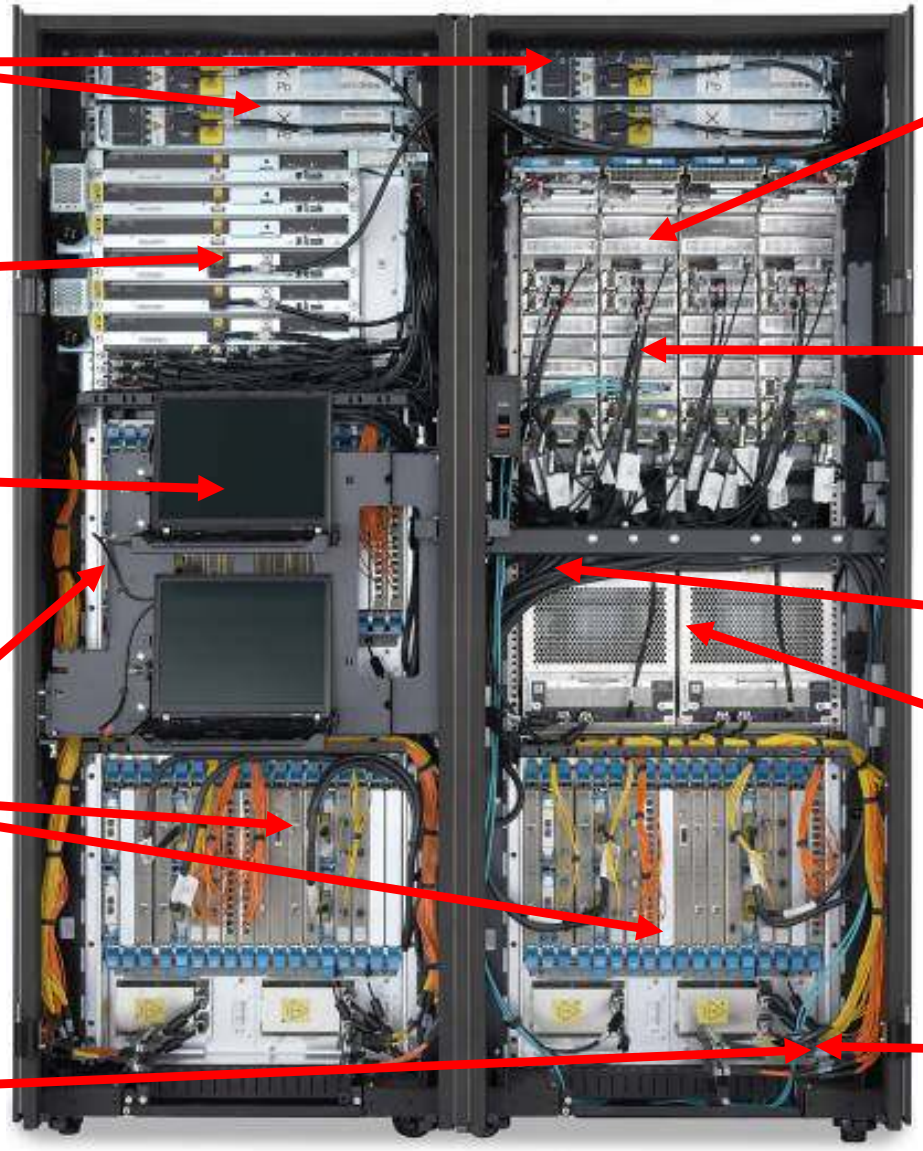
- bis zu 64 Prozessoren á 4,4 GHz
- 1,5 TB Hauptspeicher
- 288 GB pro Sekunde I/O Bandbreite

„Mainframe“

„Host“

„z“

System z10 – Ein Blick ins Innere



Einführung in den Mainframe (1/2)

- High-End Businessplattform
  - Großkunden
  - Größere Mittelstandskunden
  - Und jeder, der die Vorteile eines System z haben möchte
- Traditionelle Anwendungen auf dem Mainframe:
  - Transaktionssysteme
  - Buchungssysteme
- Anforderungen an Applikationen:
  - Sicherstellung der Hochverfügbarkeit von geschäftskritischen Anwendungen und Daten
  - Ausfälle können die Existenz des Unternehmens gefährden

Einführung in den Mainframe (2/2)

- Einsatzorte / Warum sind Kunden auf System z
  - Hochperformantes System
    - Lokale Datenanbindung (CICS, DB2)
    - Hipersockets
  - Sichere Umgebung
    - RACF Sicherheitskonzept
  - Hochverfügbare Umgebung
    - Hardwaredesign, sowie LPAR Konzept
  - Konsolidierungsumgebung
- Programmiersprachen:
  - COBOL, PL/I, C, C++, Assembler, REXX

➢ Modernisierung der historisch gewachsenen Applikationen
  - Bereitstellung dieser Daten und Anwendungen über mehrere Schnittstellen
  - Nutzen der neuen Java-Technologie auf dem Mainframe
  - Integrierung der breit verfügbaren Java-Skills des Arbeitsmarktes

JAVA auf dem Mainframe

- IBM SDK for z/OS
  - Umfasst alle Funktionen der JAVA v6 Spezifikation
- Wie harmoniert JAVA mit z/OS?
  - JZOS Toolkit
    - Teil der IBM SDK for z/OS
    - Zugriff auf z/OS Daten
    - Kommunikation mit der System Konsole
    - Einbindung von JAVA in JES
    - Started Tasks
    - Batchjobs

OpenSource Application Server auf z/OS

- JBoss

- Apache Geronimo
- Apache Tomcat

- WebSphere CE

- JOnAS*

- Glassfish*

*100% Java Code, läuft voraussichtlich, jedoch noch nicht getestet

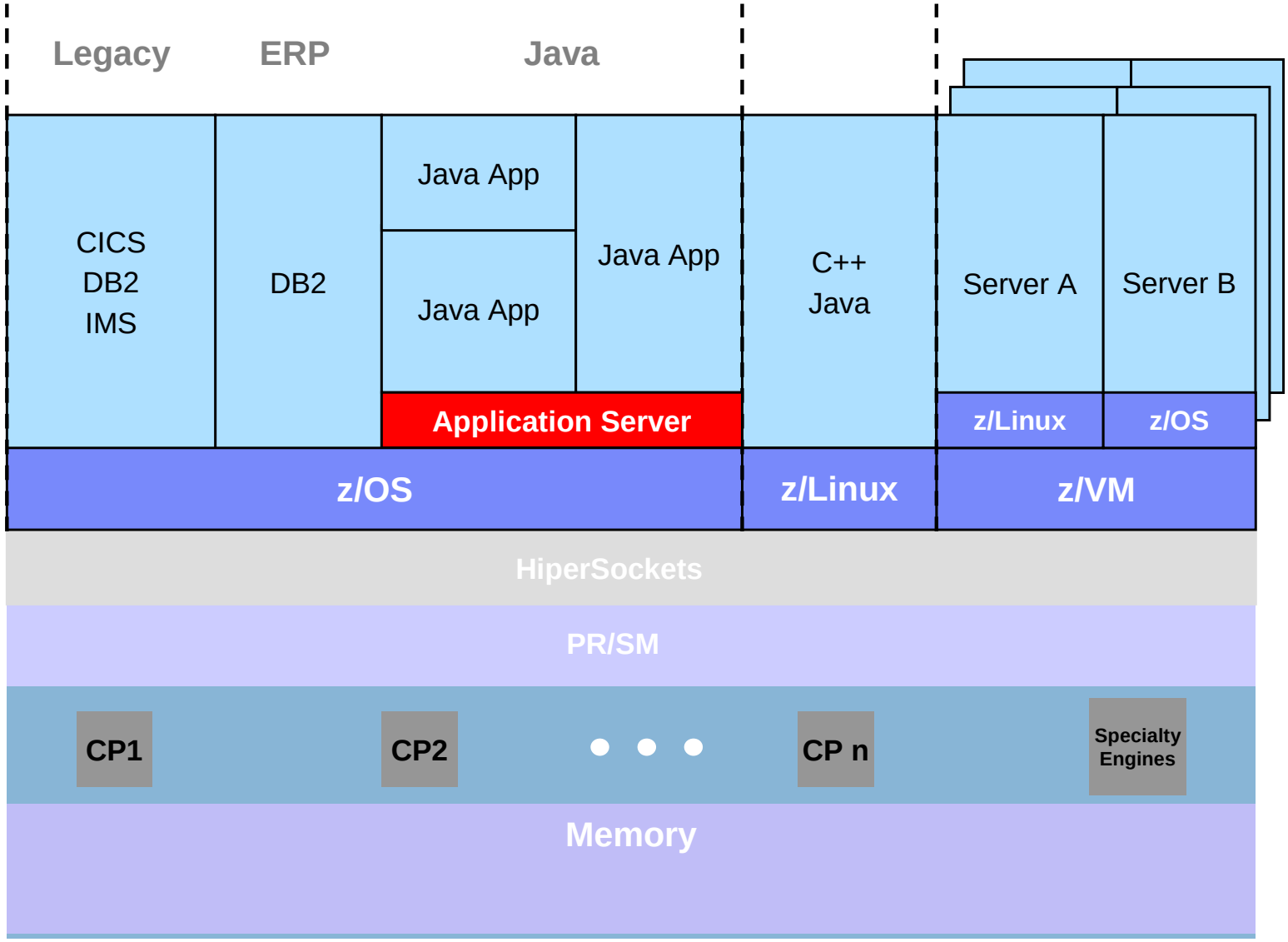
Gegenüberstellung der AppServer

	Aktuelle Version	Implementiert	Web Container
JBoss	4.2.2	J2EE 1.4 + Teile von JEE 5	Tomcat
Apache Geronimo	2.1.1	JEE 5	Tomcat / Jetty
WebSphere CE	2.0.0.2	JEE 5	Tomcat
JOnAS	4.9.2 (stable version)	J2EE1.4	Tomcat (5.5.26) / Jetty (5.1.10)

Der Mainframe – Unbekanntes Terrain??

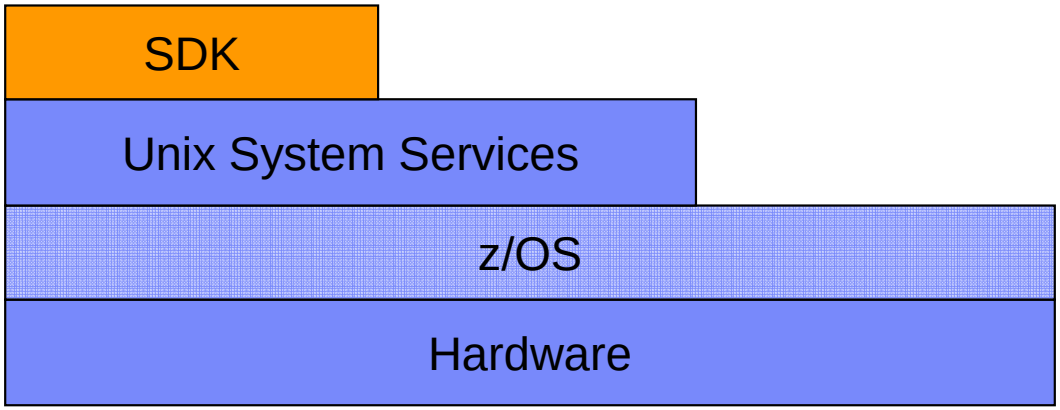
- IBM System z
  - 360 → 390 → zSeries → System z
  - z: Zero Downtime
  - Redundante Auslegung aller Komponenten
  - Kryptochips + strenges Sicherheitskonzept
  - Viele CPUs, viel Arbeitsspeicher, jedoch keine Festplatten
  - LPAR-Konzept
    - „Hermetisch abgeriegelte“ logische Partitionen
    - Höchste Sicherheitszertifizierung
    - Max. 60 LPARs pro System

LPAR Konzept



Mainframe Architektur

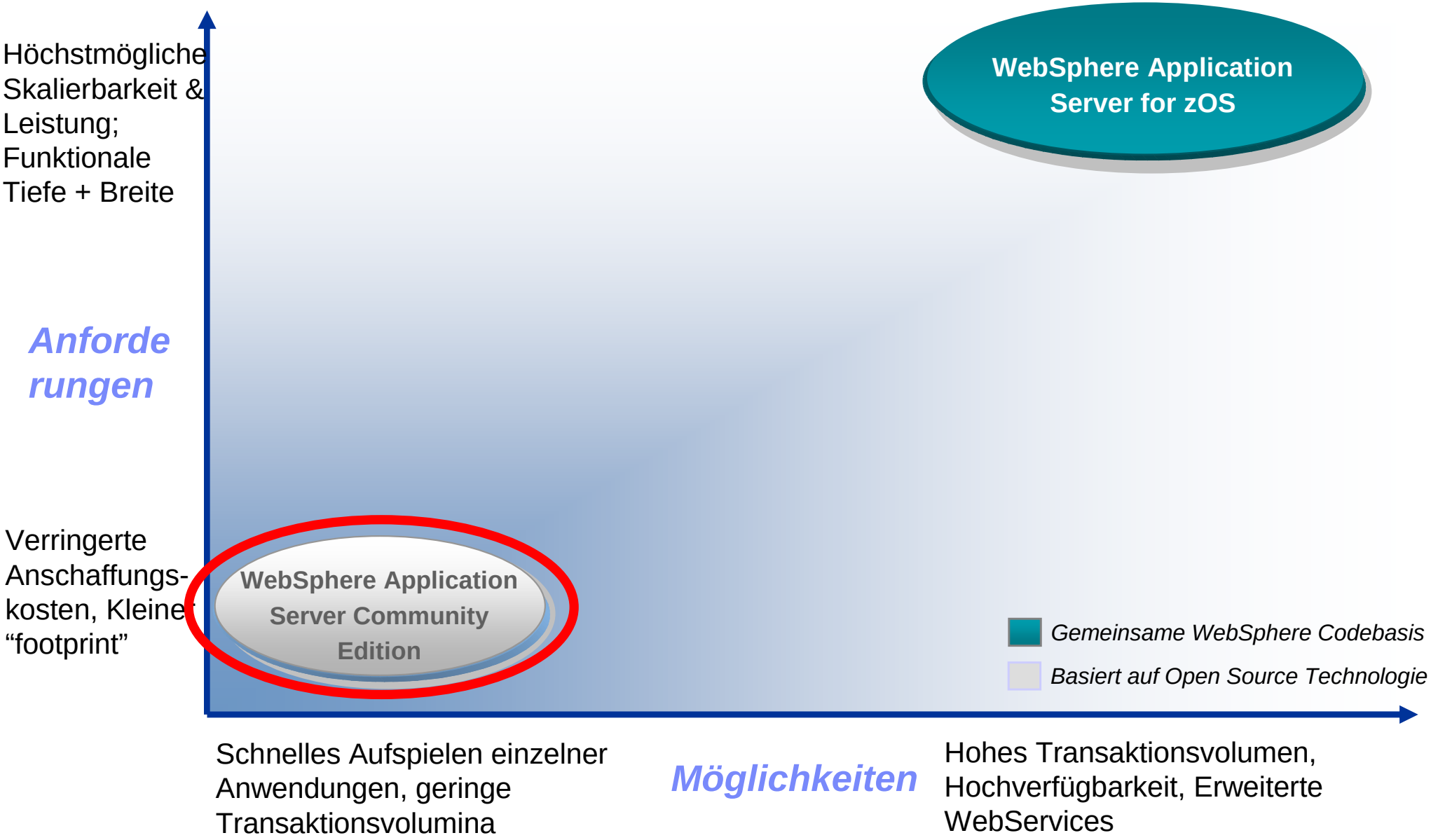
- Systemumgebung:
  - z/OS MVS
  - Unix System Services - USS
    - baut auf MVS auf, läuft innerhalb von MVS
    - POSIX-konform
- IBM SDK
  - IBM 31-bit SDK for z/OS, Java Technology Edition, V6
  - IBM 64-bit SDK for z/OS, Java Technology Edition, V6



Hardware Charakteristika + Performance

- System z10 Hardware-Leistung:
  - 64 Chips à 4,4 GHz
  - Max. 1,5 TB Hauptspeicher
  - 48 * 6 GB/s InfiniBand I/O Leistung
- Workload Management - z/OS WLM
  - Granularste, dynamische Ressourcenverteilung
    - Prozessoren
    - Hauptspeicher
    - Netzwerkadapter
  - Sichere Antwortszeiten für
    - Application Server
    - Tageszeiten
    - User
    - Internetadressen
- Lokale Connectoren - Type 2 Connector
  - Eleminierung des TCP/IP Overhead
  - Ansonsten HiperSockets wenn nicht in gleicher LPAR
- Extrem gute Skalierung
  - Coupling Facility
- Spezialprozessoren entlasten die Hauptprozessoren

Einsatz OpenSource Application Server vs. Highend Server



OpenSource AppServer on z/OS vs. WAS z/OS

	OpenSource AppServer on z/OS	WAS z/OS
Local backend connectors	✓	✓
WLM integration	Address Space ✓	✓
Transaction based goals		✓
Internal Clustering		✓
Dynamic Caching		✓
Clustering (incl. advanced)	*	✓
Dynamic Cluster		✓
Load balancing	*	✓
Failover Concepts		✓
zAAP Support (“dedicated” HW)	✓	✓

* theoretisch

Kostenaspekte des Mainframes

- Konsolidierung von Workload
  - Virtualisierungsoptionen
- Reduzierte Raumkosten
- Reduzierte Stromkosten
  - Projekt BigGreen
- Spezialprozessoren
  - zAAP für Java und XML Workload
  - zIIP für XML, IPSec und DB2
  - IFL für Linux Workload (Linux für System z)

Einsatzszenarien

- Bestehende Legacysysteme (CICS)
  - Exponieren als WebServices
  - Einbinden in Java-Anwendungen
  - Bereitstellen von Funktionalität der Host-Programme im Intra-/Internet
- Datenbanken
  - Applikationen mit rechenintensiven Operationen auf Host-Daten
- „Der Dinosaurier hat nun auch Java gelernt“

Fazit

- Open Source Application Server wählen, wenn
  - Zugriff auf Daten des Mainframes notwendig sein soll
  - Kleine Anwendung / Projekt / Pilot / Test
- Vorteile:
  - Datennähe (Performance)
  - Einbindung in bestehendes Sicherheitskonzept des Mainframes
  - Administrierbar durch Management-Software
  - Keine Anschaffungskosten
- Weitere Überlegungen
  - Wird das Projekt irgendwann „größer“?
  - Wie wichtig ist eventuell Clustering, Ausfallsicherheit?
  - Upgrade-Fähigkeit berücksichtigen


IBM
Fragen?
Vielen Dank!
GENERATION z. GENERATION ZUKUNFT.