J2EEKurs

J2EE—eine Plattform für betriebliche Anwendungen

Peter Thiemann

Universität Freiburg, Germany

Sommercampus J2EEKurs, Freiburg, Germany,
10.-14.10.2005

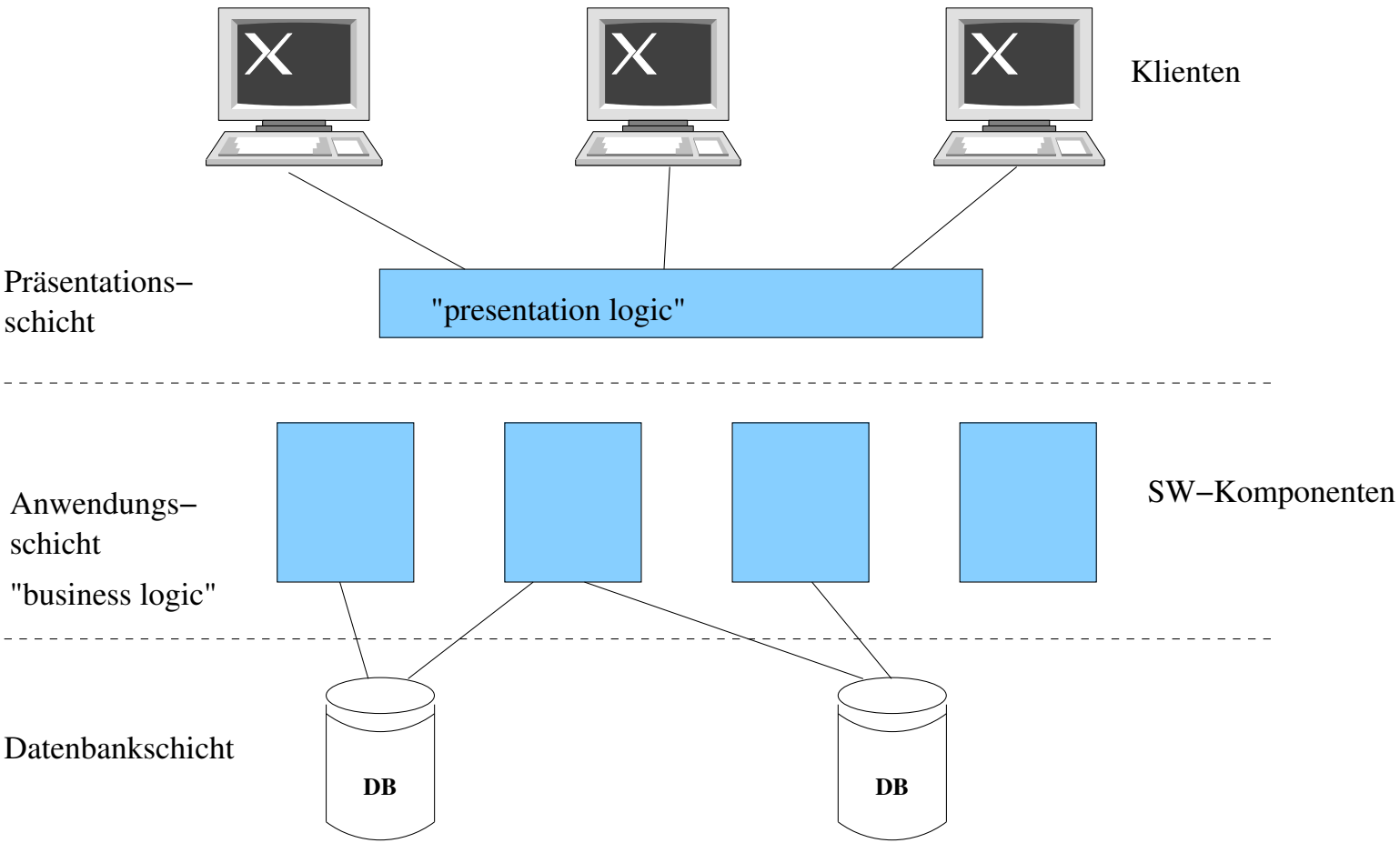
Was ist eine Plattform?

- Richtlinien für Entwurf und Entwicklung von Programmen
- API Spezifikationen
- Unterstützt durch Programmierwerkzeuge und -umgebungen
- Dienste der Plattform
  - Nebenläufigkeit und Verteilung
  - Persistenz
  - Transaktionen
  - Sicherheit
  - Lastbalancierung
  - Fehlertoleranz
  - Zuverlässigkeit

Was ist eine betriebliche Anwendung?

- Software, die Geschäftsfunktionen wie Buchführung, Produktionsplanung, Kundenverwaltung oder Lagerverwaltung unterstützt. Z.B. enterprise resource planning (ERP), customer relationship management (CRM), and supply chain management (SCM).
- Software, die auf zentralen Servern läuft und gleichzeitig viele Benutzer bedient.
- Anwendung, die über das Netz (Intra- oder Internet) verfügbar ist und (meist) auf mehreren Rechnern verteilt implementiert ist.
- Client-Server-Architektur.
- Architektur: 3-tier bzw n-tier Architektur oder Web-Architektur.

3-Schichten-Architektur



Was verspricht J2EE?

Was SUN sagt

Java 2 Platform, Enterprise Edition (J2EE) defines the standard for developing component-based multitier enterprise applications. J2EE simplifies building enterprise applications that are portable, scalable, and that integrate easily with legacy applications and data. J2EE is also a platform for building and using web services. It incorporates web services standards such as those in the WS-I Basic Profile. This means that web services in a J2EE-compliant environment can interoperate with web services in non-J2EE environments such as .NET.

- Achtung neuer Name ab nächster Release:
  Java Platform, Enterprise Edition 5 (Java EE 5).

Was verspricht J2EE?

Was andere sagen

Java 2 Platform, Enterprise Edition (J2EE) is a programming platform for developing and running distributed multi-tier architecture applications, based mostly on components running on an application server. The Java EE platform is defined by a specification. The specification serves as a standard because providers must agree to certain conformance requirements to declare their products as Java EE compliant.

J2EE vs. .NET

- Enterprise JavaBeans (EJB, Teil von J2EE)
  - Java-basierte Komponentenarchitektur
  - IDE-Unterstützung durch z.B. ECLIPSE
  + unabhängig von HW/BS-Plattform
  – sprachabhängig (Java). CORBA möglich, aber selten.
- Microsoft .NET
  - Microsofts Komponentenarchitektur (Fortentwicklung von DCOM, seit 2002)
  - IDE-Unterstützung durch VisualStudio.NET
  + sprachunabhängig: Komponenten können in jeder von .NET unterstützten Sprache geschrieben werden. Z.B.: C#, J#, VisualBasic.NET, C++, ...
  – abhängig von MS.NET (Microsoft: plattform-neutral)

Java für Cient-Server-Anwendungen

- Eigenschaften von Java
  - Vorteile: Stabilität, Plattformunabhängigkeit und Sicherheit.
  - Nachteile: geringe Geschwindigkeit und fehlende Kooperation mit OS-APIs.
- Java im Servereinsatz
  - Wichtig: Stabilität und Sicherheitsarchitektur
  - Portabilität auf viele verschiedene Betriebssysteme
  - Geschwindigkeit unkritisch, da Netzwerk und Datenbank den Flaschenhals bei Serveranwendungen bilden
  - Entwicklung erster Serverkomponenten für Java (ca. 1997)

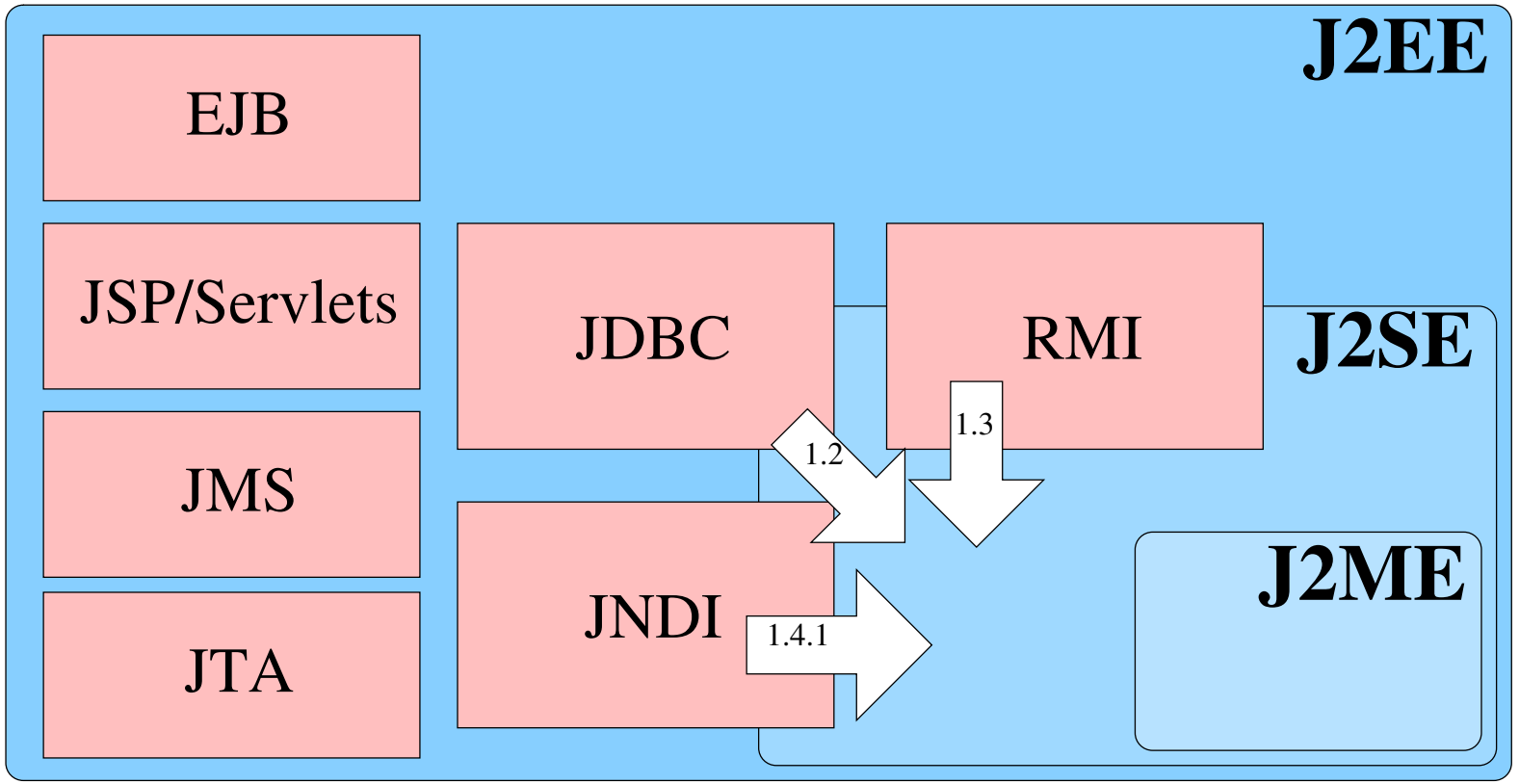
Java Editionen

- J2EE (Java 2 Enterprise Edition)
  Umfangreiche, standardisierte Bibliotheken für Serverfunktionalitäten, z.B. Datenbankanbindung, Verteilte Anwendungen
  Einsatz auf Mainframes und Servern
- J2SE (Java 2 Standard Edition)
  Plattform zur Entwicklung von Anwendungen und Applets. Sprachumfang identisch zu J2EE. Weniger Bibliotheken.
  Einsatz auf Home- und Personal Computern
- J2ME (Java 2 Micro Edition)
  Beschränkter Sprachumfang, auf geringen Speicherverbrauch optimiert. Sehr wenige Bibliotheken.
  Einsatz auf Java-fähigen Handys und Handhelds

J2EE ist keine neue Technik

- J2EE ist eine Sammlung *miteinander kombinierbarer* Enterprise-APIs. Es bildet die Plattform für eine funktionierende Enterprise-Server Architektur.
- Vorteile des J2EE-Ansatzes
  - Bündelung der APIs zu einer gemeinsam weiterentwickelten Edition
  - Differenzierung der Aufgabenbereiche der einzelnen APIs
  - Referenzimplementierung der Schnittstellen durch Sun als kleinster gemeinsamer Nenner (nicht für den Einsatz in geschäftskritischen Anwendungen gedacht)
  - Kompatibilitätskennzeichen von Sun für Produkte von Drittanbietern

Editionen im Überblick



J2EE APIs

- Servlets, JSP: JavaServer Pages
  - Generieren von dynamischem Webinhalt
  - `http://java.sun.com/products/servlet/`
  - JSP: Generierung von XML-Dokumenten aus Mustern
  - `http://java.sun.com/products/jsp/`
- EJB: Enterprise Java Beans
  - Serverseitige Komponentenarchitektur
  - Rahmen für Datenbankzugriff und Geschäftslogik
  - `http://java.sun.com/products/ejb/`
- JNDI: Java Naming and Directory Interface
  - Namens- und Verzeichnisdienst
  - Bekanntmachen und Auffinden von Java-Objekten im Netz
  - `http://java.sun.com/products/jndi/tutorial/`

J2EE APIs/2

- RMI (IIOP)
  - Remote Method Invocation (Internet Inter-ORB Protocol)
  - Entfernter Methodenaufruf
  - `http://java.sun.com/docs/books/tutorial/rmi/`
- JDBC
  - Java Database Connectivity
  - API für Datenbankverbindungen insb. zu relationalen DBs
  - `http://java.sun.com/products/jdbc/`

J2EE APIs/3

- JMS: Java Messaging Services
  - Zuverlässige, asynchrone Kommunikation von Geschäftsdaten und -ereignissen
  - `http://java.sun.com/products/jms/`
- JTA: Java Transaction API
  - Interface für transaktionale Programmierung
  - `http://java.sun.com/products/jta/`

Die Rolle der APIs