Seminář Java

X

J2EE

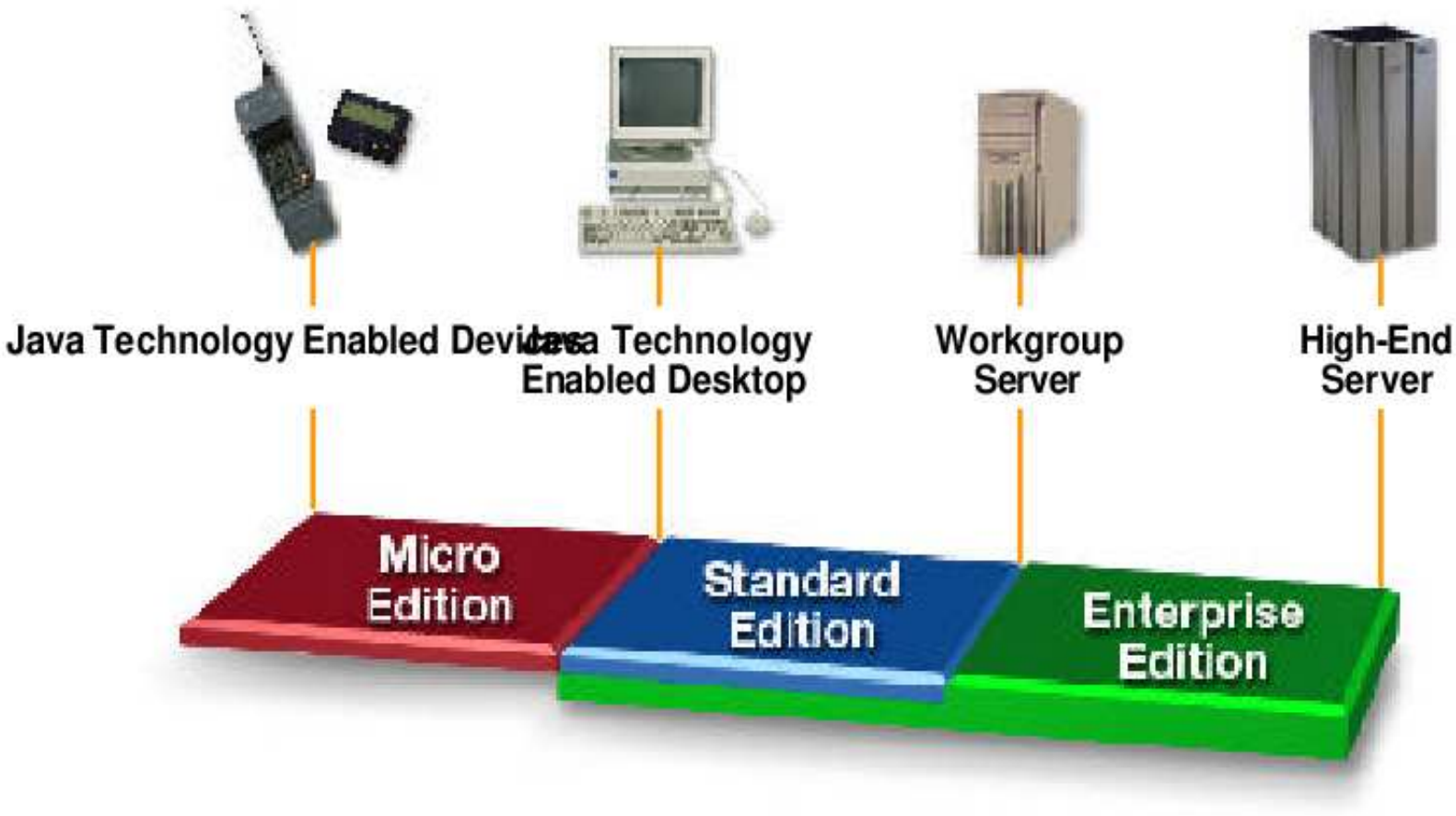
J2EE

- Složitost obchodních aplikací ↑
- → robusní, distribuované, spolehlivé aplikace s transakcemi na straně serveru, klientské aplikace co nejjednodušší
- Snaha : Návrh, vývoj, testování, spotřeba zdrojů = co nejméně peněz

- snížení nákladů pro vývoj těchto aplikací v Jave = Java Enterprise Edition (J2EE)
- J2EE = komponentní přístup pro návrh, vývoj, sestavení a nasazení obchodních aplikací
- Java = platformově nezávislé (Windows Server, Linux, Unix, . . .)

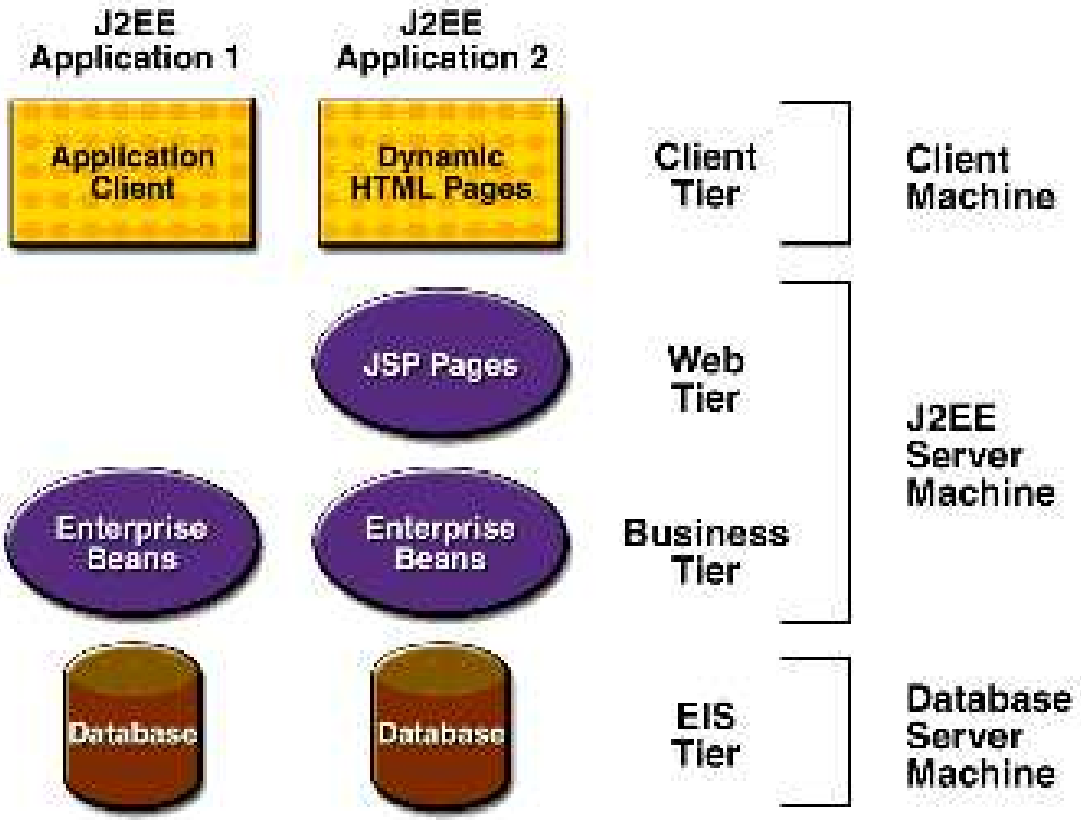
- J2EE nabízí : vícevrstvý distribuovaný aplikační model, opakovaně použitelné komponenty, flexibilní kontrolu transakcí, . . .

Kde se J2EE používá?



Distribuované vícevrstvé aplikace

J2EE platforma používá distribuovaný vícevrstvý model pro obchodní aplikace. Co znamená distribuovaný, vícevrstvý, obchodní logika, prezentační vrstva?



J2EE komponenty

- Celá J2EE aplikace je složena z komponent
- J2EE komponenta = nezávislá funkční SW jednotka, která je vložena do J2EE aplikace a která komunikuje s dalšími komponentami
- Specifikace J2EE definuje tyto typy komponenty:
  - Klientské aplikace, applety (klient)
  - Java servlet, JSP (WEB komponenty)
  - Enterprise JavaBeans (EJB) komponenty (server)

J2EE klienti

- Web klienti
  - 2 části – dynamická stránka (HTML, XML), browser
  - „thin" klient
  - http
- Applet
  - Malá klientská aplikace bežící v prohlížeči
  - Komunikace například se servletem
  - http
- Aplikace
  - Aplikace běžící na straně klienta
  - GUI (Swing)
  - „thin" klient, neobsahuje obchodní logiku
  - RMI

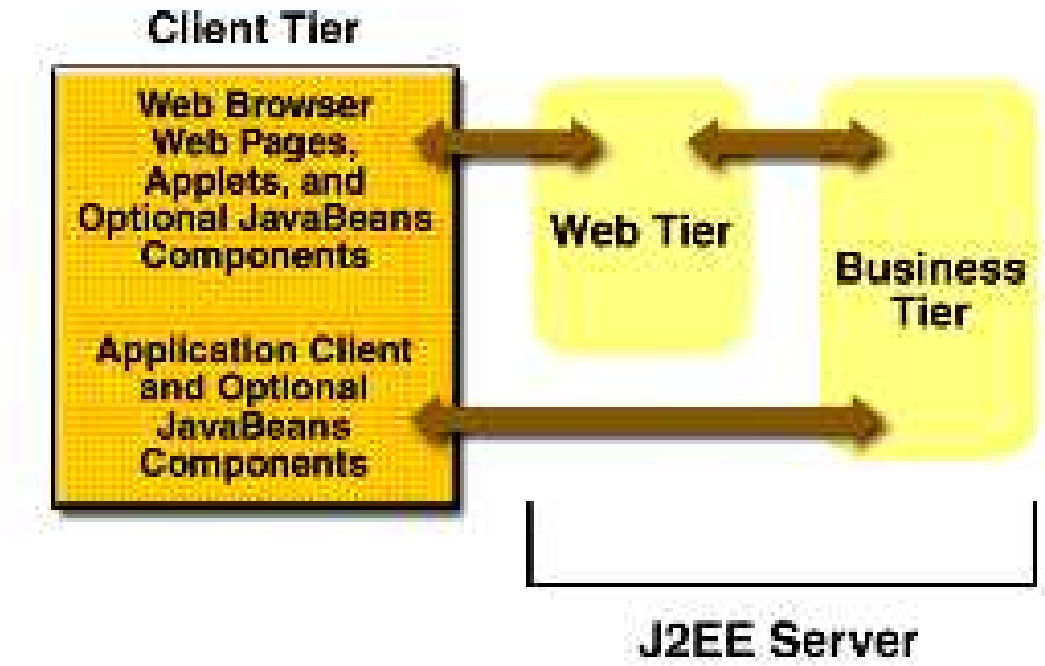
J2EE web komponenty

- Servlety
- Stránky vytvořené technologií JSP

Tyto technologie budou probírány na další přednášce

J2EE komunikace

Komunikace přes jednotlivé vrstvy



J2EE Containers

- Sdílení prostředků, transakce, . . .
- Složité pro implementaci – J2EE server
- Containers = rozhraní mezi komponentou low-level platformově závislou funkcionalitou
- „system-level" transakce, zdroje
- WebContainer, EJBContainer

Enterprise JavaBeans (EJB)

- server-side komponenta, obchodní logika aplikace
- „jádro aplikace"
- Běží v EJB container

- Zjednodušení aplikace, efektivní řízení zdrojů kontejnerem
- Znovupoužitelné komponenty
- Kdy použít EJB?
  - Lehce rozšířitelná aplikace
  - Transakce, integrita dat
  - Různé typy klientů

Typy EJB

- **Session**
  - Interakce s klientem (aplikace, web)
- **Entity**
  - Reprezentace „obchodní" entity
  - Perzistentní objekt
- **Message-Driven**
  - Jako „posluchač" pro JMS API
  - Zpracování zpráv asynchronně

Session Bean

- Reprezentuje jednoho klienta uvnitř aplikačního serveru
- Klient volá metody session beanu
- Session bean není sdílený = jeden klient má jeden session bean
- „nákupní košík"
- 2 typy
  - **Stateless**
    - Stav neobsahuje specifická data pro klienta
    - Neudržuje si stav pro klienta, pouze při provádění volané metody
    - „vrať seznam zaplacených faktur"
  - **Stateful**
    - Stav obsahuje specifická data pro klienta
    - Každý klient má svůj
    - „nákupní košík"

Session Bean – kdy použít?

- Obecně
  - Pouze jeden klient má přístup
  - Stav beanu není perzistentní
- Stateful session bean
  - Stav reprezentuje interakci mezi beanem a klientem
  - Je třeba udržovat stav mezi voláními metod
  - Zjednodušuje pohled klienta na ostatní komponenty
- Stateless session bean
  - Všechny ostatní případy
  - Stav beanu neudržuje specifická data pro klienta

Typy EJB

- **Session**
  - Interakce s klientem (aplikace, web)
- **Entity**
  - Reprezentace „obchodní" entity
  - Perzistentní objekt
- **Message-Driven**
  - Jako „posluchač" pro JMS API
  - Zpracování zpráv asynchronně

Entity Bean

- Reprezentuje „obchodní objekt" v systému trvalého uložení
- Zákazník, faktura, objednávka, . . .
- systém trvalého uložení = obvykle relační databáze („řádek tabulky")
- Hlavní rozdíly oproti session beanům
  1. Entitní bean je perzistentní
  2. Sdílený přístup
  3. Případná relace s jinými ent. beany
- 2 typy
  - CMP (Container–Managed Persistence)
    - Veškerý přístup do DB obstarává Container
    - Vytváření, změna hodnot apod.
    - Jednoduchá implementace
  - BMP (Bean–Managed Persistence)
    - Přístup do DB implementuje programátor v jeho metodách
    - Složitá implementace

Entity Bean – kdy použít?

- Pro reprezentaci „obchodní" entity. Kreditní karta bude EntitníBean, Verifikace kreditní karty SessionBean.
- Stav beanu musí být perzistentní
  1. Po zastavení aplikačního serveru musí být zachována perzistentnost (relační databáze)

Typy EJB

- **Session**
  - Interakce s klientem (aplikace, web)
- **Entity**
  - Reprezentace „obchodní" entity
  - Perzistentní objekt
- **Message-Driven**
  - Jako „posluchač" pro JMS API
  - Zpracování zpráv asynchronně

Message-Driven Bean

- Bean který umožňuje J2EE aplikaci zpracovávat zprávy asynchronně
- JMS
- Hlavní rozdíly oproti session a entity beanům
  1. Klient nekomunikuje přes rozhraní
  2. Klient zasílá zprávy
  3. Všechny instance jsou si ekvivalentní (zpráva jakémukoliv mess. beanu)
  4. Možnost zpracovávat zprávy od několika klientů

Definice přístupu pro klienta

- Pomocí rozhraní (interface)
- 2 typy rozhraní
  1. Vzdálené
     (a) Klient na různém JVM
     (b) web komponenta, aplikační klient, jiný bean
  2. Lokální
     (a) Stejné JVM
     (b) web komponenta, jiný bean

Rekapitulace

- J2EE pro vývoj robusních obchodních aplikací
- Urychluje vývoj, komponentní architektura
- Obchodní logika na serveru
- Oddělení obchodní logiky a prezentační vrstvy
- „thin" klient, různé typy
- Efektivní řízení zdrojů zajišťuje server

Implementace J2EE – aplikační servery

- Sun Java System Application Server Edition 8 (J2EE SDK v1.4)
- JBoss (www.jboss.org)
- Oracle Application Server
- Borland Application Server
- SyBase Application Server

Kde lze nalézt takové aplikace?

- Bankovnictví
- Pojišťovny
- Podnikové systémy
- Složité www aplikace

Co příště?

- Webové aplikace
- Servlety
- JSP