



Server-side technologie pro webové aplikace

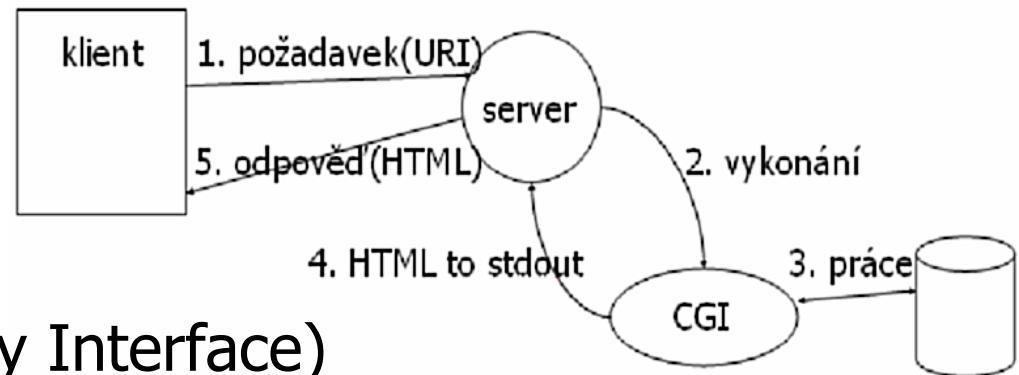
PIA 2011/2012
Téma 6

Server-side scriptování

- Cíl
 - dynamické generování webového obsahu/rozhraní
 - integrace legacy aplikací
- Prostředky
 - externí aplikace napojená na web server
 - aplikační server, který řeší i webový přístup
 - webový kontejner, který umožňuje aplikační komunikaci
 - embedded skriptování

Externí aplikace

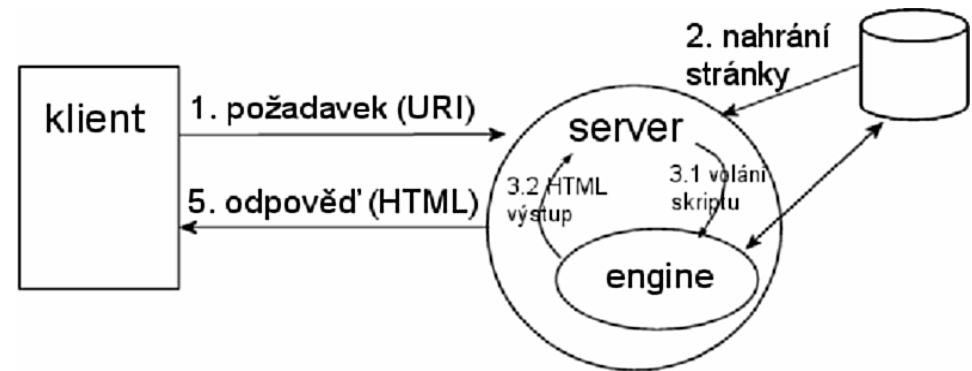
Server spouští samostatný spustitelný soubor



- **CGI** (Common Gateway Interface)
 - kompilované (C) nebo interpretované (perl) jazyky
 - identifikováno a spuštěno serverem na základě URI
 - flexibilita, integrace legacy aplikací
 - ale: run-time náklady, netransparentnost, bezpečnost
- `mod_perl`
 - interpret perl vestavěný v Apache serveru
 - snižuje režijní náklady (spawn, šifrovací cache)

Scripty zabudované do HTML

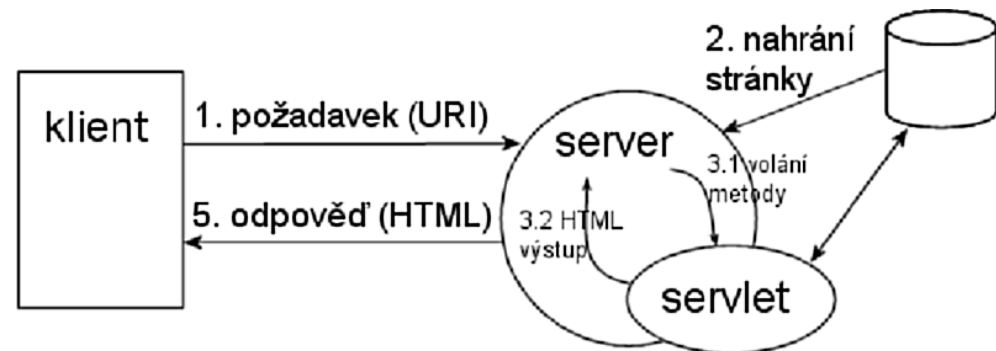
Server obsahuje
interpretovací engine,
scripty součástí stránek



- server parsuje kód stránek, přes engine spouští script
 - » skripty interpretovány serverovým modulem/rozšířením
- skripty ani data nejsou persistentní
- rychlý vývoj, malý footprint
- ale: nedá se škálovat, komplikovaná integrace

Servlety

(Externí) web-aware,
kompilované aplikace



- aplikace (servlety) běží v kontejneru = část serveru
- objekty jsou persistentní mezi požadavky
- nízké režijní náklady, vysoký výkon, vstup aplikace
- větší run-time náklady, vývojový čas



Technologie



CGI – Common Gateway Interface

Externí skripty/aplikace spouštěné *httpd*

- specifikace: v1.1 – NCSA (hoo.hoo.ncsa.uiuc.edu)
- první, dnes zastaralé

- Klient posílá data přes GET nebo POST
 - URI / tělo požadavku
 - application/x-www-form-urlencoded
- Server spouští CGI, vrací jeho *stdout*
 - skript spouštěn v samostatném procesu
 - generuje kompletní odpověď (včetně HTTP hlaviček)
 - hlavičky dotazu a parametry prostředí přes proměnné

CGI: vybrané proměnné

- Požadavek
 - QUERY_STRING, REQUEST_METHOD
 - CONTENT_TYPE, CONTENT_LENGTH
 - HTTP_* pro ostatní hlavičky (HTTP_ACCEPT atd.)
- Informace o serveru
 - SERVER_SOFTWARE, SERVER_NAME, SERVER_PORT
 - PATH_INFO (jaké bylo URI)
 - SCRIPT_NAME
- Informace klienta
 - REMOTE_HOST, REMOTE_ADDR
 - HTTP_USER_AGENT
 - HTTP_AUTHORIZATION

Scripty zabudované do HTML

- PHP (PHP: Hypertext Preprocessor)
 - <http://www.php.net>, R.Lerdorf et al.; Apache etc.
 - malý footprint, rychlé, integrace db
 - mnoho vlastností, open source
 - ale: špatně škálovatelné, nehomogenní
 - použito: O'Reilly, freshmeat, root.cz
- JSP (JavaServer Pages) – ne až tak úplně !
 - <http://java.sun.com/products/jsp/>; Java EE platforma (Java Platform, Enterprise Edition)
 - Tomcat etc.
 - podpora, přenositelnost, integrace ostatních J2EE služeb
 - použito: Rational, Sun

Scripty zabudované do HTML

- ASP/ASP.NET (Active Server Pages)
 - <http://msdn.microsoft.com/asp.net/>; Microsoft (DCOM) platforma; IIS, Apache
 - výkon, integrace BackOffice
 - ale: single vendor, platformová nestabilita
 - použito: Microsoft, Atlas, iDnes, Compaq
- Oracle, Sybase
 - zabudované jazyky PL/SQL
 - pevná integrace databáze engineu

Další možnosti

- Apache SSI
 - jednoduché HTML rozšíření pro stranu serveru
- Aplikační servery
 - HTTP server částí aplikačního serveru
 - Oracle – PL/SQL volané přes URI, dává HTML
 - EJB servery – distribuované Java aplikace, které používají HTML pro uživatelský vstup
- Aplikační frameworky
 - těsná integrace serveru, scriptovacího jazyka, knihoven a vývojového prostředí
 - Zope – aplikační server + webové IDE pro Python

Server-side includes

Útržky HTML kódu vkládané serverem při posílání dokumentu

– www.apache.org/docs/mod/mod_include.html

- Použití

- vytvoření standardního designu stránky
- vkládání opakovaných částí HTML
- vkládání samostatně editovaného obsahu

Základní SSI

Zdroj base.html

```
<!--#set var="nadpis" value="PIA"-->
<!--#include virtual="../_header.html" -->

<p>Na této stránce jsou ...
```

Zdroj
../_header.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
 3.2//EN">
<html> <head>
<title><!--#echo var="nadpis" --></title>
</head>

<body>
<h1><!--#echo var="nadpis" --></h1>
<hr>
</div>
```

Související technologie

- Abyste mohli implementovat webovou aplikaci, můžete potřebovat také
 - jmenné služby (LDAP, JNDI)
 - zabezpečení, autentikaci (GS API, JAAS)
 - persistenci, přístup k datové vrstvě (JDBC, EJB, Hibernate)
 - transakce (JTA)
 - práci s XML (SAX, JAXP)



Architektury aplikací

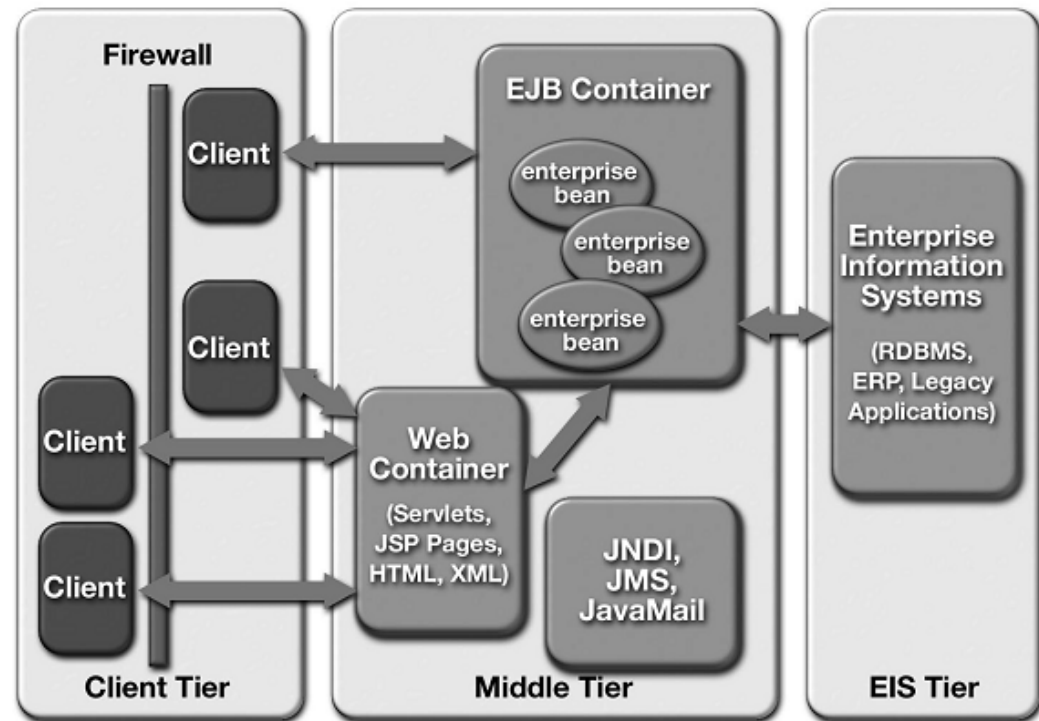


Základní struktury

- Vždy klient-server
 - tlustý klient × tenký klient
- Nativní klient
 - logika v klientu, data na serveru
 - aplikační protokol
 - Delphi, Java, Oracle Forms
- Aplet
 - bez instalace, omezení přístupu na hostitele
 - ActiveX, Java, .NET 3.0

Třívrstvá architektura

- Tenký klient
 - GUI rendering
 - » HTML-CSS-JS
- Aplikační server
 - business logika
 - vazba na legacy
 - » Java, .NET, ...
 - » EJB, JCA
 - podpora GUI
 - » JSP, JSF, ASP.NET
- Datový server
 - obvykle RDBMS
 - » MySQL ... Oracle



Další alternativy

- Mixovaný model
 - tlustý klient někde, tenký jinde
- Webové služby
 - přímá komunikace mezi aplikačními vrstvami
 - RPC: aplikace přistupující k jiné aplikaci
 - WSDL: popis rozhraní aplikace
 - SOAP: protokol pro přenos vzdáleného volání přes HTTP
 - UDDI: registr dostupných rozhraní
 - REST: webové aplikace/služby s plain HTTP a čistými URL