



Vývoj aplikací v prostředí .NET

© Katedra řídicí techniky,
ČVUT-FEL Praha

Přednáška – 9. týden

Omluva

*Vzhledem k tomu,
že se předmět přednáší
i v angličtině a není
v mých silách vyrobit
dvojjazyčné prezentace,
nejsou všechny snímky
v češtině.*

Děkuji za pochopení.



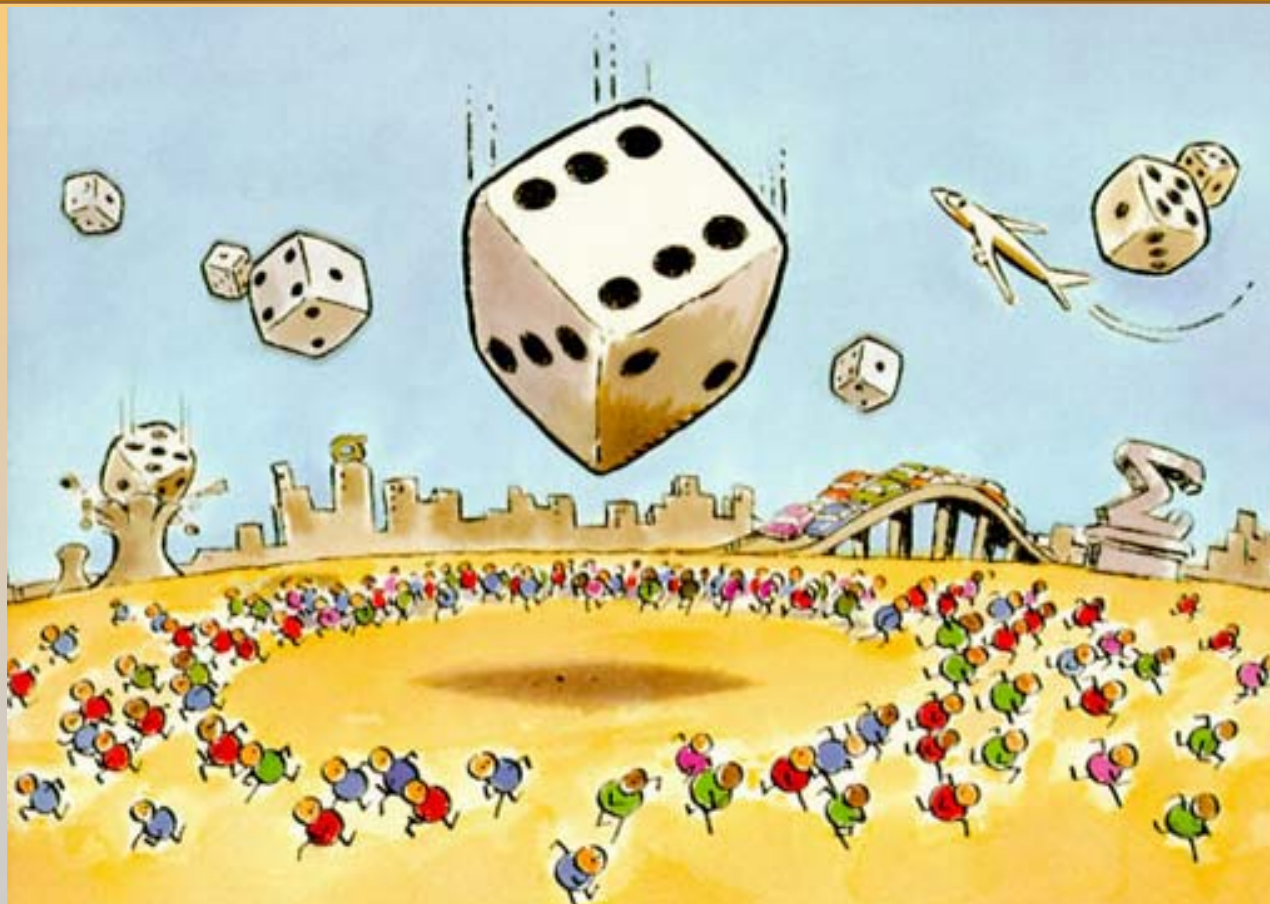
NETwork of ASP.NET

vytvoříme si webové stránky v C# ASP.NET





Napřed trochu statistiky

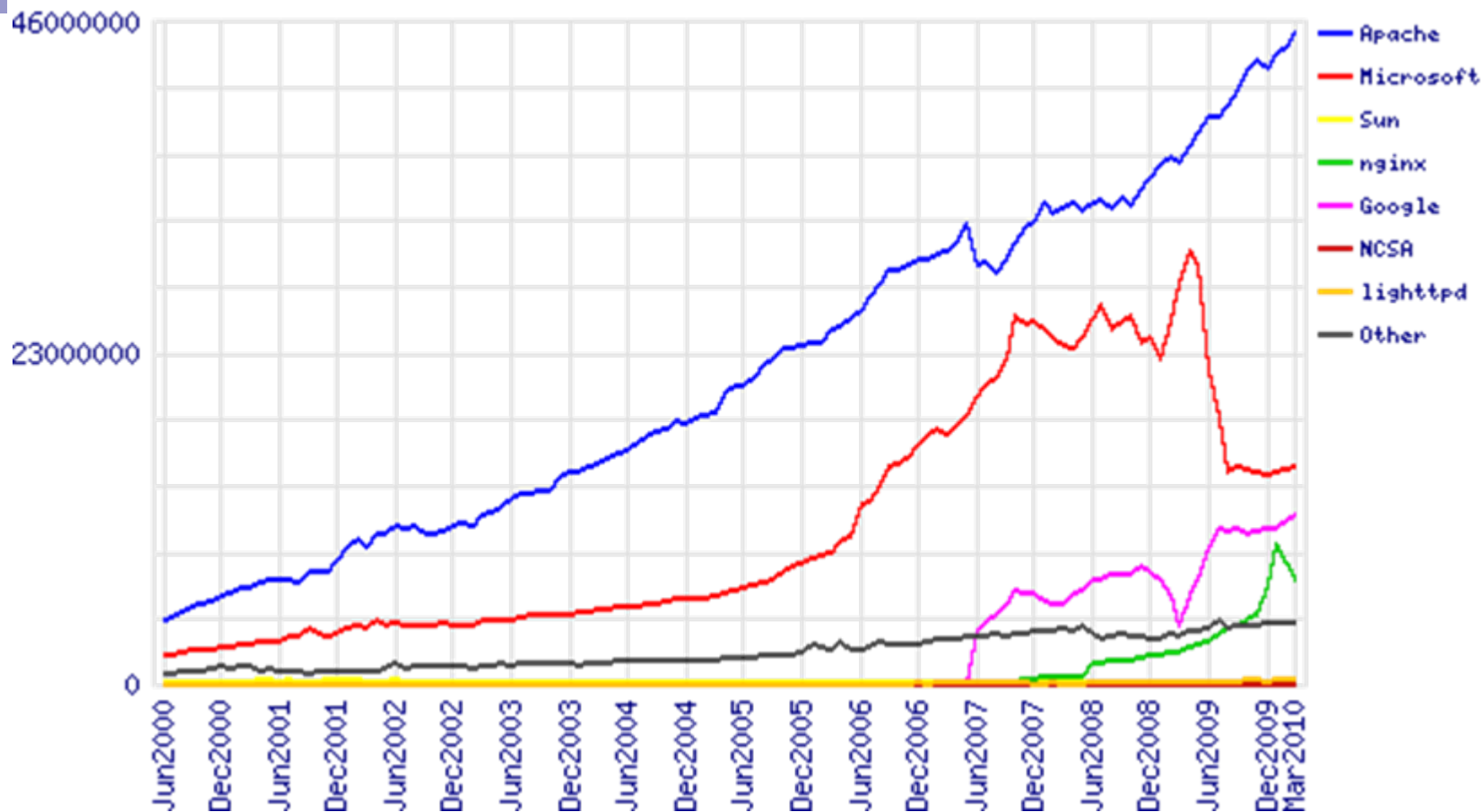


Obálka knihy: Larry Gonick, *Cartoon Guide to Statistics*

19.4.2010

© K 13135, CVUT FEL Praha

206,6 milionů URL / 72 milionů serverů

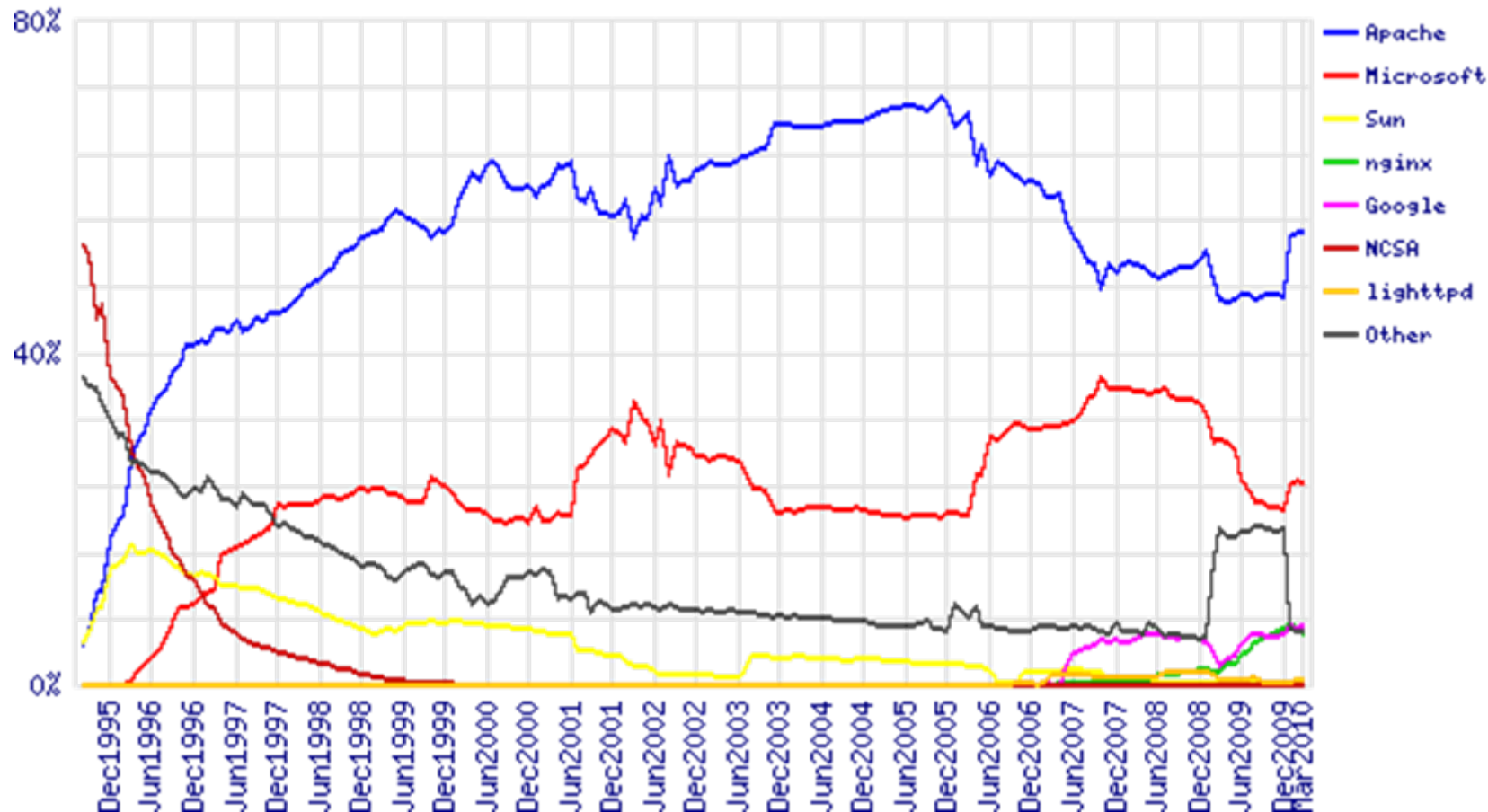


Zatím pořád vede LAMP

Linux, Apache, MySQL, Perl/Python/PHP

Zdroj: NetCrafts, [<http://news.netcraft.com/>]

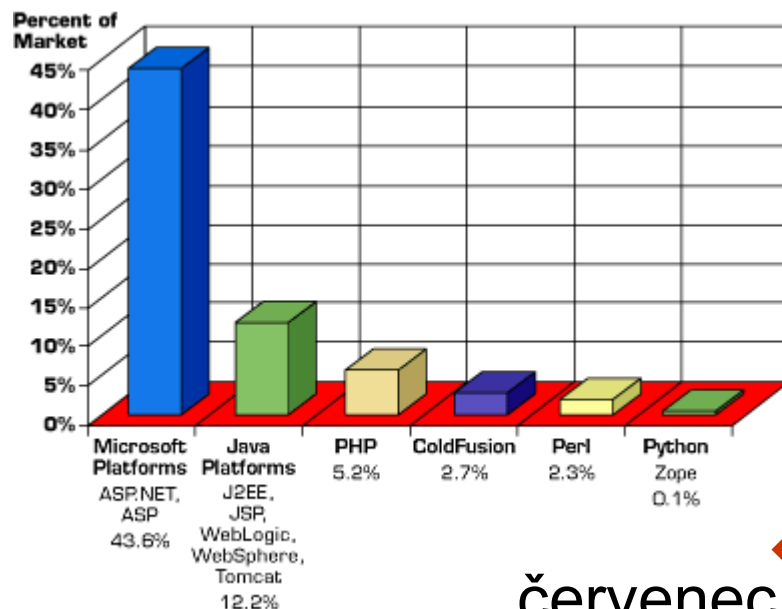
„Top Servers“



*Pokles v poslední době se objevil vlivem krize – rušení firem,
nebo Google web server s GWT zamávaly se statistikou ???
Bude to dočasné, trvalé ??? Kdo z pavouků tohle ví.... ☺*

Porovnání ASP.NET a velké firmy

→ ASP vítězí(lo) ve velkých složitých projektech

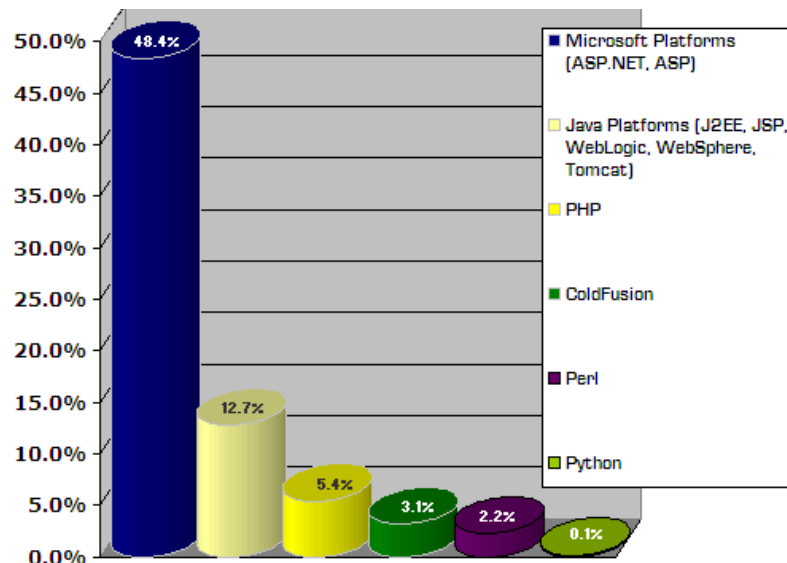
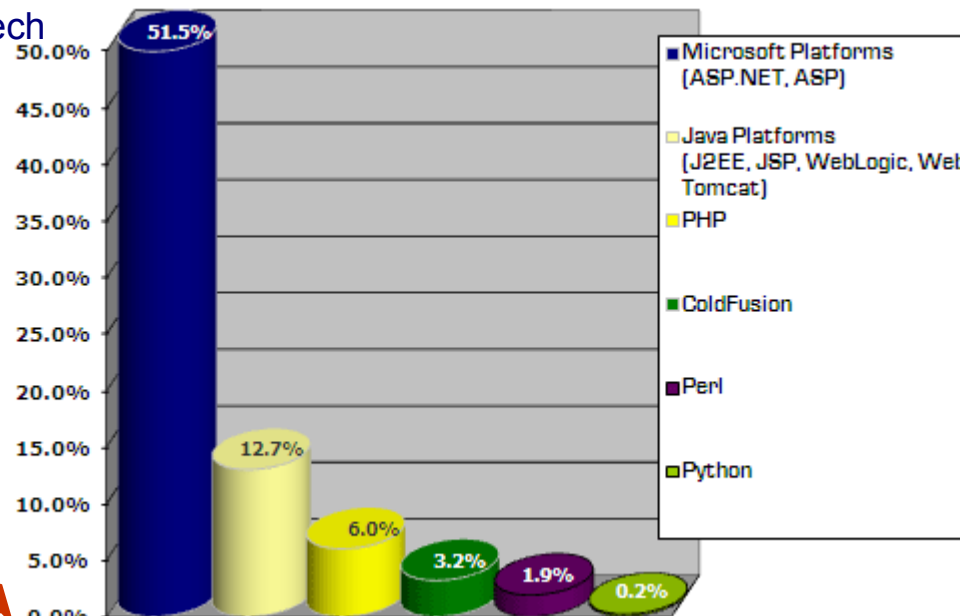


květen 2005

červenec 2007

srpen 2006

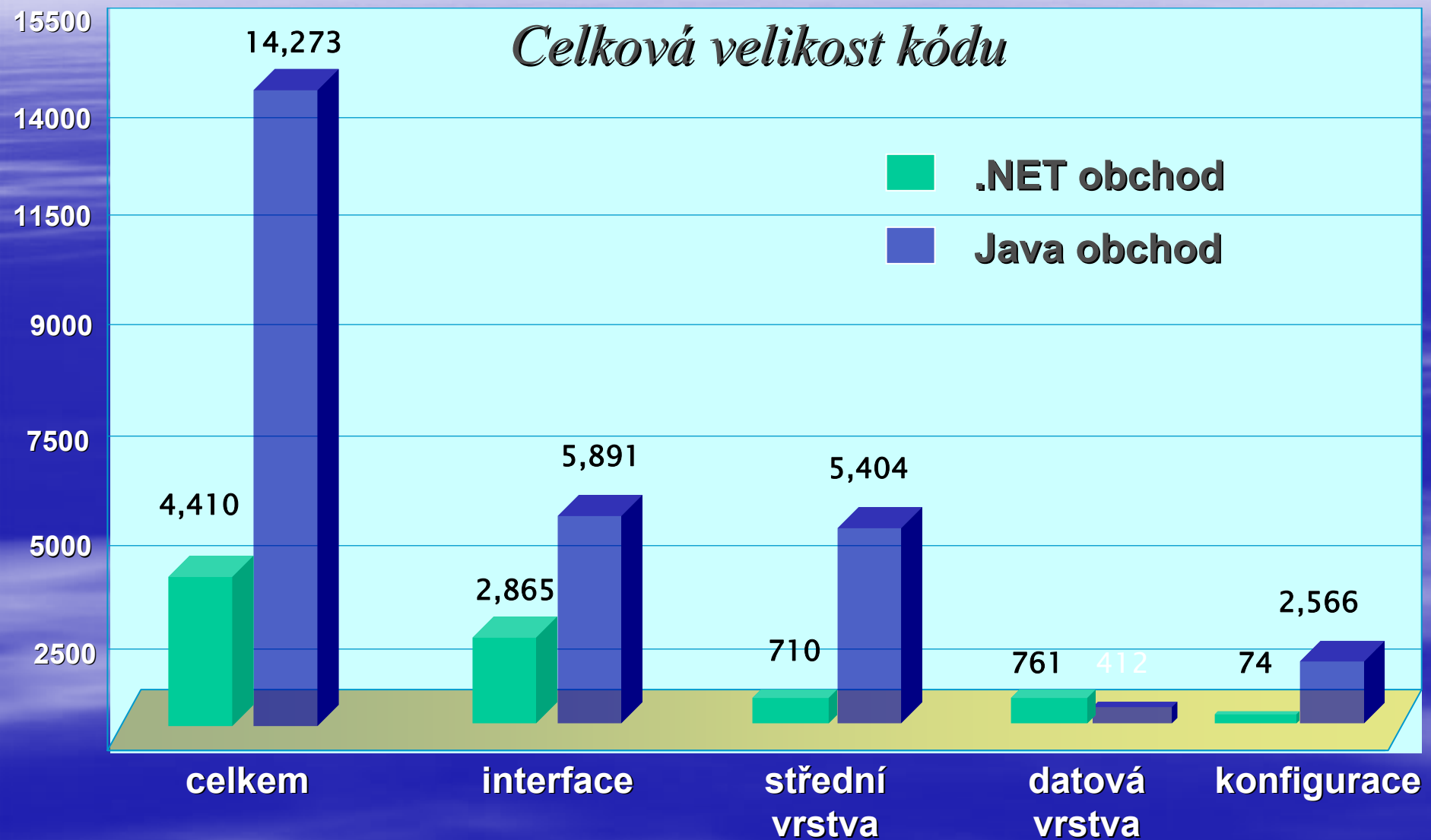
Zdroj: Port80 Surveys the Top 1000 Corporations' Application Servers and Scripting Platforms, [<http://www.port80software.com/surveys/top1000appservers/>]



Poslední data žel z roku 2007

Srovnání aplikace Pet-shop

převzato od [Bent Thomsen, Dept. of Computer Science Aalborg University]

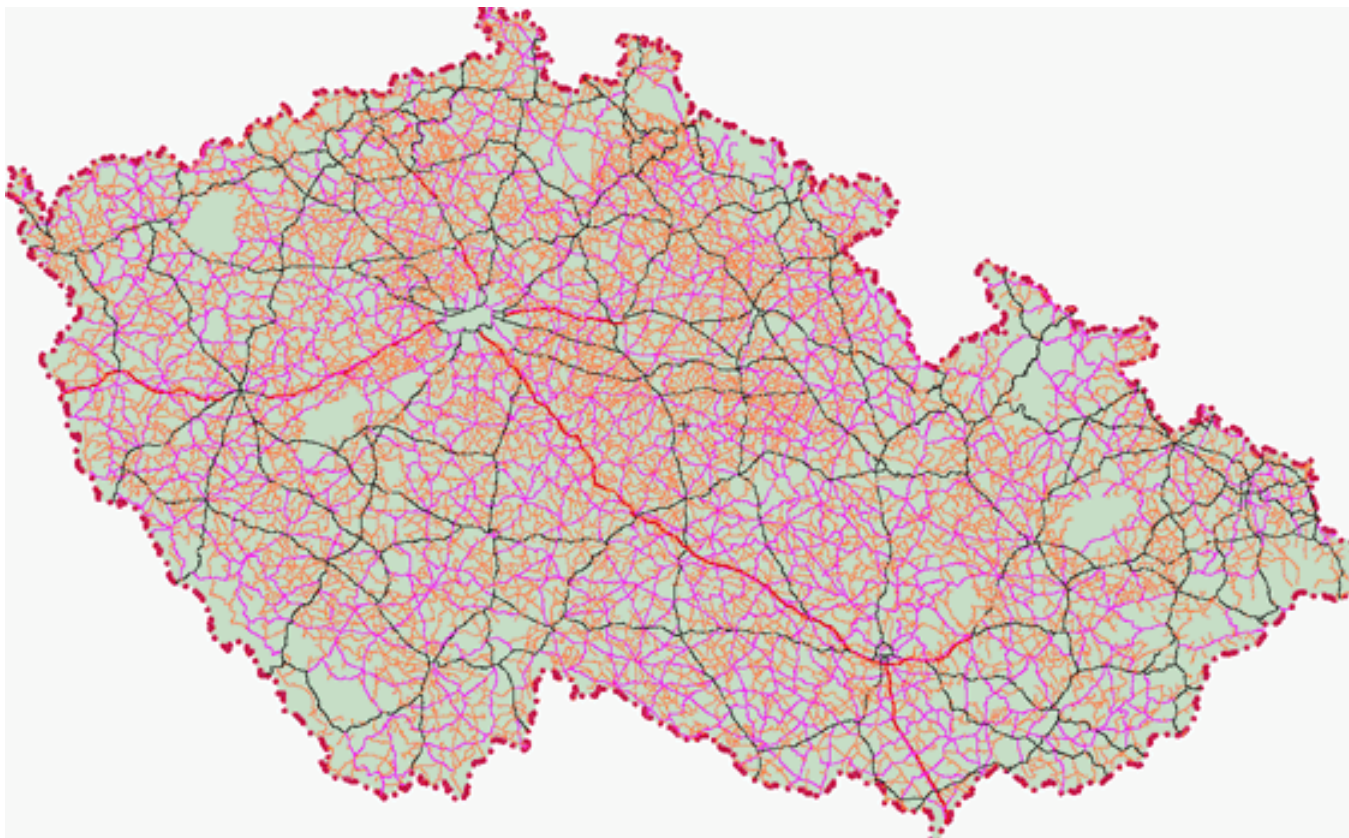


1. část - Internet

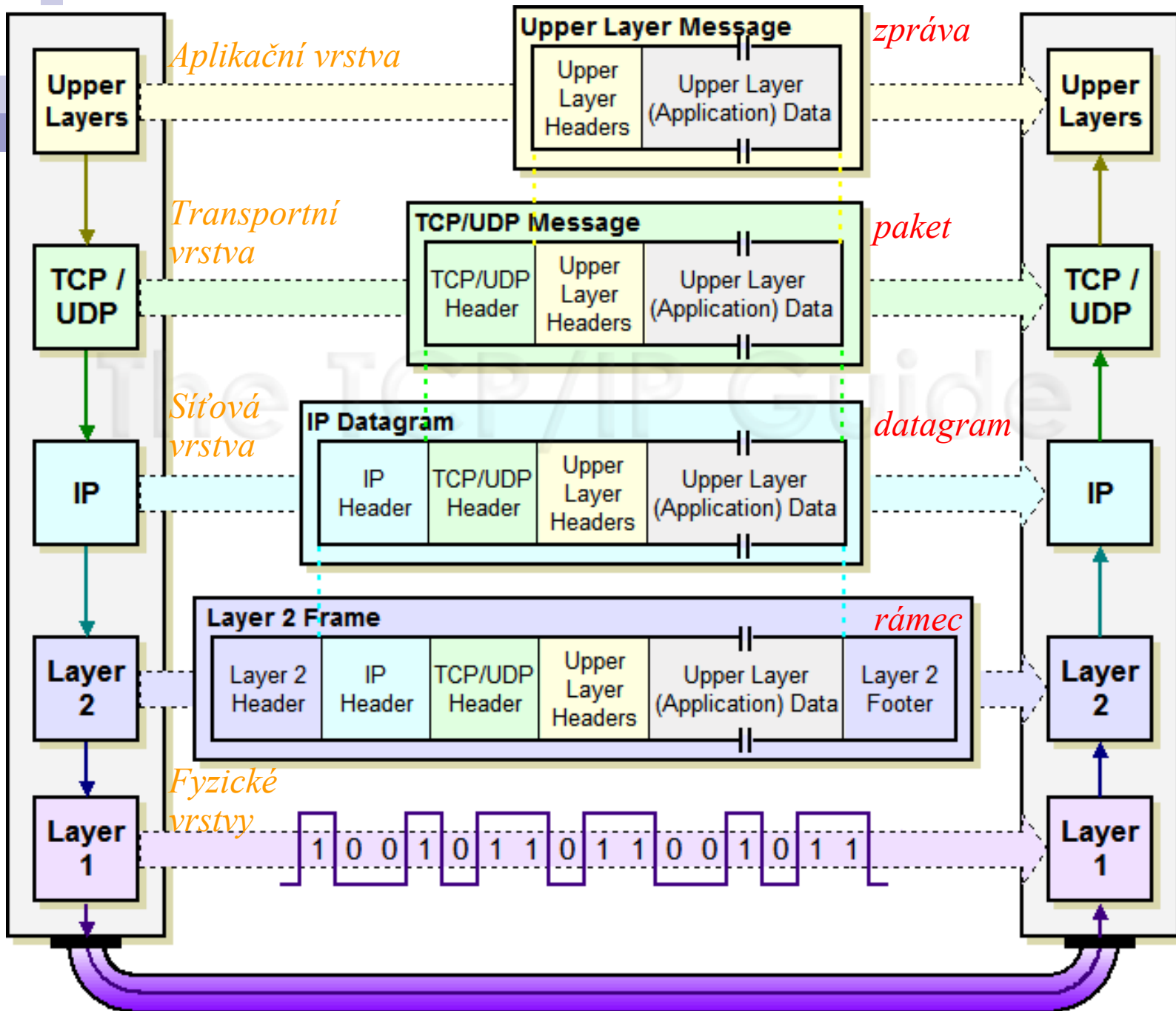
Stručné opakování základních pojmů



- Co má silniční síť společného s Internetem ?



[obrázek převzatý z webu http://www.mmc.cz/mmc_web_cz/Home.aspx]



Aplikační vrstva

- **FTP** (File Transfer Protocol) - *služba FTP*
- **SMTP** (Simple Mail Transfer Protocol) - *pro E-mail*
- **POP3** (Post Office Protocol) - *pro službu E-mail*
- **HTTP** (Hypertext Transfer Protocol) - *pro službu WWW*

Transportní vrstva

- **TCP** (Transmission Control Protocol) - *kladně potvrzovaný přenos dat*
- **UDP** (User Datagram Protocol) - *nepotvrzovaný přenos dat*

Síťová vrstva

- **IP** (Internet Protocol) - *přenos paketů, směrování přes IP adresy*
- **RARP** (Reverse Address Resolution Protocol) - *automatické přidělení IP adresy připojenému počítači*
- **ICMP** (Internet Control Message Protocol) - *přenos služeb. informací*

- IP - přenáší pakety podle cílové a zdrojové IP adresy
- TCP – segmenty přidávají informaci o zdrojových a cílových portech → rozlišení různé obsluhy protokolů

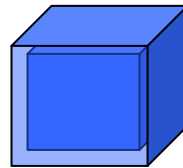
<input type="checkbox"/> <i>výchozí (default)</i>	<i>port</i>
■ HTTP	:80
■ HTTP se SSL (HTTPS)	:443
■ FTP	:21
■ SMTP	:25
■ POP	:110
■ SSH	:22

HTTP zpráva přenášená TCP/IP



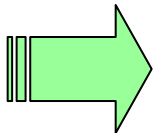
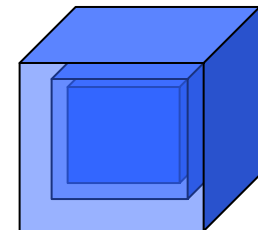
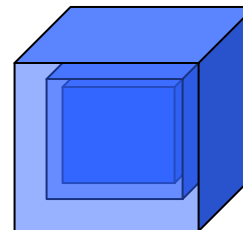
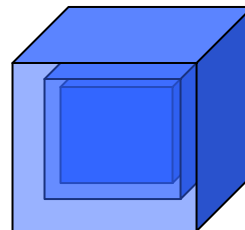
```
GET /index.php HTTP/1.1<CRLF>
Host: dce.felk.cvut.cz...
```

HTTP zpráva se rozdělí na části dostatečně malé pro TCP segmenty



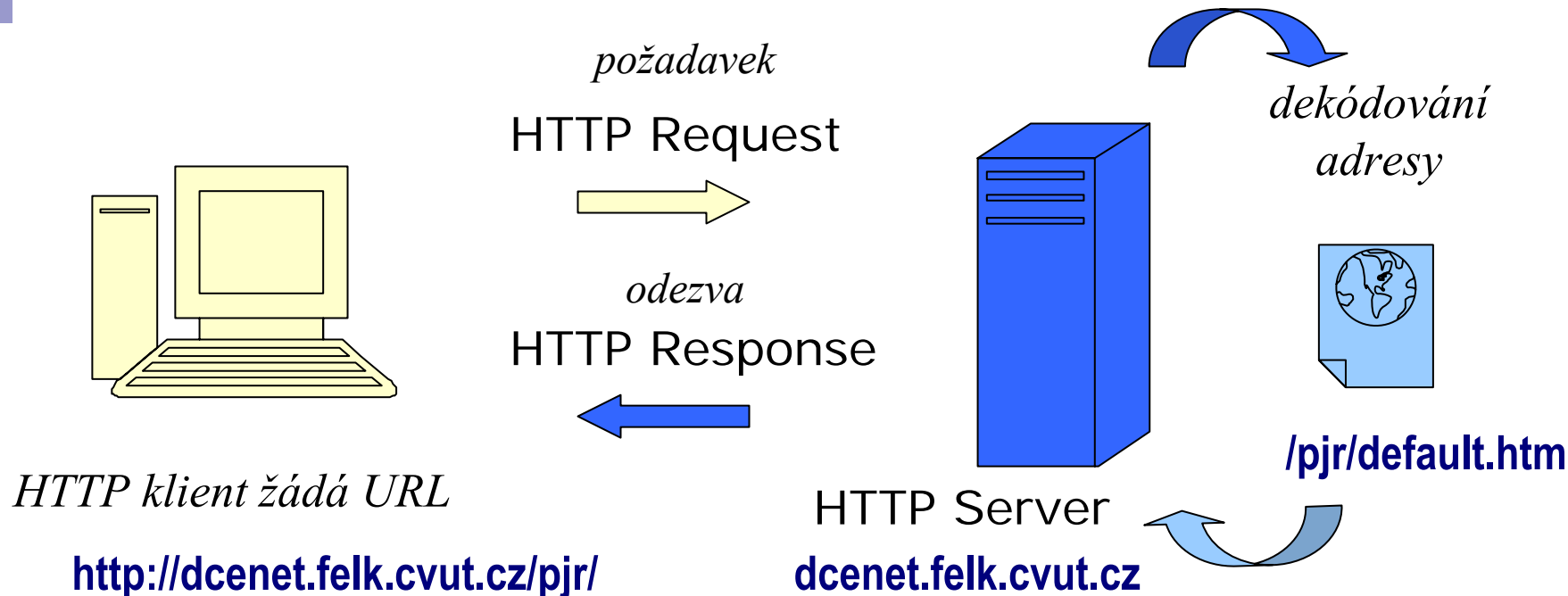
Části HTTP zprávy se vkládají do TCP segmentů

- *Maximum MTU (Transmission Unit) je 1500 bytů (+18) pro síť Ethernet (jiné sítě až 65280 – Fiber Channel)*
- *Rozhodující je min. MTU = 576 bytů.*
- *Po odečtení hlaviček IP a TCP (20+20 bytů) zůstane zaručených 536 bytů pro přenos HTTP.*
- *Přenos delších paketů je možný, ale nezaručený → "path MTU discovery".*



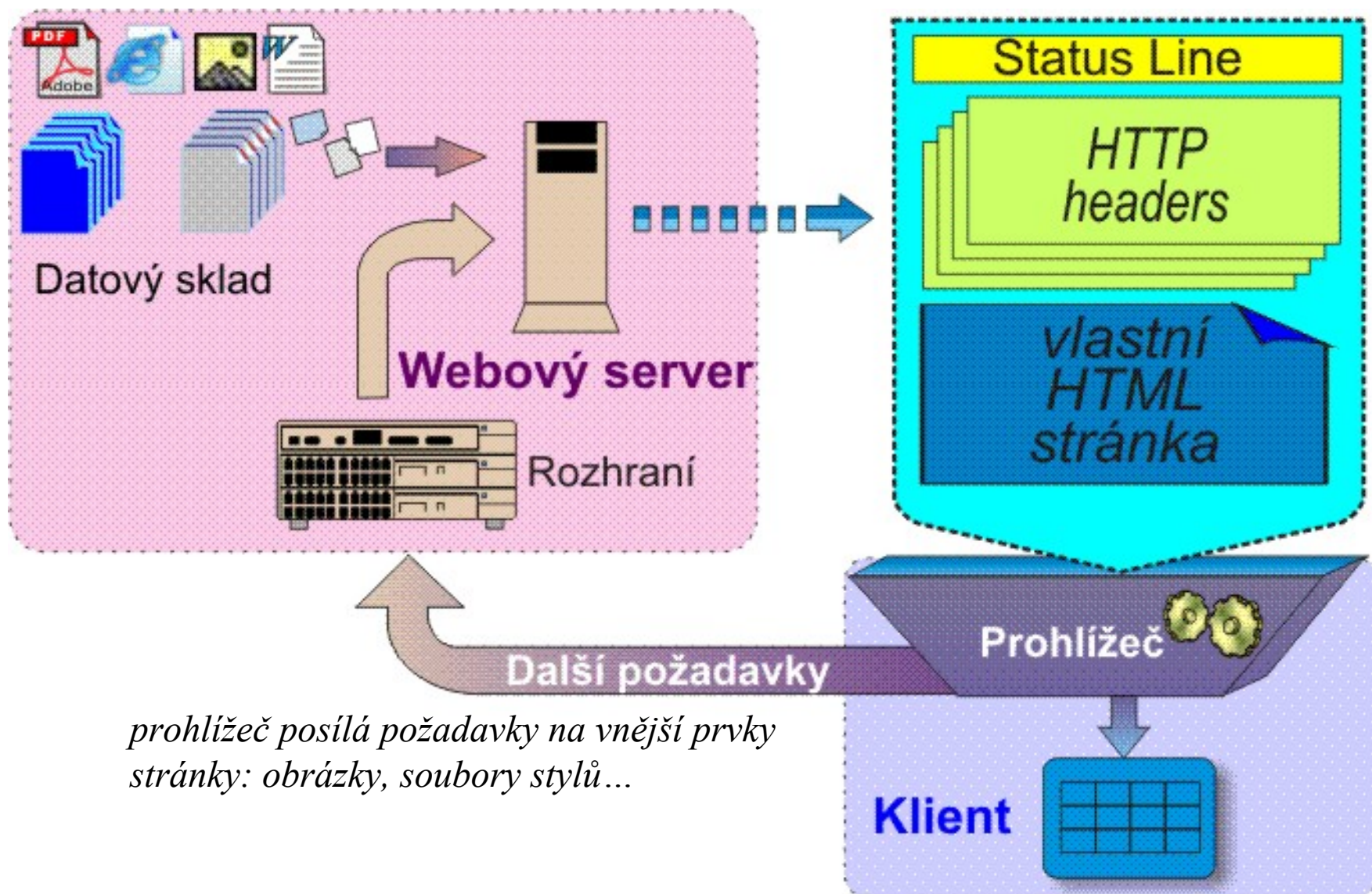
TCP segmenty putují uvnitř IP datagramů k příjemci zprávy

Základní HTTP Request/Response cyklus



- **URL** * *Uniform Resource Locator* - RFC 1738 (1994)
 - * *Universal Resource Identifier* - RFC 1630 (též 1994)
 - Request for Comments (RFC) - *internetový normovací mechanismus (původně ARPANET)*
 - blíže viz. síť Internet Engineering Task Force [<http://www.ietf.org/rfc.html>]

Struktura vlastní HTML stránky



Příklad HTTP "response"

HTTP/1.1 200 OK

*Verze HTTP
a status kód*

Content-Length: 366

Content-Type: text/html

Last-Modified: Mon, 02 Jan 2006 14:26:23 GMT

Accept-Ranges: bytes

Server: Microsoft-IIS/6.0

X-Powered-By: ASP.NET

Connection: close

*Okamžitá hodnota
"entity tag",
tj. verze objektu*

ETag: "9627d281a8fc61:1079"

MicrosoftOfficeWebServer: 5.0_Pub

Date: Mon, 02 Jan 2006 14:40:50 GMT

HTTP "headers"

HTTP head
(hlavička)

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

*nutné pro validní
stránku, ale
často opomíjené,*

<html lang="cs">

<head>

*Hlavička doplňující
HTTP headers*

<title>Moje stránka</title>

<meta http-equiv="content-type" content="text/html; charset=windows-1250">

<meta name="description" content="Testovací stránka" lang="cs">

</head>

<body>

-Vlastní definice obsahu stránky zobrazené v prohlížeči-

</body>

</html>

Zdrojový kód
stránky,
který zobrazí
prohlížeč

Hlavičku umí zobrazit například: <http://www.rexswain.com/httpview.html>

Některé metody pro "Request"

■ GET

- ☐ nejčastější způsob
- ☐ podporuje přenos řetězců argumentů, např.
<http://dce.felk.cvut.cz/pub/index.php?language=2&id=36&category=browse>

■ POST

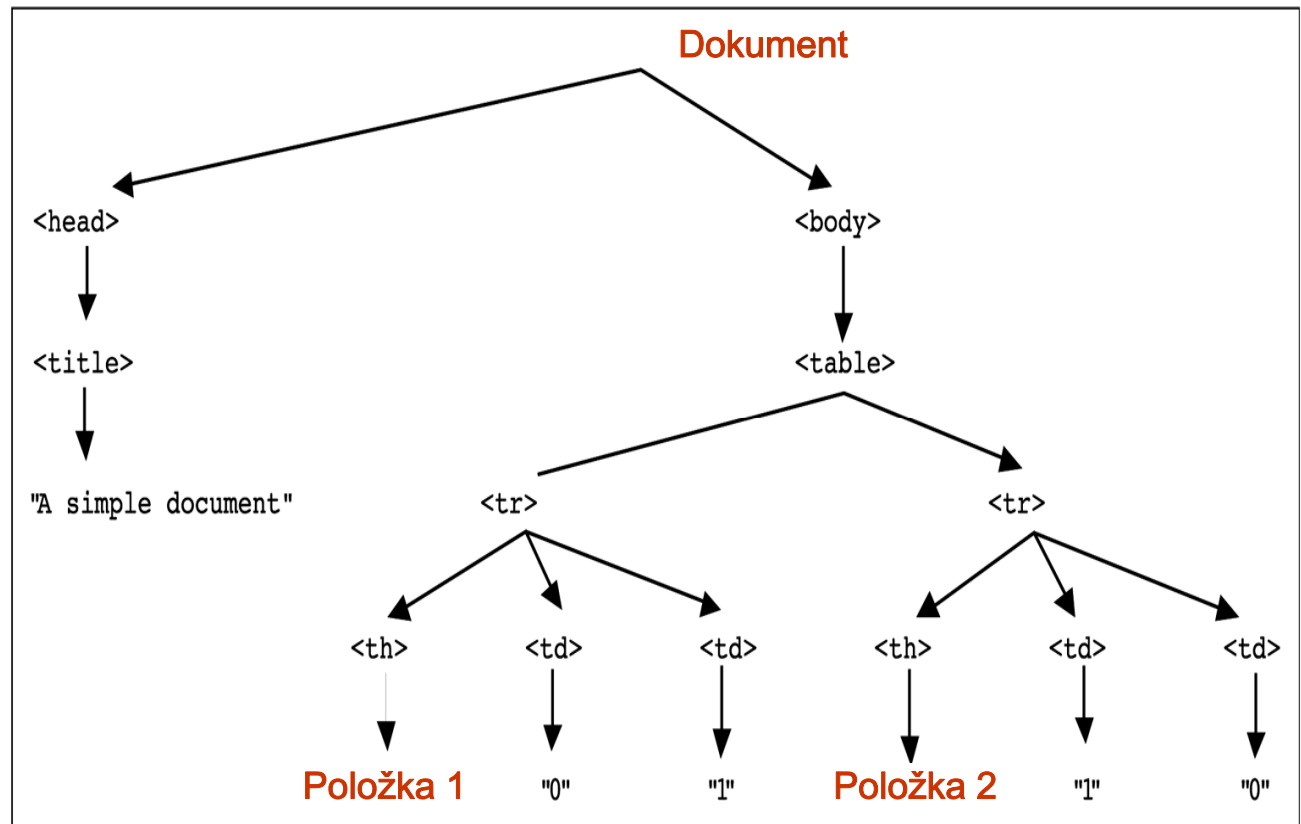
- ☐ přenáší data v hlavičkách HTTP protokolu
- ☐ může přenést větší argumenty než GET
- ☐ argumenty nejsou zobrazené v URL, tj. nelze na ně odkázat

■ HEAD

- ☐ čteme jenom hlavičky, nikoliv entitu samotnou
- ☐ vhodné se pro analýzu a diagnostiku

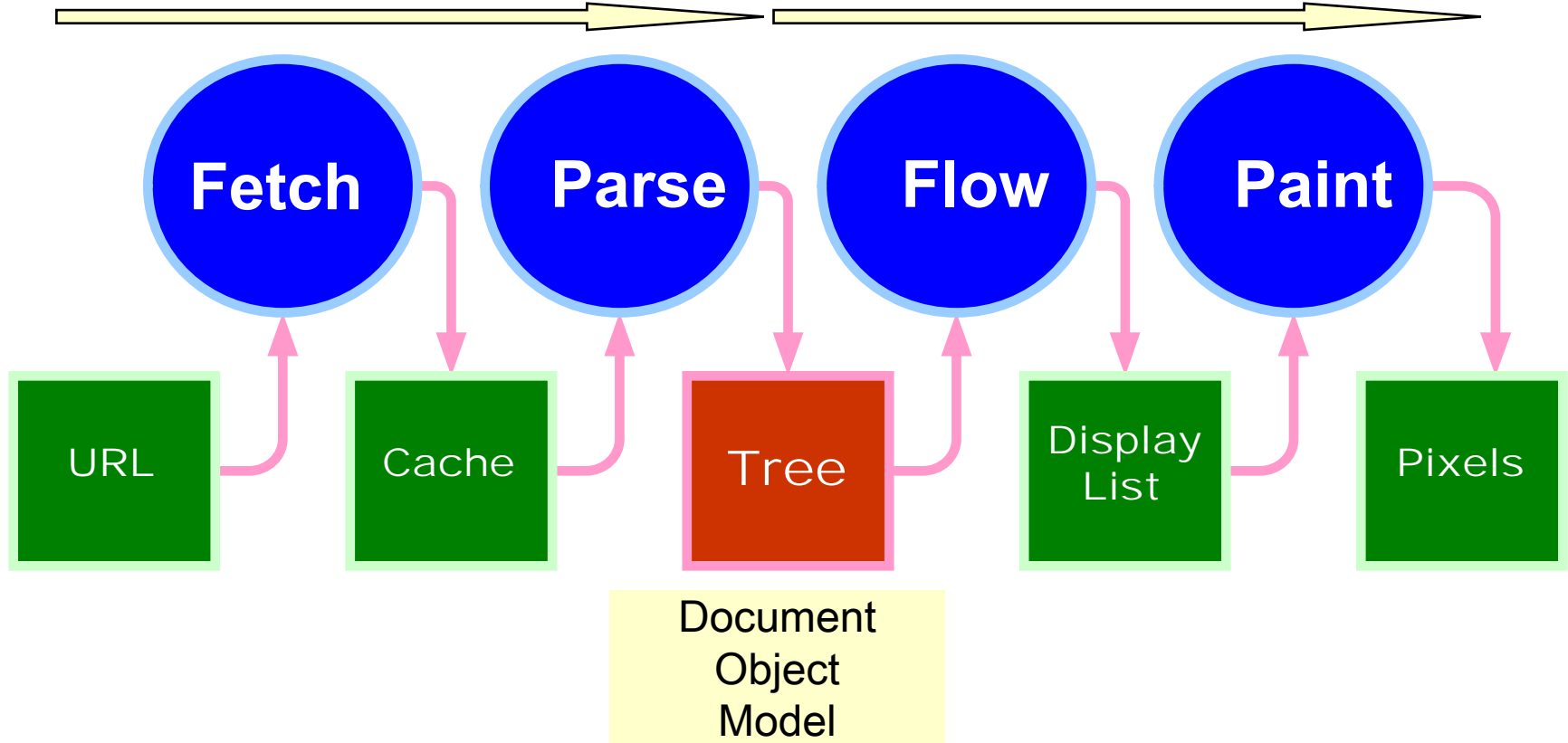
■ HTML stránky mají stromovou strukturu

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Dokument</title>
</head>
<body>
  <table>
    <tr>
      <th>Položka 1</th>
      <td>0</td>
      <td>1</td>
    </tr>
    <tr>
      <th>Položka 2</th>
      <td>1</td>
      <td>0</td>
    </tr>
  </table>
</body>
</html>
```

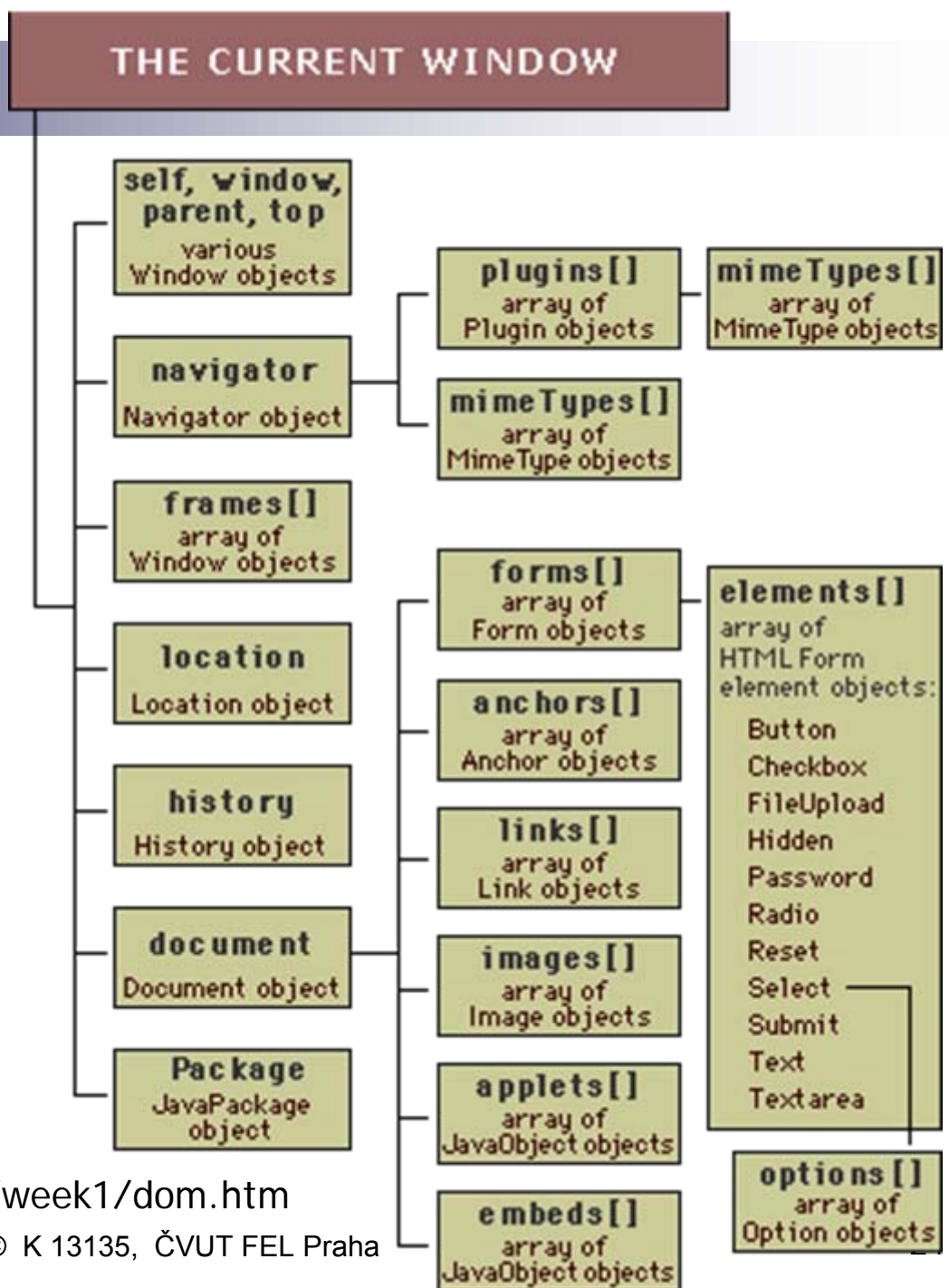


Čtení a předzpracování stránky

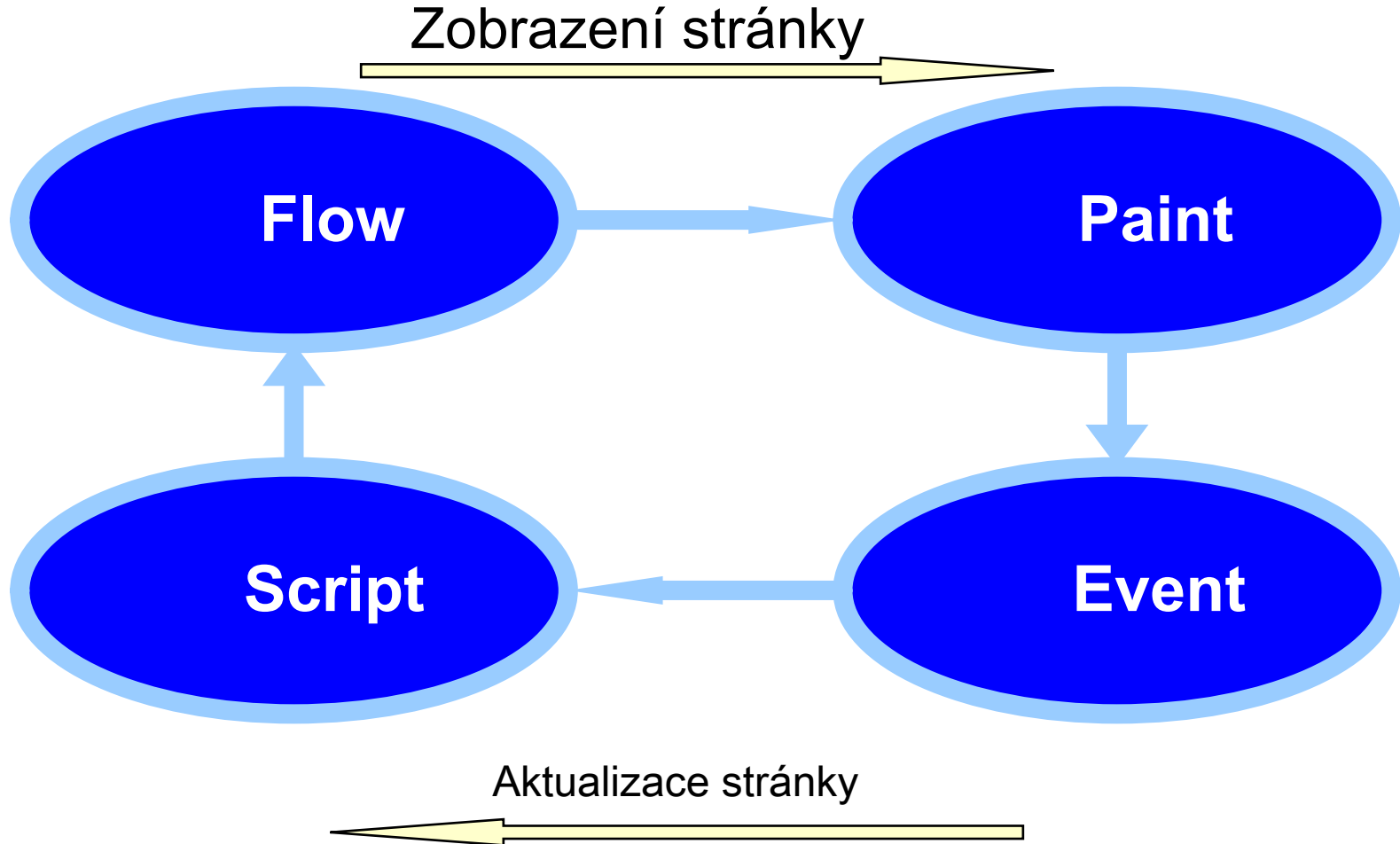
Zobrazení stránky



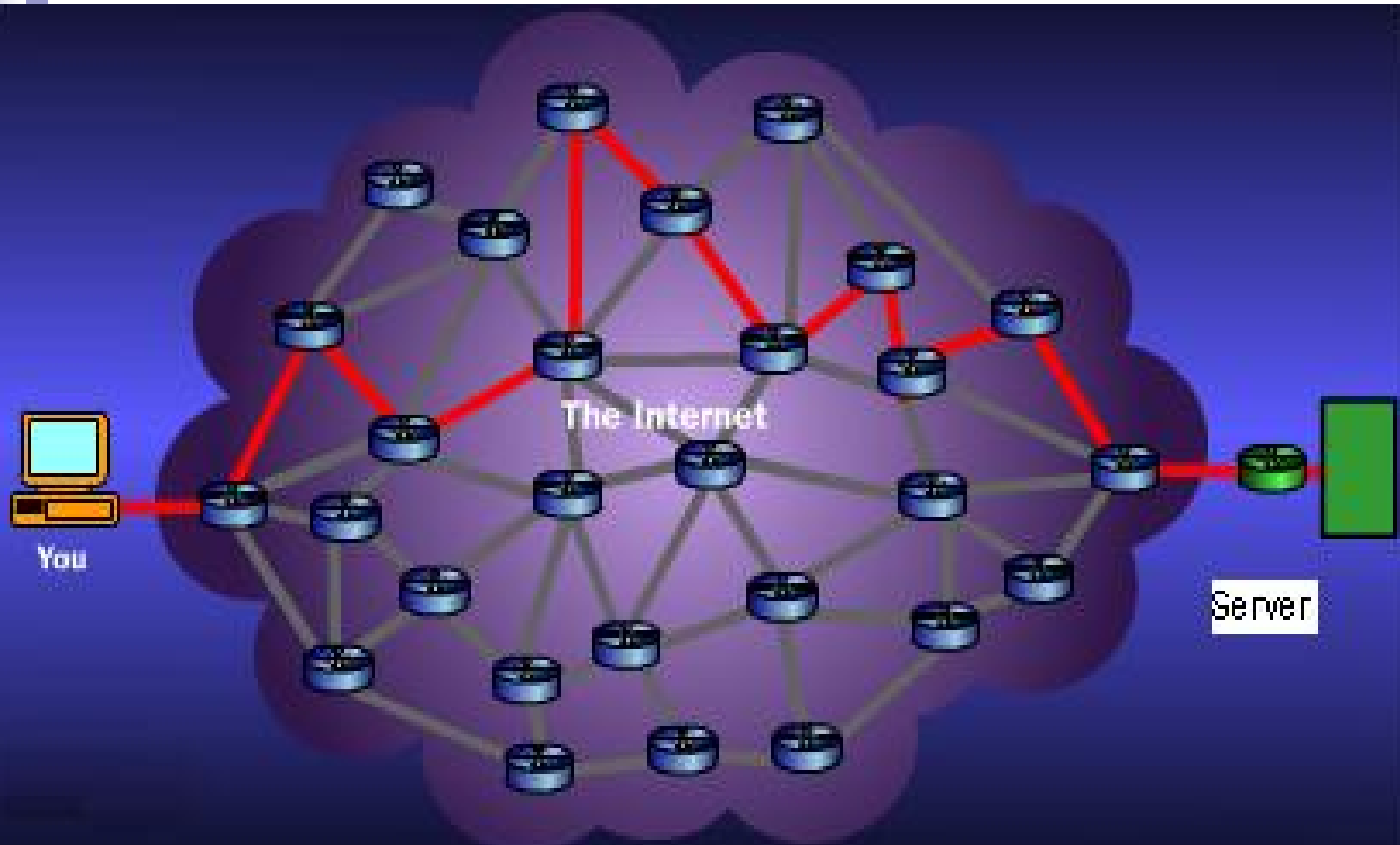
The DOM hierarchy



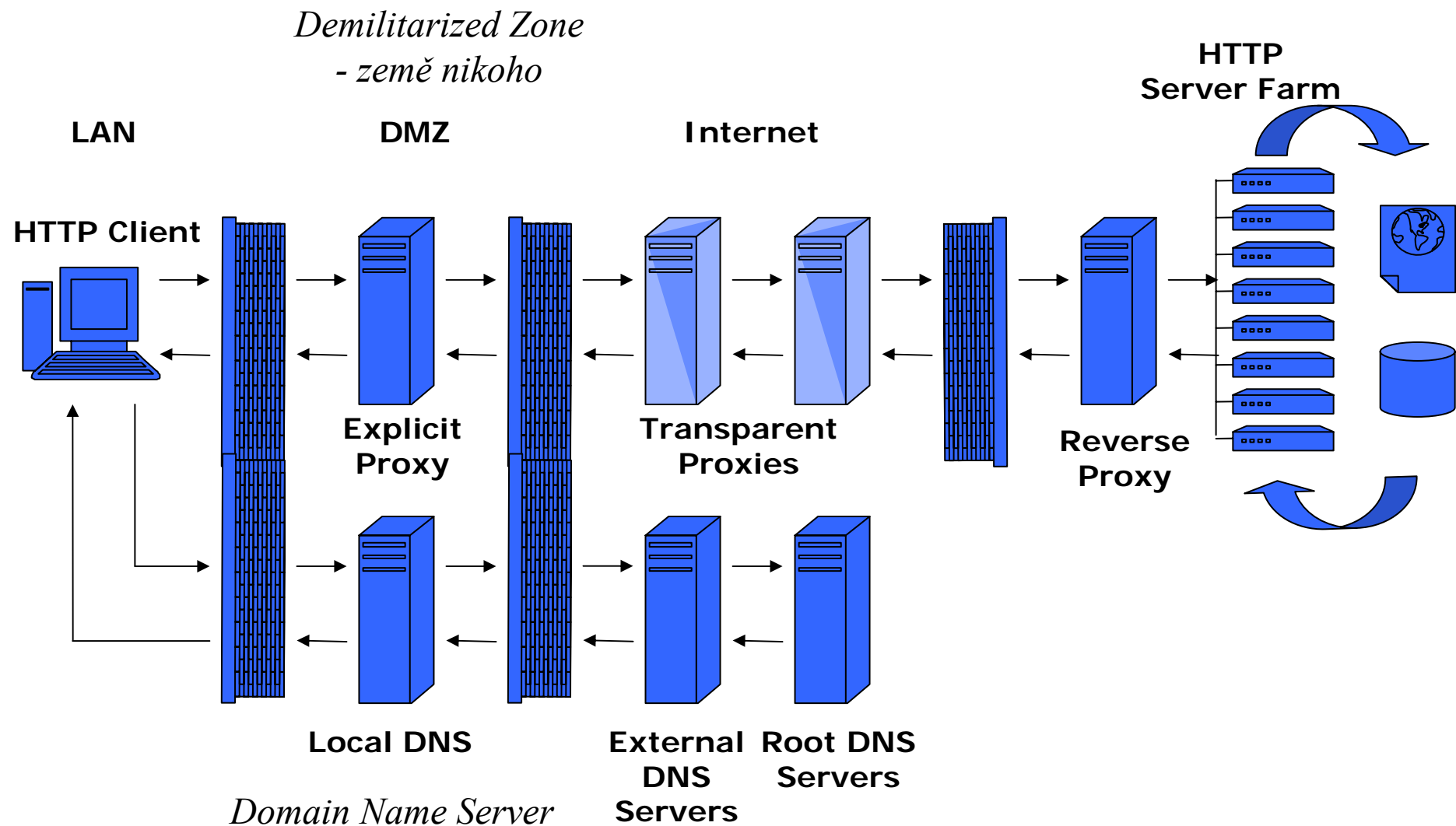
Source: <http://sislands.com/coin70/week1/dom.htm>



Cesta server-kleint



Skutečný HTTP Request/Response Chain



[Obrázek převzat z: Murray M.,: AMCIS Workshop on ASP.NET and Web Services, New York City, NY August 5-8, 2004]

- Proxy – *server mezi klientem a webovým serverem plní dvě hlavní role*
 - *vyrovnávací (cache) paměť pro redukci zatížení sítě*
 - *bezpečnostní úloha, brání přímému přístupu na síť.*
- Explicitní proxy (též web proxy) – *prezentuje všechny požadavky klientů sítě LAN pod jednou IP adresou tím, že modifikuje hlavičky posílaných paketů.*
- Transparentní proxy – *jen odděluje sítě, neobsahuje žádná lokální pravidla, a proto nemodifikuje hlavičky zprávy.*
- Reverse proxy – *bud' obslouží požadavek sama ze své vnitřní paměti, nebo pošle požadavek (bud' celý nebo jeho část) na příslušný web server.*

<head>

... *<!-- no cache headers – pro zaručení načtení stránky ze serveru -->*

<meta http-equiv="Pragma" content="no-cache">

<meta http-equiv="no-cache">

<meta http-equiv="Expires" content="-1">

<meta http-equiv="Cache-Control" content="no-cache">

<!-- end no cache headers -->

</head>

- <http://validator.w3.org/>
- *test validity stránky – doporučeno!*
- <http://www.rexswain.com/httpview.html>
- *zobrazení hlavičky HTTP*
- <http://www.tcpipguide.com/free/>
- *TCP/IP průvodce*

ASP.NET

Všechny příklady v této části se opírají o Visual Studio 2008

Zde částečně použité snímky

[JS] Stoklasa J.: *ASP.NET*

programování webových aplikací, Microsoft 2004

[SSW] *ASP.NET*. University of Linz, Institute for System Software,
2004, published under the Microsoft Curriculum License
příklady však upravené pro Visual Studio 2008

- Starší technologie skriptů na straně serveru
 - ASP, PHP, ColdFusion, ...
- ... používají lineární programovací model
 - html a výkonný kód vnořený do speciálních závorek
 - `<% %>`
 - `<? ?>`
- Důsledek:
 - špatná udržitelnost kódu
 - nejasná struktura aplikace
- ASP.NET umí
 1. oddělit kód od stránky
 2. skládat stránky z objektů.
 3. zpracovat i starší ASP stránky → lze je přímo použít

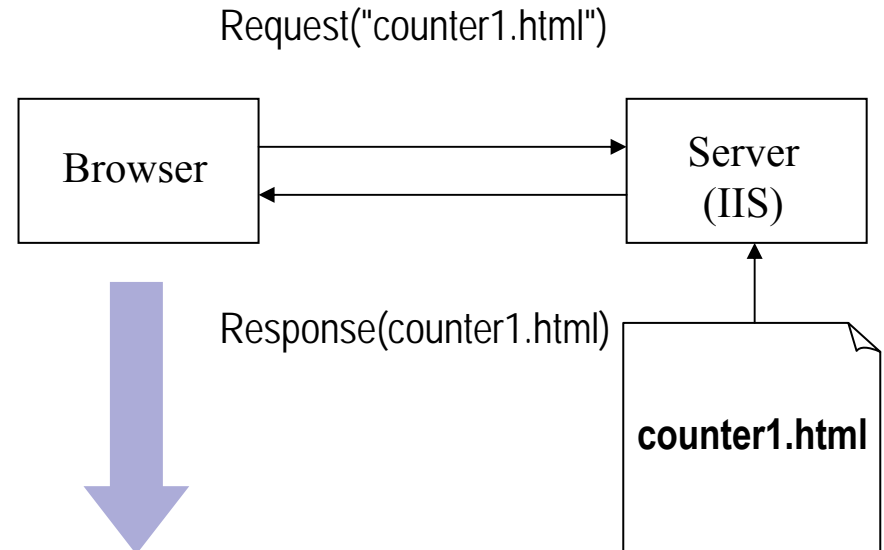
!S server lze v případě potřeby doplnit i o PHP modul, takže můžeme využít i dříve napsané stránky

Příklad statické stránky

Čisté HTML

```
<html>
  <head>
    <title>Simple HTML page</title>
  </head>
  <body>
    <h1>Welcome</h1>
    You are visitor number 1!
  </body>
</html>
```

counter.html



Demo

Visual Studio 2008 program

Příklad dynamické ASPX stránky

Counter1.aspx

```
<%@ Page Language="C#" %>
```

```
<%@ Import Namespace="System.IO" %>
```

```
<% Response.Cache.SetCacheability(HttpCacheability.NoCache); %>
```

```
<html>
```

```
<head> <title>Page counter</title> </head>
```

```
<body> <h1>Welcome</h1> You are visitor number <%
```

```
FileStream s = new FileStream(Server.MapPath("./data/counter.dat"),  
                               FileMode.OpenOrCreate);
```

```
int n;
```

```
try { BinaryReader r = new BinaryReader(s); n = r.ReadInt32();  
    } catch { n = 0; } // if the file is empty
```

```
n++; s.Seek(0, SeekOrigin.Begin);
```

```
BinaryWriter w = new BinaryWriter(s);
```

```
w.Write(n); s.Close();
```

```
Response.Write(n);
```

```
%>!
```

```
</body>
```

```
</html>
```



Programming Basics

Import Directive

- ◆ Adds code namespace reference to page

```
<%@ Import Namespace="System.Data" %>  
<%@ Import Namespace="System.Net" %>  
<%@ Import Namespace="System.IO" %>
```

types

- Equivalent to the C# using directive

HTML kód vrácený serverem

```
HTTP/1.1·200·OK
Server:·Microsoft-IIS/5.0
Date:·Tue,·03·Jan·2006·11:17:14·GMT
X-Powered-By:·ASP.NET
Connection:·close
X-AspNet-Version:·2.0.50727
Cache-Control:·no-cache
Pragma:·no-cache
Expires:·-1
Content-Type:·text/html;·charset=utf-8
Content-Length:·137
```

```
<html>
(HT)<head>·<title>Page·counter</title>·
</head>
(HT)<body>·<h1>Welcome·2</h1>·
····You·are·visitor·number·6·!·
</body>·
</html>
```

důsledek řádky

```
<%
Response.Cache.SetCacheability
(HttpCacheability.NoCache);
%>
```

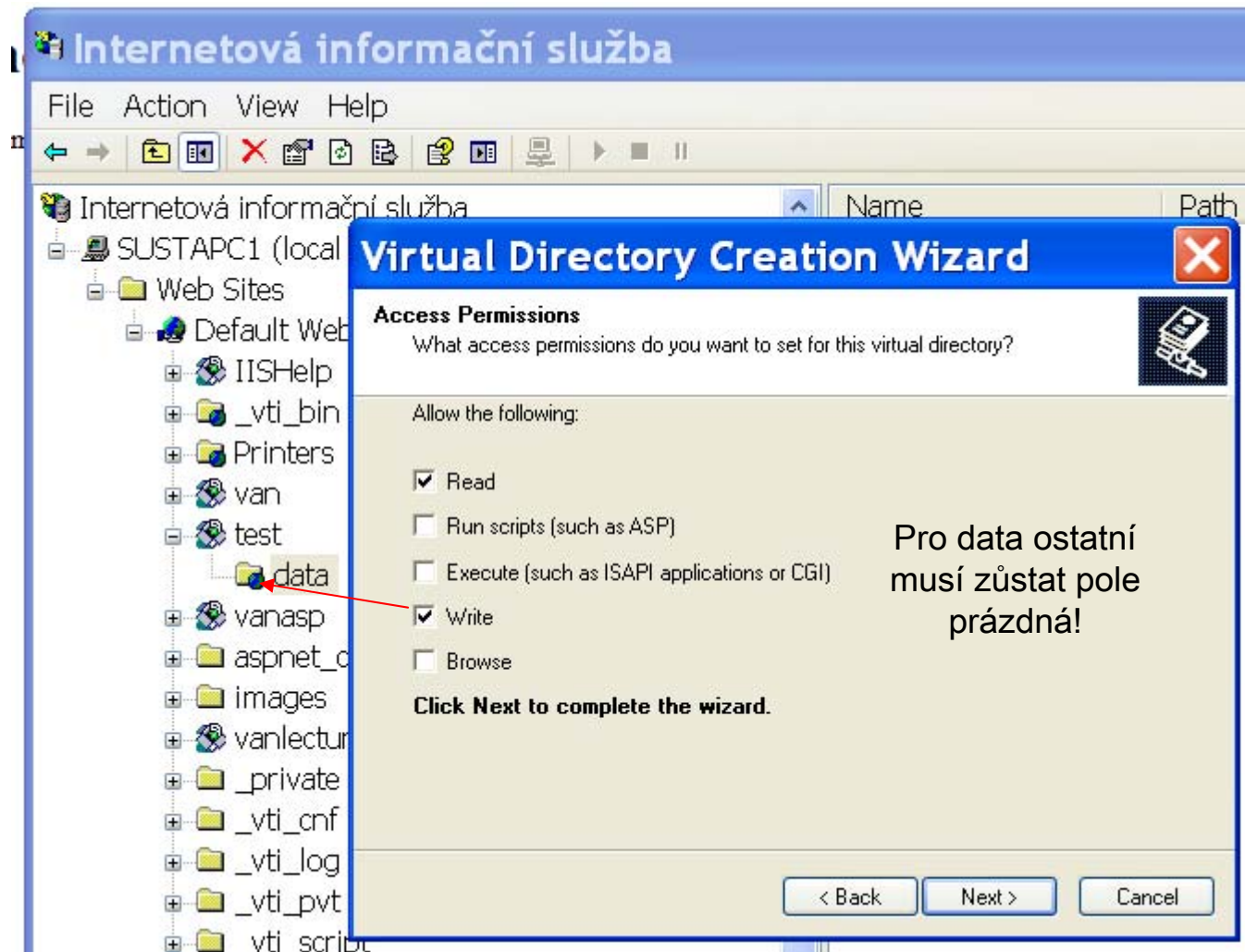
- prostý HTML kód
 - zobrazí ho jakýkoliv prohlížeč

Server obsahuje četná zabezpečení, a proto nutno

1. konfigurovat virtuální adresář
2. konfigurovat adresář pro soubor čítače ./data
3. ... *mnohé další úkony*:
 - ☐ ve Windows XP musíte povolit správu adresářů, pokud již není nastaveno. Výchozí stav je totiž zjednodušená správa složek.
 - ☐ nastavit přístup na SQL server
 - ☐ práva uživatelů...

A nastavení se liší podle verze IIS...

Data jako IIS virtuální adresář



Syntaxe bloků kódu

```
<%@ Page Language="C#" %>
```

```
...
```

```
<% for (int i=0; i < 8; i++) { %>
```

```
  <FONT size="<%=i%>">
```

```
    Hello World!
```

```
  </FONT>
```

```
  <BR>
```

```
<% } %>
```

Inline příkazový fragment

Inline výraz: zkratka pro
Response.Write(i);

Syntaxe bloků <script>

```
<script runat="server" [language="..."] >  
..... code .....  
</script>
```

- ◆ Proměnné a metody se přeloží a zařadí do dynamicky vytvořené třídy Page

```
<script runat="server" [language="..."]  
[src="external filename"]  
><000000
```

Kód jako skript

Counter2.aspx

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.IO" %>
<% Response.Cache.SetCacheability(HttpCacheability.NoCache); %>
<script Language="C#" Runat="Server">
    int CounterValue() {
        FileStream s = new FileStream(Server.MapPath("./data/counter.dat"),
            FileMode.OpenOrCreate);

        ...
        n = r.ReadInt32(); n++;
        ...
        return n;
    }
</script>
<html> <head> <title>Page counter</title>
    </head>
    <body> <h1>Welcome 2</h1>
    You are visitor number <%=CounterValue()%> !
    </body>
</html>
```



↑
zkratka pro
Response.Write(CounterValue());

Příklad stránky s kódem vně (Code Behind)

```
<%@ Page Language="C#" CodeFile="counter3.aspx.cs" Inherits="counter3" %>
<% Response.Cache.SetCacheability(HttpCacheability.NoCache); %>
<html>
  <head> <title>Page counter</title> </head>
  <body> <h1>Welcome 3</h1>
  You are visitor number <%=CounterValue() %> !
</body>
</html>
```

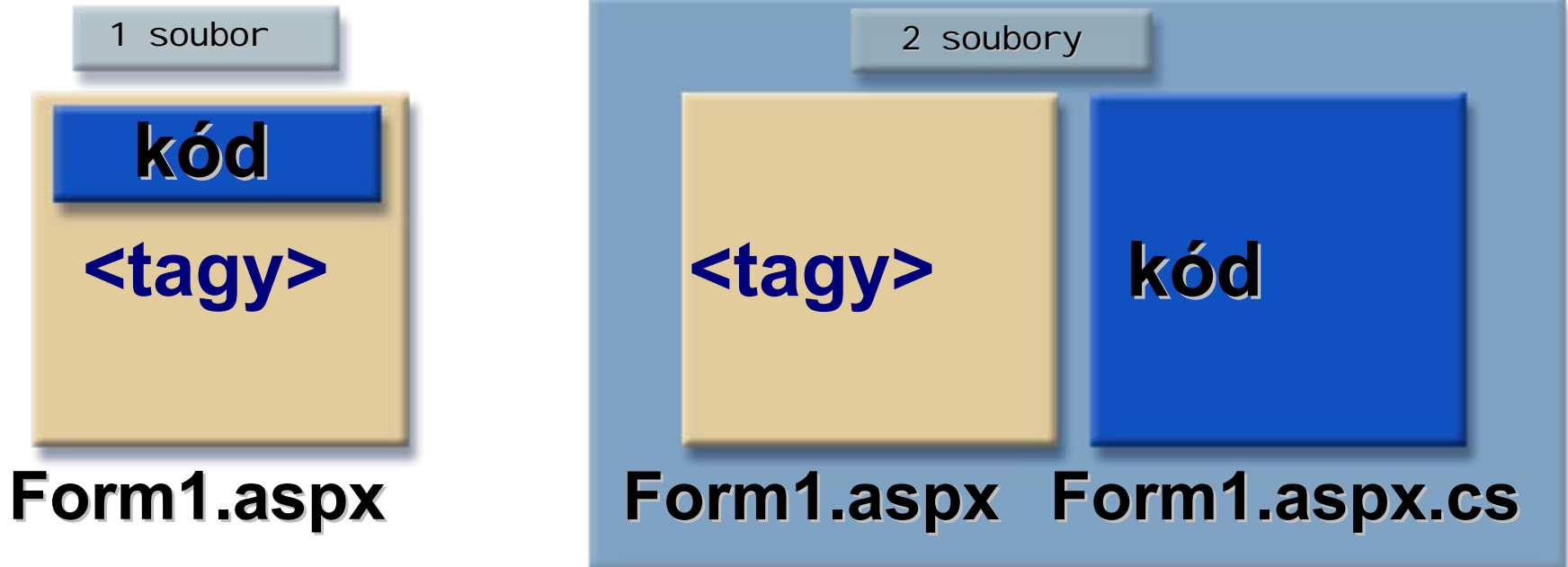
counter3.aspx

```
using System.IO;
public partial class counter3 : System.Web.UI.Page
{ public int CounterValue()
{
  FileStream s = new FileStream(Server.MapPath("./data/counter.dat"),
                               FileMode.OpenOrCreate);

  ...
  n++;
  ...
  return n; }
}
```

counter3.aspx.cs

Vzhled a kód se oddělil



■ Praktická poznámka k deklaraci

- `<%@Page CodeFile=... Inherits...>` *preferováno ve VS2005.NET*
- `<%@Page CodeBehind=... Inherits=...>` *- v dřívějších verzích VS*
- `<%@Page Src=...>` *- v dřívějších verzích VS.*
VS2005 používá pro kód s BuildProvider → tvorba zdrojového kódu

Příklad counter4

ASP_code Library

ASP.Code in Library

```
<%@ Page Language="C#" CodeFile="counter4.aspx.cs" Inherits="counter4" %>
```

```
<% Response.Cache.SetCacheability(HttpCacheability.NoCache); %>
```

```
<html>
```

```
  <head> <title>Page counter</title> </head>
```

```
  <body> <h1>Welcome 4</h1>
```

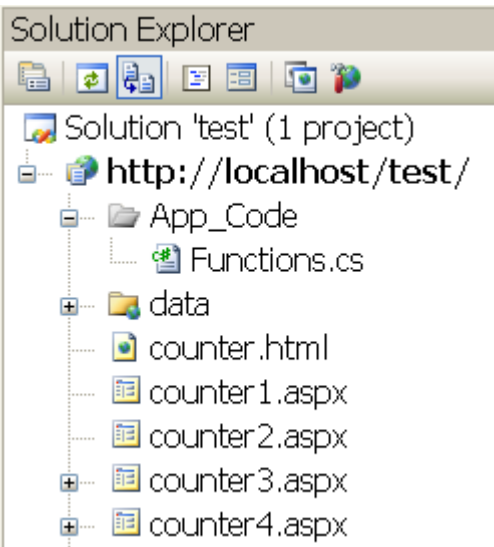
```
    You are visitor number <%=Functions.CounterValue(this)%> !
```

```
  </body>
```

```
</html>
```

```
public class Functions
{
    public static int CounterValue(
        System.Web.UI.Page page)
    {
        FileStream s = new FileStream(
            page.Server.MapPath("./data/counter.dat"),
            FileMode.OpenOrCreate);

        /* ... */
        return n;
    }
}
```

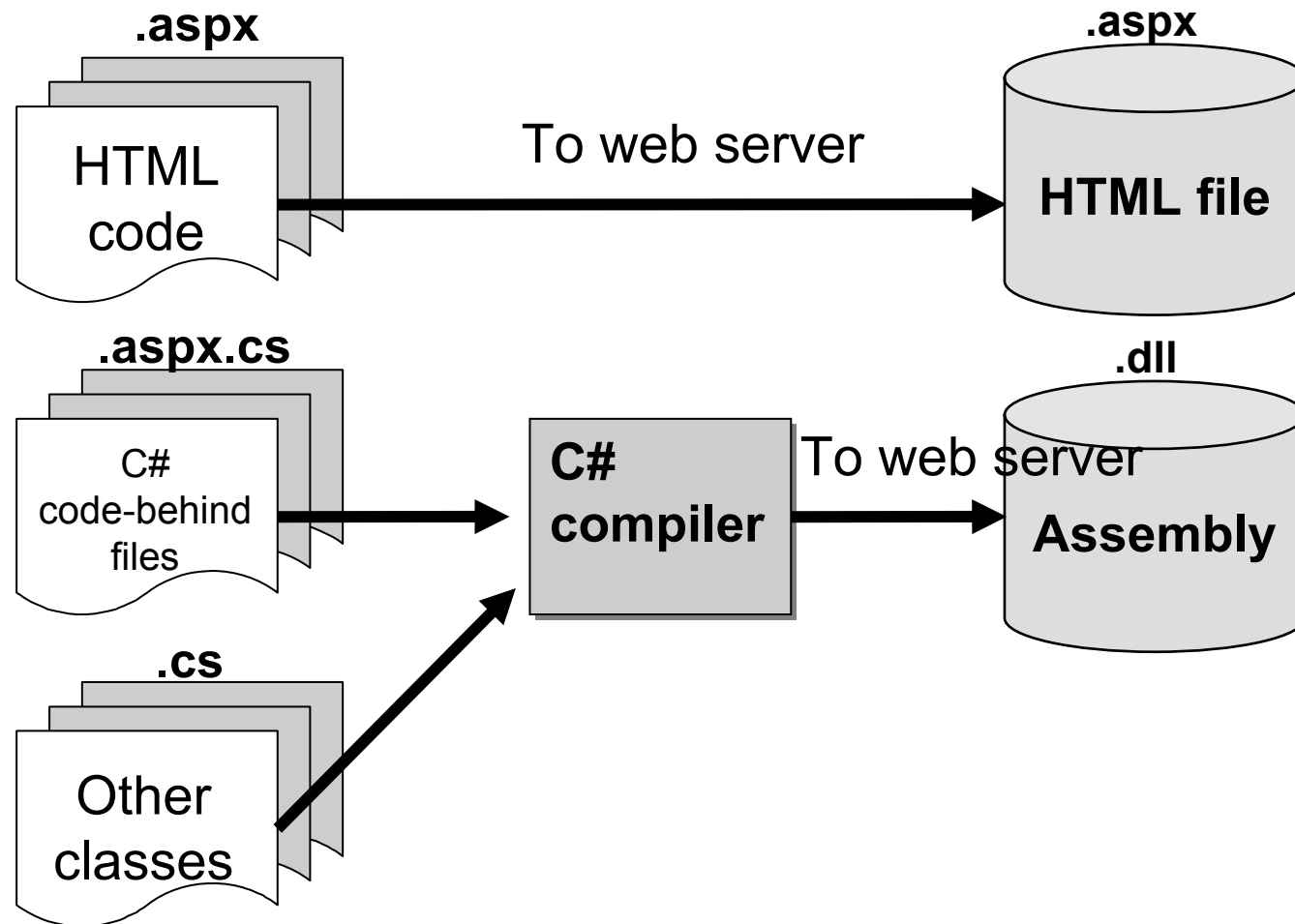




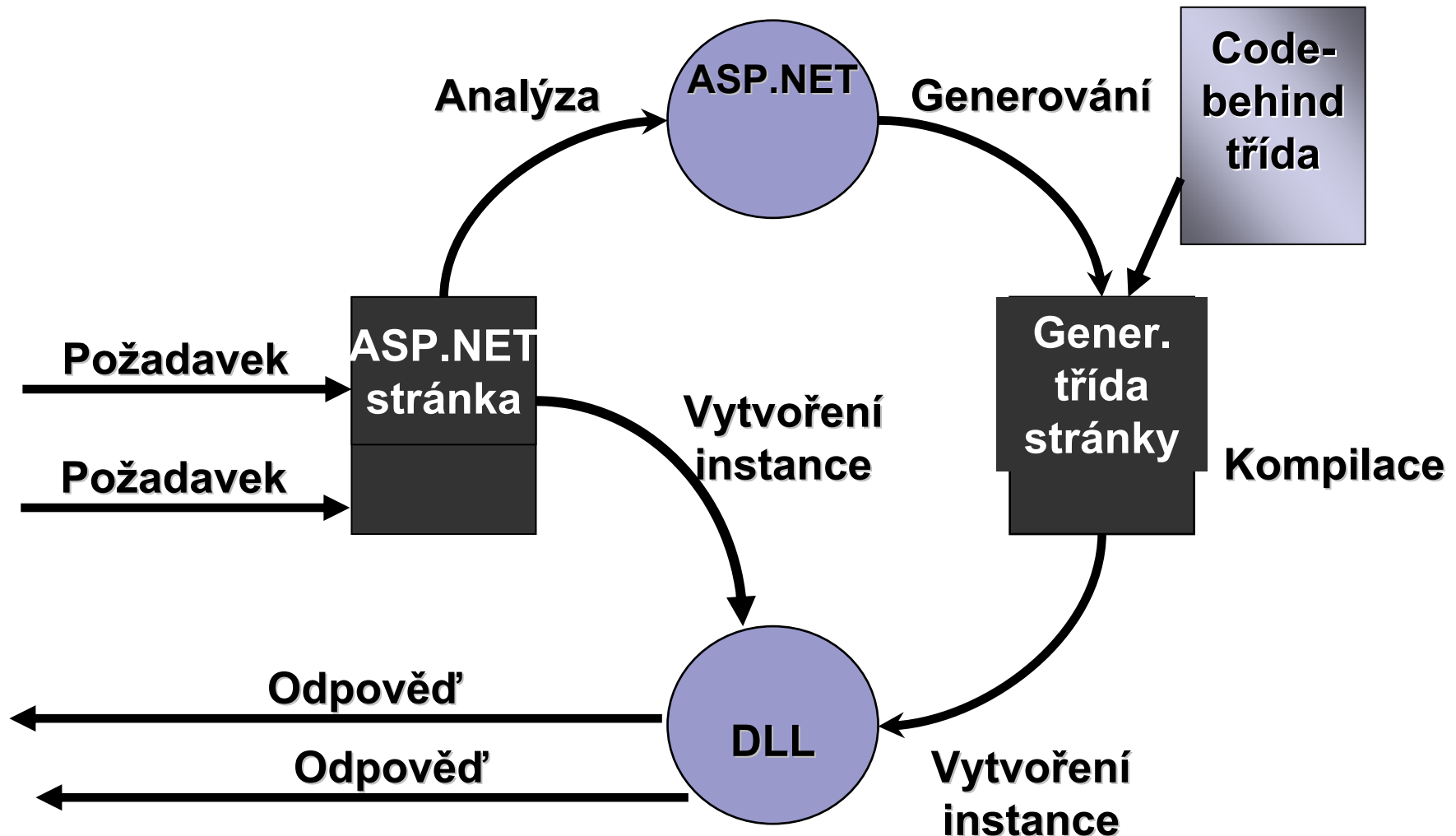
ASP.NET

a jeho objektová struktura

Processing aspx pages

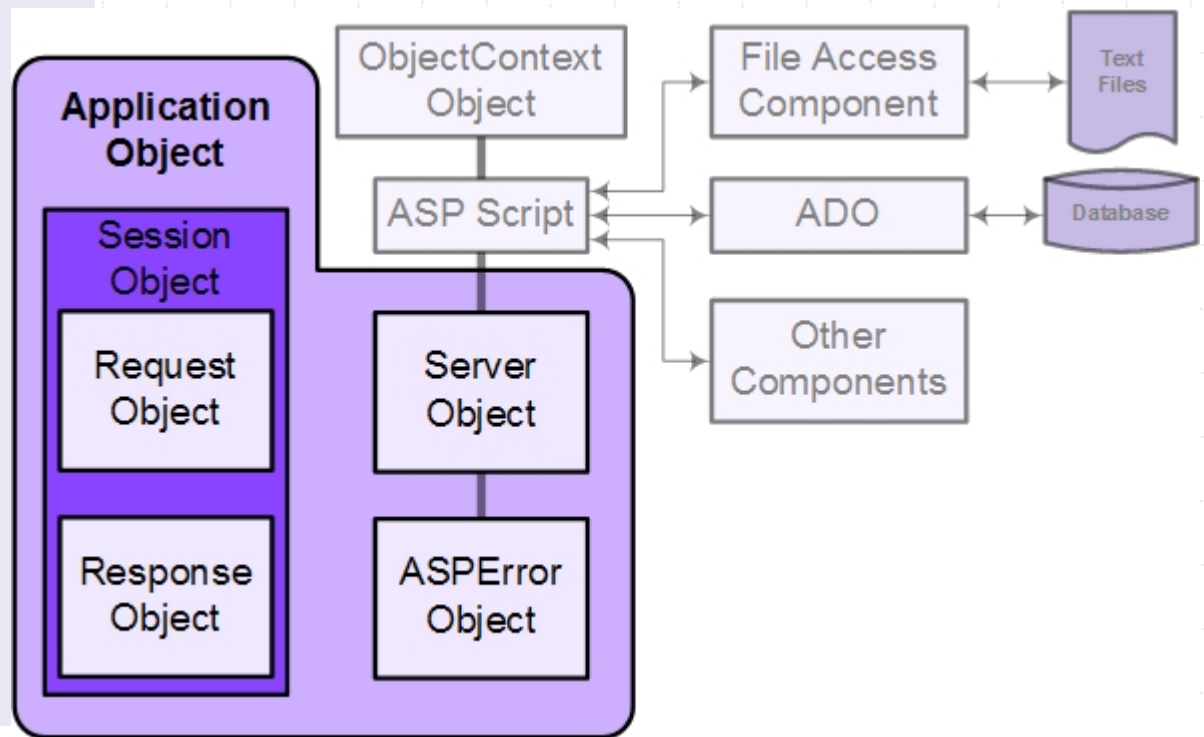


Kompilace ASP.NET Web Forms



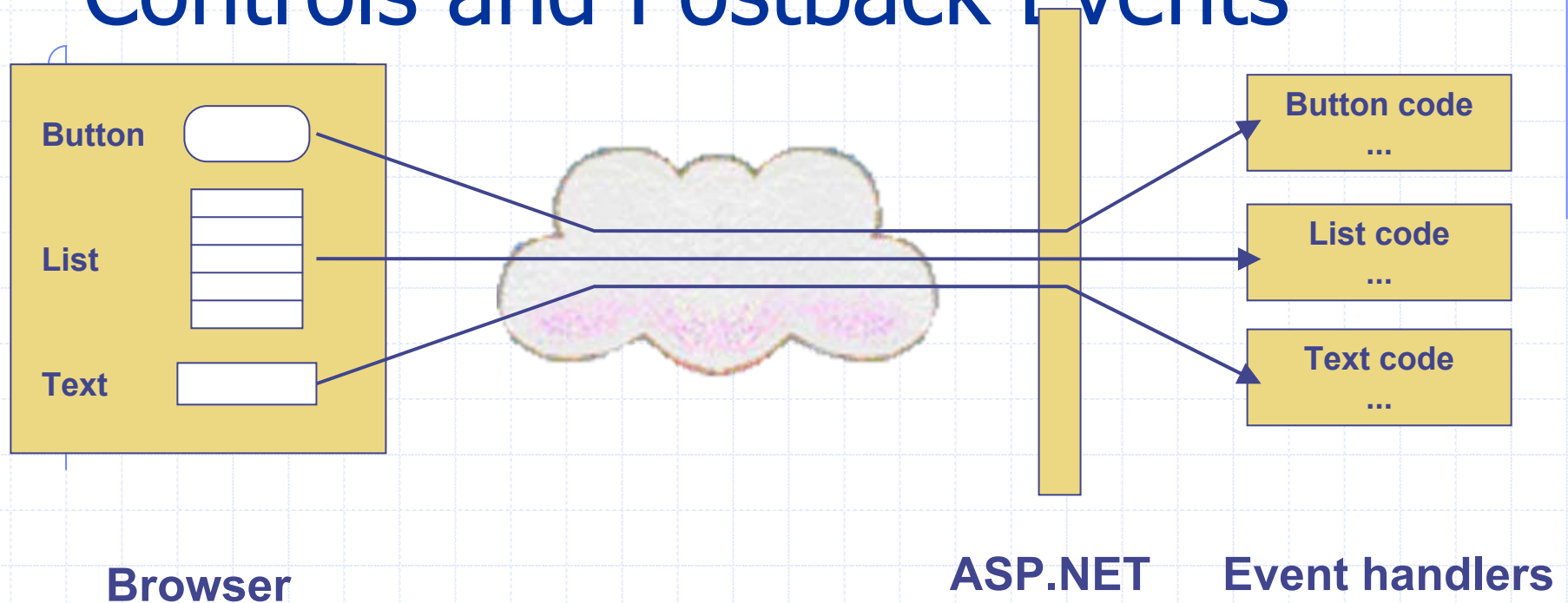
Hlavní objekty ASP.NET

- ◆ Application
- ◆ Session
- ◆ Request
- ◆ Response
- ◆ Server
- ◆ ASPError

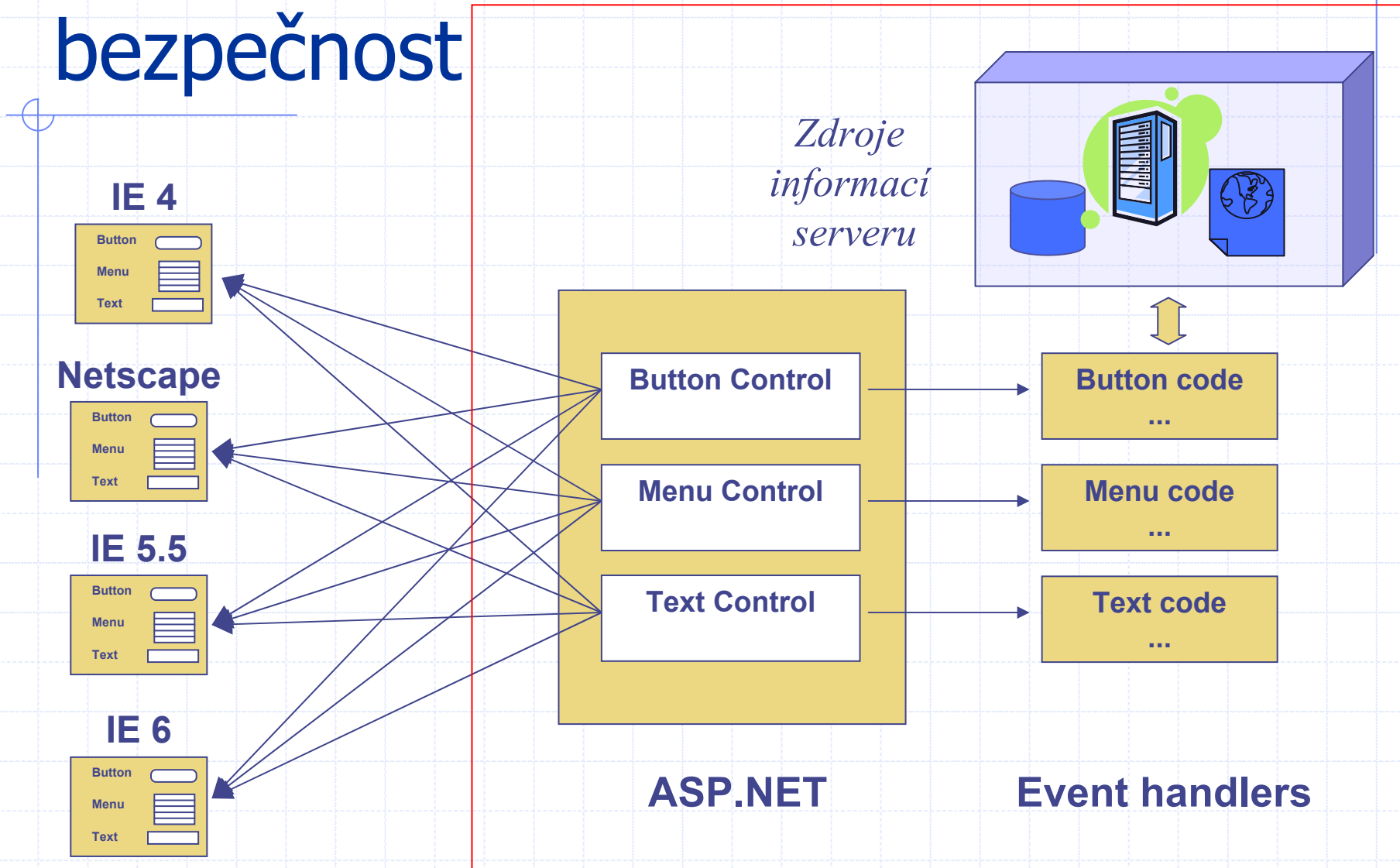


E. Petroustos 2000, *ASP 3: Instant Reference*, Sybex, San Francisco, p 29.

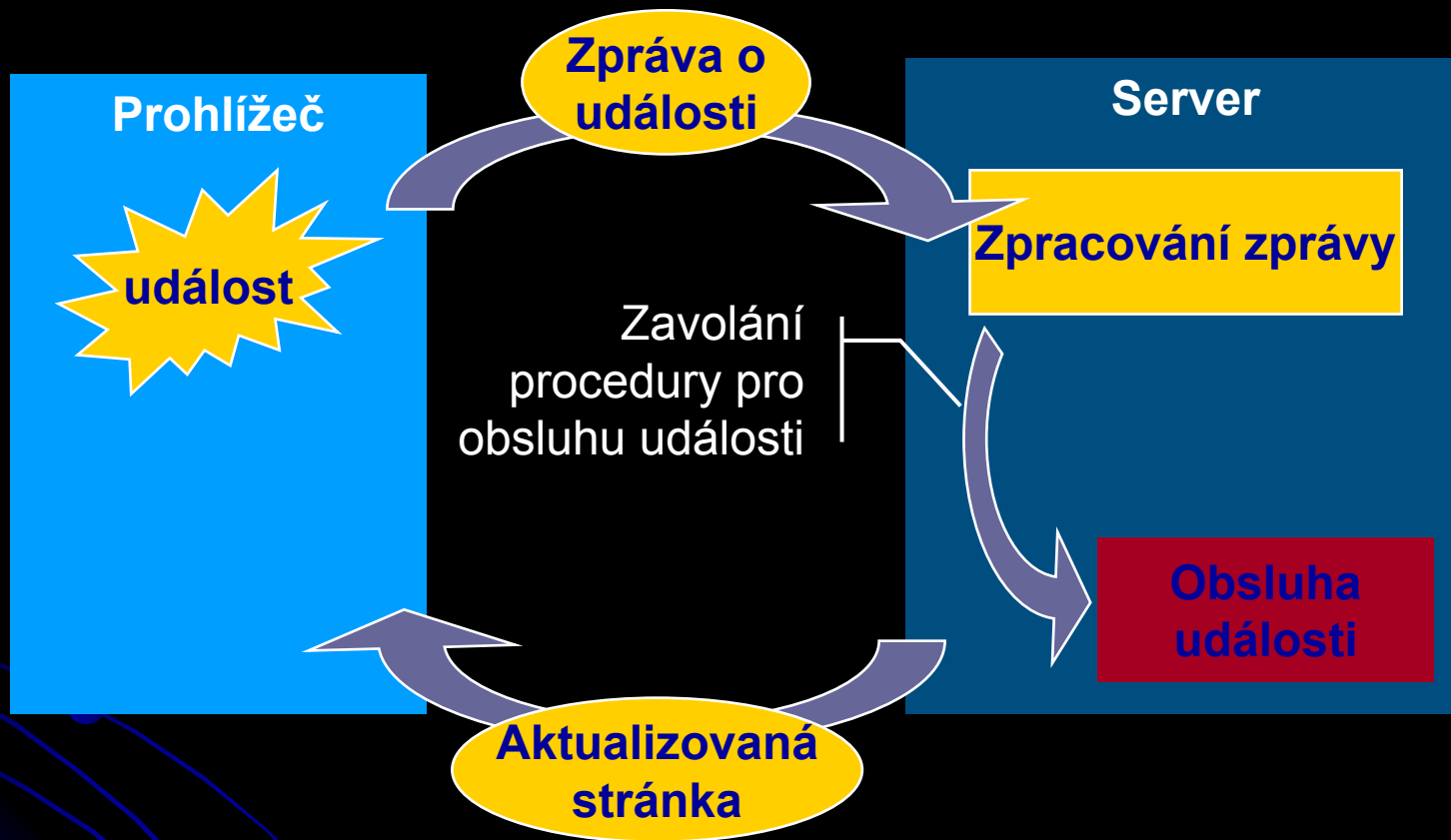
Controls and Postback Events



Vysoká kompatibilita a bezpečnost

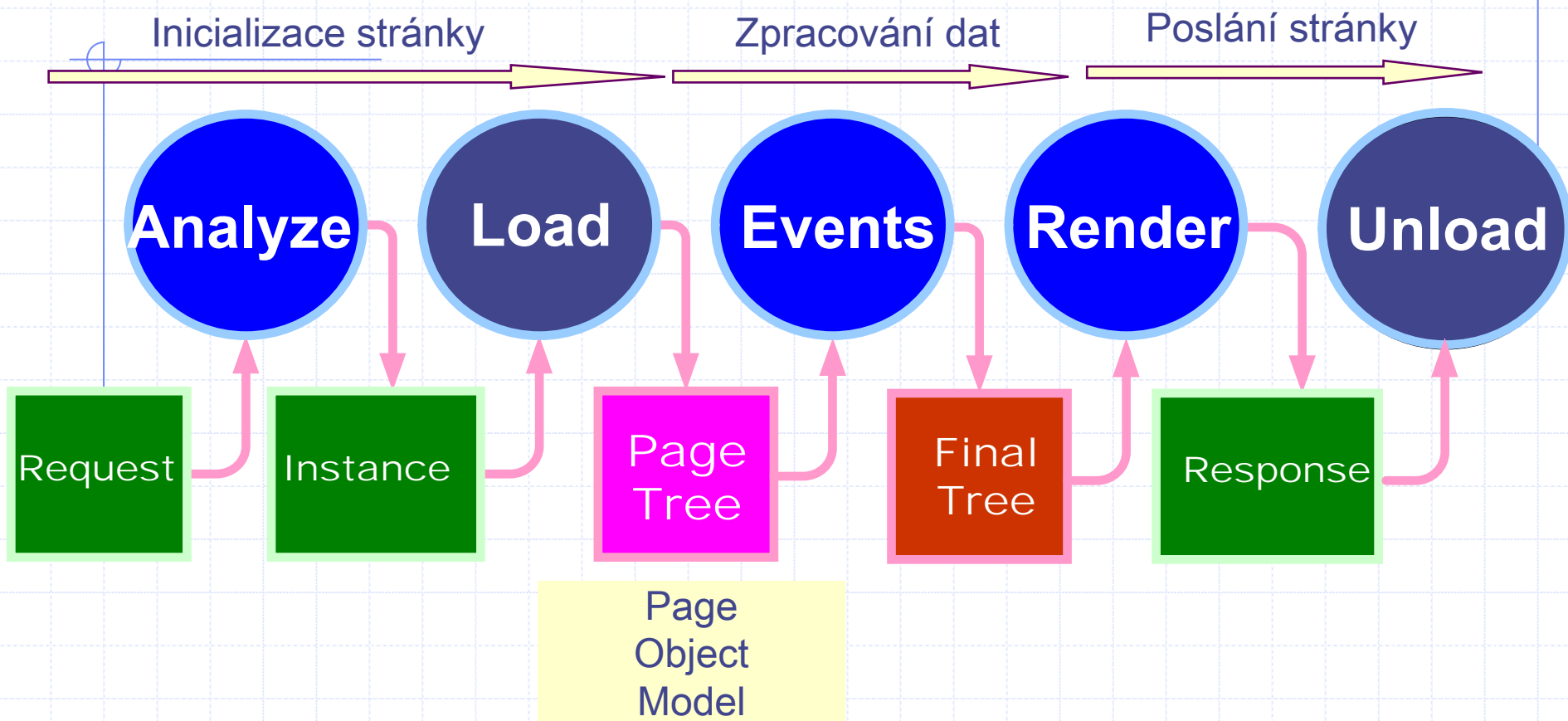


Zpracování události (PostBack)



Vysoká kompatibilita a bezpečnost

ASP.NET – princip objektového chování



Demonstrace ASP.NET objektů

☒ Polomer ☐ Obvod = ?

(C) CVUT-FEL

Demo

Napřed si ukážeme objektové
chování ASP.NET stránky

Kód ukázky 1/3

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        StringBuilder sb = new StringBuilder();
        foreach (string s in this.Request.Params)
            sb.AppendLine(s+":"+this.Request.Params[s]);
        string par = sb.ToString();
        //Label lbl=new Label(); lbl.Text="(C) CVUT-FEL";
        //form1.Controls.Add(lbl);
    }
}
```


Kód ukázky 2/3

```
protected void btVypocti_Click(object sender, EventArgs e)
{
    double vstup;
    CultureInfo ci = CultureInfo.GetCultureInfo("cs-CZ");
    CultureInfo cin = CultureInfo.InvariantCulture;
    if (!double.TryParse(tbVstup.Text, NumberStyles.Float, ci, out vstup)
        && !double.TryParse(tbVstup.Text, NumberStyles.Float,
                           cin, out vstup))
    {
        lbVysledek.Text = "?"; return;
    }

    if (rbObvod.Checked)
        lbVysledek.Text = (2 * Math.PI * vstup).ToString("0.000");
    else lbVysledek.Text = (vstup / (2 * Math.PI)).ToString("0.000");
}
```

Kód ukázky 3/3

```
protected override void OnUnload(EventArgs e)
{
    base.OnUnload(e);
}
protected void rbPolomer_CheckedChanged(object sender, EventArgs e)
{
    btVypocti_Click(sender, e);
}
protected void rbObvod_CheckedChanged(object sender, EventArgs e)
{
    btVypocti_Click(sender, e);
}
```