Distribuované systémy a výpočty (02)

Jan Janeček

katedra počítačů

České vysoké učení technické v Praze

HTML technologie

- HTML komunikace
  - WWW stránky
  - CGI skripty
  - Java servlety
  - JSP skriptlety

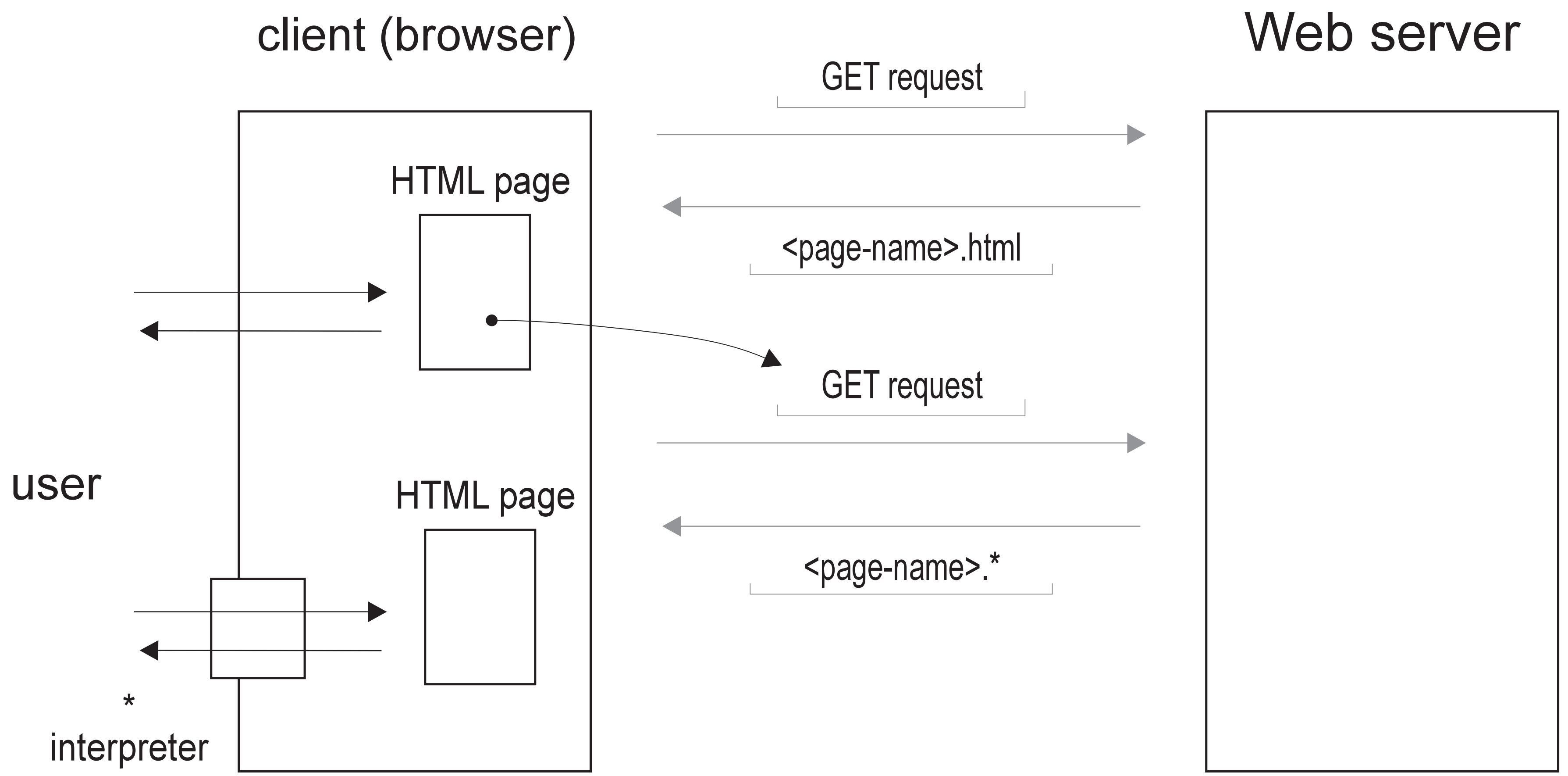
XML technologie

- XML komunikace
  - XML dokumenty
  - DOM / SAX parsing
  - nástroje XSLT a XPath

HTML - HyperText Markup Language

```
<html>
    <head>
        <title> HelloWorld </title>
    </head>

    <body>
        <h1> Header<h1>

        <p>
        Text . . .

        <table>
            <tr>
                        . . .
            </tr>
        </table>

        <a href="page-a/index.html">Page A</a>
    </body>
</html>
```

HTML stránka



HTML stránka

HTML request

http: \\ dsn.felk.cvut.cz \ HelloWorld.html

HTML response

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 1.0 ...
<html>
    <head>
        <title> HelloWorld </title>
    </head>

    <body>
        <h1> Hello, world! </h1>
    </body>
</html>
```

HTTP protokol

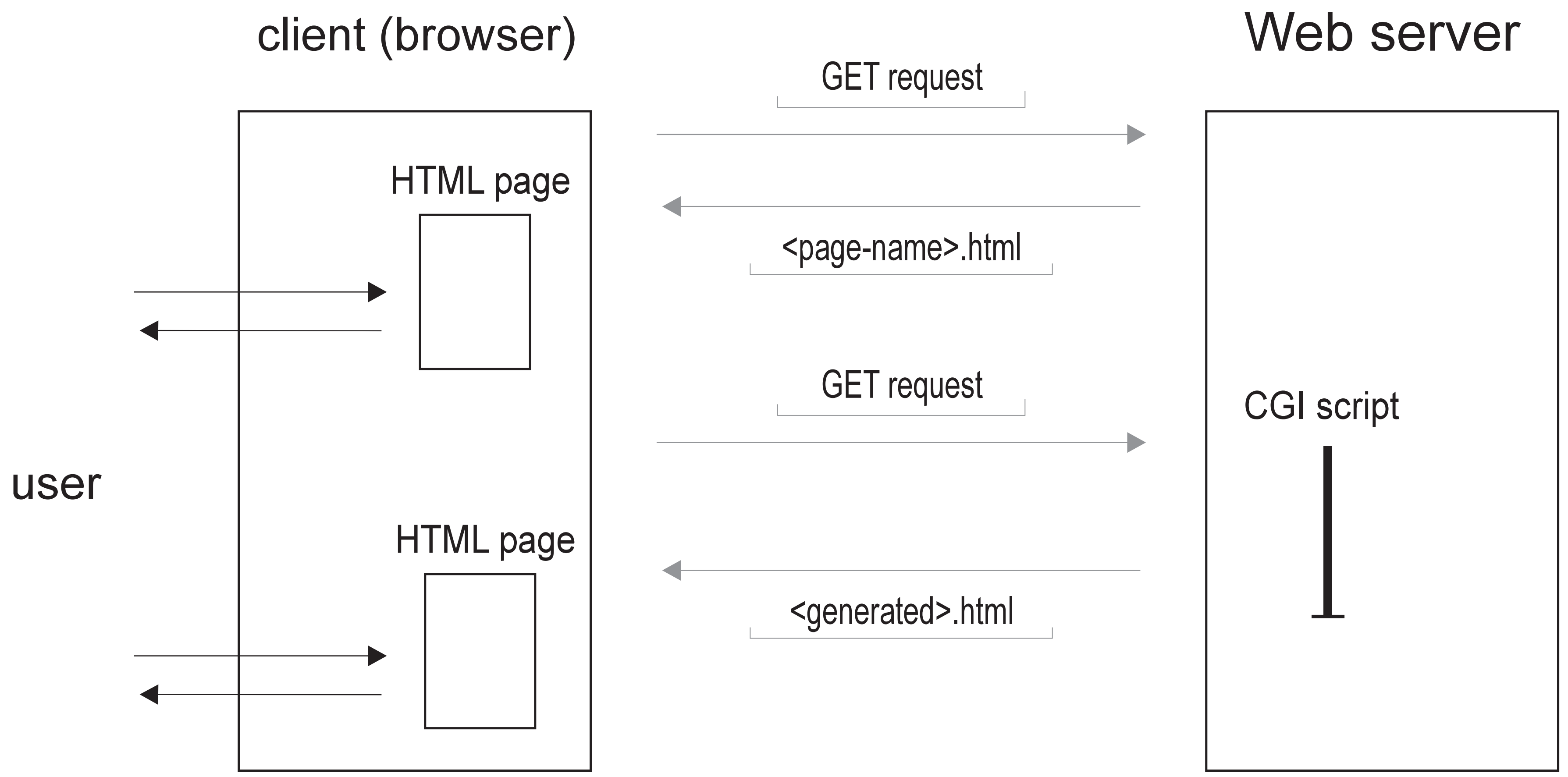
HTTP request

```
GET /index.html HTTP/1.1
Host: dsn.felk.cvut.cz
```

HTTP response

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
```

CGI script - metoda GET



CGI script - metoda GET

Metoda GET

http://dsn.felk.cvut.cz/HelloWorld.cgi

```
#include <stdio.h>

int main(void)
{
    printf("Content-Type: text/plain \n \n");
    printf("Hello World in C! \n");
}
```

CGI script - metoda GET

```
#include <stdio.h>

int main() {
    printf("Content-type: text/html \n \n") ;

    printf("<html> \n") ;
        printf("<head> \n") ;
            printf("<title> HelloWorld </title> \n") ;
        printf("</head> \n") ;

        printf("<body> \n") ;
            printf("<h1> Hello, world! </h1> \n") ;
        printf("</body> \n") ;
    printf("</html> \n") ;
    exit(0) ;
}
```

CGI script - metoda GET

Předání parametrů

http://dsn.felk.cvut.cz/HelloWorld.cgi ? name = value

```
<form action="http://dsn.felk.cvut.cz/cgi-bin/HelloWorld.cgi"
    method=GET>
Name: <input type=text name="name"><BR>
<input type=submit value="Send">
</form>
```

dcse ctu prague

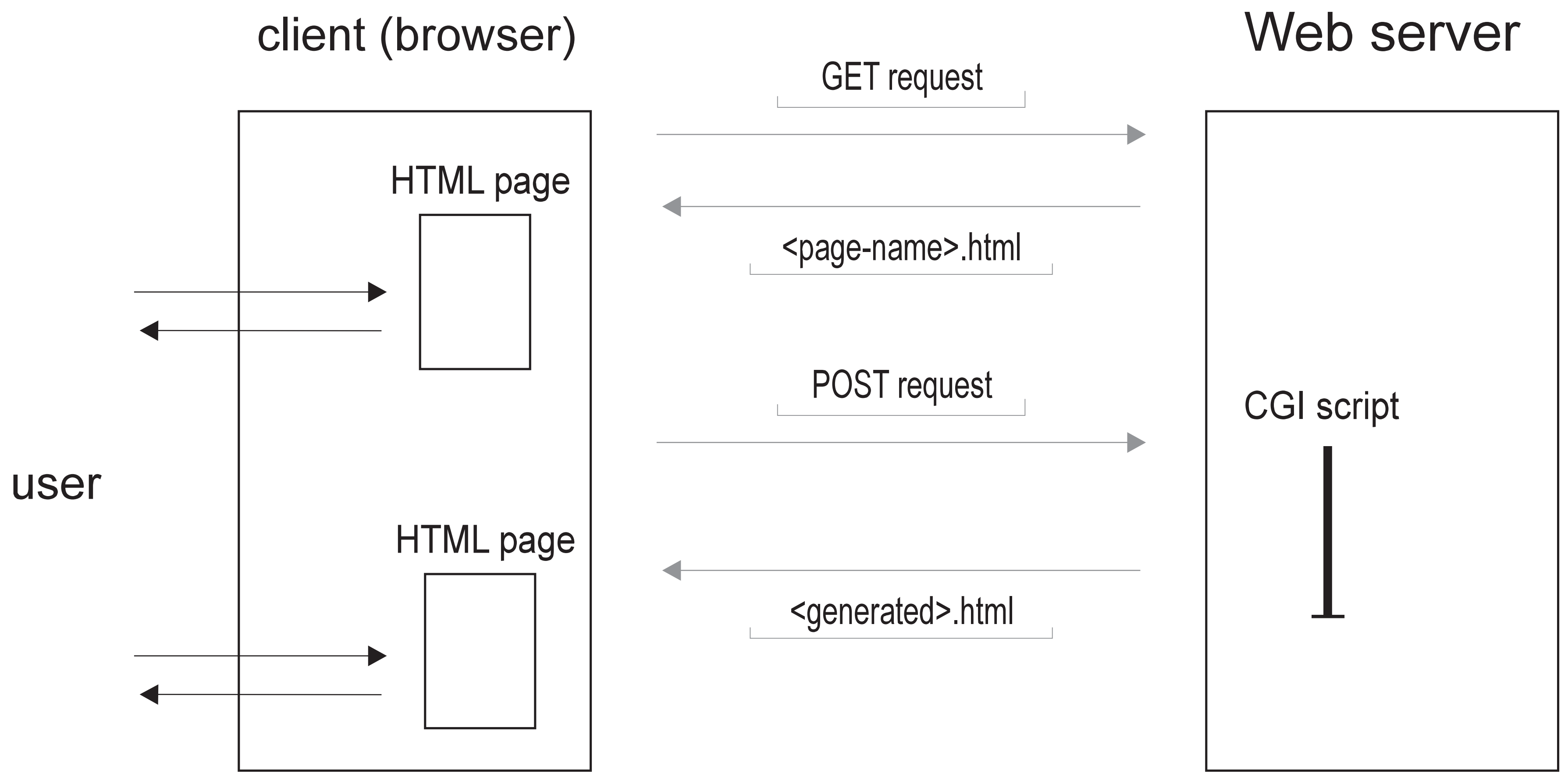
CGI script - metoda GET

```
#include <stdio.h>
int main() {
    char **cgivars ; int i ;
    cgivars = getcgivars() ;
    printf("Content-type: text/html \n \n") ;
    printf("<html> \n") ;
        printf("<head><title>CGI Results</title></head> \n") ;
        printf("<body> \n") ;
            printf("<h1> Hello, world! </h1> \n") ;
            printf("Your CGI input variables were: \n") ;
            printf("<ul> \n") ;
            for (i=0; cgivars[i]; i+= 2)
                printf("<li> [%s] = [%s] \n", cgivars[i], cgivars[i+1]) ;
            printf("</ul> \n") ;
        printf("</body> \n") ;
    printf("</html> \n") ;
    exit(0) ;
}
```

CGI script - metoda GET

```
#!/bin/sh
echo ’Content-type: text/html’
echo
echo ’<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">’
echo ’<HTML>’
echo ’<HEAD>’
echo ’<TITLE>CGI GET Service</TITLE>’
echo ’</HEAD>’
echo ’<BODY>’
echo ’<H1>Response :</H1>’

    zpracování parametrů z    $QUERY_STRING

echo ’</BODY>’
echo ’</HTML>’
```

CGI script - metoda POST



CGI script - metoda POST

Metoda POST

```
POST /addMessage HTTP/1.0
Host: www.mailtothefuture.com
Content-type: application/x-www-form-u
Content-length: 133

         data
```

CGI script - metoda POST

```
<form action="http://dsn.felk.cvut.cz/cgi-bin/HelloWorld.cgi"
    method=POST>
Name: <input type=text name="name"><BR>
<input type=submit value="Send">
</form>
```

parametry jsou zpracovány jako standardní vstup ...

Servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWWW extends HttpServlet {
  public void doGet(HttpServletRequest rq,
            HttpServletResponse rsp)
            throws ServletException, IOException {

      . . .

  }
}
```

Servlet

```
public void doGet(HttpServletRequest rq, HttpServletResponse rsp)
                throws IOException, ServletException
{
String name = rq.getParameter("name");

rsp.setContentType("text/html");

PrintWriter out = rsp.getWriter();
out.println("


             . . .


}
```

Servlet

```
public void doPost(HttpServletRequest rq, HttpServletResponse rsp)
            throws ServletException, IOException
{
    doGet(rq, rsp);
}
```

JSP - Java Server Pages

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; . . . >
<title>Java Server Pages (JSPs)</title>
</head>
    <body style="font-family:Comic Sans MS; color:darkblue">
    <form action="http://localhost/example.jsp" method="GET">
        <table border="0">
            <tr>
                <td>First name:</td>
                <td><input type="text" size="10" name="First"></td>
            </tr>
                        ...
        </table>
        <p><input type="submit" value="Send"> </p>
    </form>
</body>
</html>
```

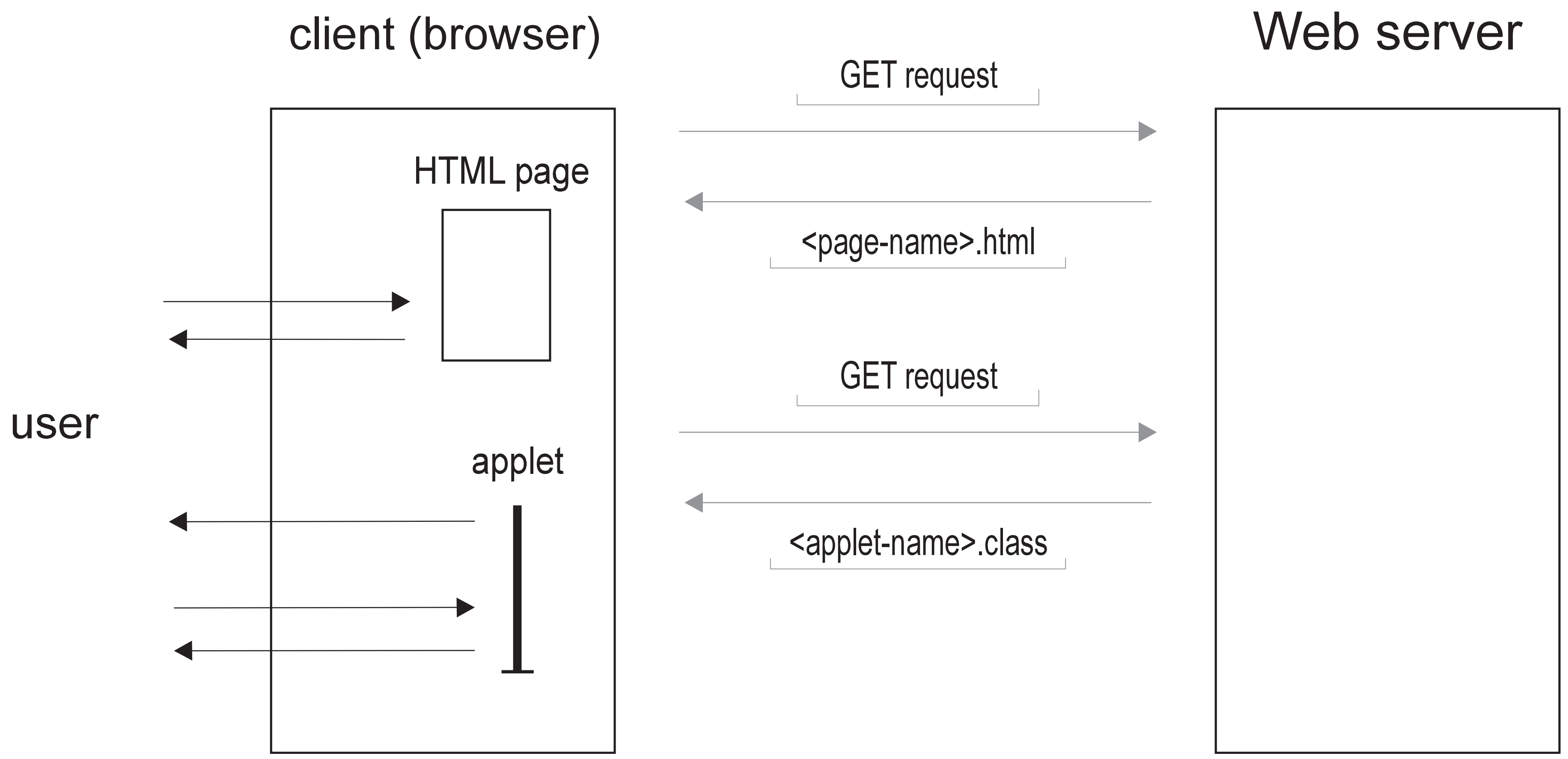
JSP - Java Server Pages

```
</html>

<p>
          . . .

Your name:
   <%= getParameter("First") %> <% = getParameter("Family") %>

          . . .
</html>
```

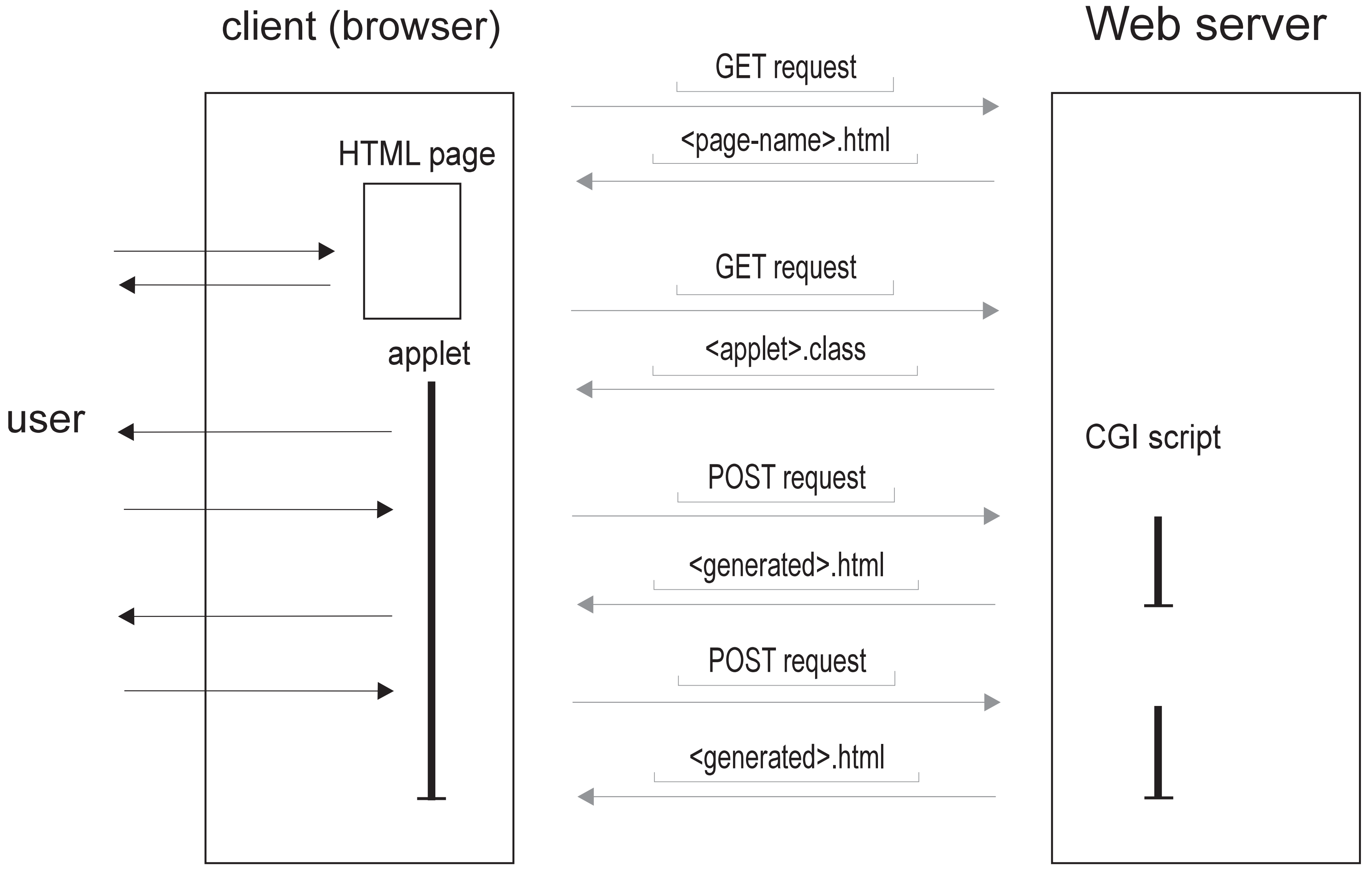
JSP - skriplet

```
<%@ page contentType="text/xml" import="java.util.*" %>

<%
    String firstName = request.getParameter("First").toString();
    String familyName = request.getParameter("Family").toString();
%>

            . . .

</html>

<p>

            . . .

Your name:
    <%= firstName + " " + familyName %>

            . . .
</html>
```

Applet


client (browser)
Web server
GET request
HTML page
<page-name>.html
GET request
user
applet
<applet-name>.class

Applet + CGI script



XML - Extensible Markup Language

```
<?xml version="1.0" encoding="UTF-8">
    <dokument>
        <data>
                Tady mám nějaká data . . .
        </data>
        . . . a tohle je jen výplň . . .
    </dokument>
```

XML - Příklad

```
<qa:book xmlns:qa="http://www.qa.com">

    <qa:title>A Few Good Men</qa:title>

    <qa:lentTo maxLoan="28">
        <B>Doe</B>, John
    </qa:lentTo>

</qa:book>
```

XML - Definice struktury dokumentu

DTD - Data Type Definition

```
<!DOCTYPE booklist [

<!ELEMENT booklist (book)+>
<!ELEMENT book (person, ...)>
<!ATTLIST book maxLoan CDATA #REQUIRED>
<!ELEMENT person (#PCDATA)> ]>
```

XML Schema

- nahrazuje dnes již zastaralý DTD
- formát definic je XML Schema

XML Schema - definice prvků

```
<?xml version="1.0"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="booklist">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="book" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="person" type="xsd:string"/>
              . . .
            </xsd:sequence>
            . . .
          <xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

XML Schema - definice atributů

```
<?xml version="1.0"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="booklist">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="book" maxOccurs="unbounded">
          <xsd:complexType>
            . . .
            <xsd:atribute name="maxLoan" type="xsd:string"/>
          <xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

XML Schema - definice prvků s atributy

```
. . .
<xsd:element name="booklist">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="book" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="person" type="xsd:string"/>
            . . .
          </xsd:sequence>
          <xsd:atribute name="maxLoan" type="xsd:string"/>
        <xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
. . .
```

dcse ctu prague

DOM parser - test XML dokumentu

```
import org.apache.xerces.parsers.DOMParser;
import org.xml.sax.SAXException;
import java.io.IOException;
public class XercesChecker {
  public static void main(String[] args) {
    if (args.length <= 0) {
      System.out.println("Usage: java XercesChecker URL");
      return; }
    String document = args[0];
    DOMParser parser = new DOMParser();
    try {
      parser.parse(document);
      System.out.println(document + " is well-formed."); }
    catch (SAXException e) {
      System.out.println(" . . . "); }
    catch (IOException e) {
      System.out.println(" . . . "); }
}
```

DOM parser - načtení XML dokumentu

```
import org.apache.xerces.parsers.DOMParser;
import org.w3c.dom.*;
public class XMLDOMParser {
  public static void main(String[] args) {
    if (args.length <= 0) {
      System.out.println("Usage: java XercesChecker URL");
      return; }
    String document = args[0];
    DOMParser parser = new DOMParser();
    try {
      Document doc = parser.getDocument(document);

      . . .

    catch (IOException e) {
      System.out.println(" . . . "); }
}
```

DOM parser - uzly dokumentu

```
package org.w3c.dom;
public interface Node {
  public static final short ELEMENT_NODE = 1;
  public static final short ATTRIBUTE_NODE = 2;
  public static final short TEXT_NODE = 3;
  public static final short CDATA_SECTION_NODE = 4;
  public static final short ENTITY_REFERENCE_NODE = 5;
  public static final short ENTITY_NODE = 6;
  public static final short PROCESSING_INSTRUCTION_NODE = 7;
  public static final short COMMENT_NODE = 8;
  public static final short DOCUMENT_NODE = 9;
  public static final short DOCUMENT_TYPE_NODE = 10;
  public static final short DOCUMENT_FRAGMENT_NODE = 11;
  public static final short NOTATION_NODE = 12;
```

DOM parser - metody

```
  public String getNodeName();
  public String getNodeValue() throws DOMException;
  public void setNodeValue(String nodeValue) throws DOMException;
  public short getNodeType();
  public String getNamespaceURI();
  public String getLocalName();
  . . .
  public Node getParentNode();
  public boolean hasChildNodes();
  public NodeList getChildNodes();
  public Node getFirstChild();
  public Node getLastChild();
  public Document getOwnerDocument();
  public boolean hasAttributes();
  public NamedNodeMap getAttributes();
. . .
```

SAX parser

```
import org.xml.sax.*;
import org.xml.sax.helpers.XMLReaderFactory;
import java.io.IOException;
public class SAXChecker {
  public static void main(String[] args) {
    if (args.length <= 0) {
      System.out.println("Usage: java SAXChecker URL");
      return; }
    try {
      XMLReader parser = XMLReaderFactory.createXMLReader();
      parser.parse(args[0]);
      System.out.println(args[0] + " is well-formed."); }
    catch (SAXException e) { . . . }
    catch (IOException e) { . . . }
  }
}
```

SAX parser - callbacks

```
package org.xml.sax;
public interface ContentHandler {
public void setDocumentLocator(Locator locator);
    public void startDocument() throws SAXException;
    public void endDocument() throws SAXException;
    public void startElement(String namespaceURI, String localName,
            String qualifiedName, Attributes atts) throws SAXException;
    public void endElement(String namespaceURI, String localName,
            String qualifiedName) throws SAXException;
    public void characters(char[] text, int start, int length)
            throws SAXException;
    public void ignorableWhitespace(char[] text, int start,
            int length) throws SAXException;
    public void processingInstruction(String target, String data)
            throws SAXException;

        . . .

}
```

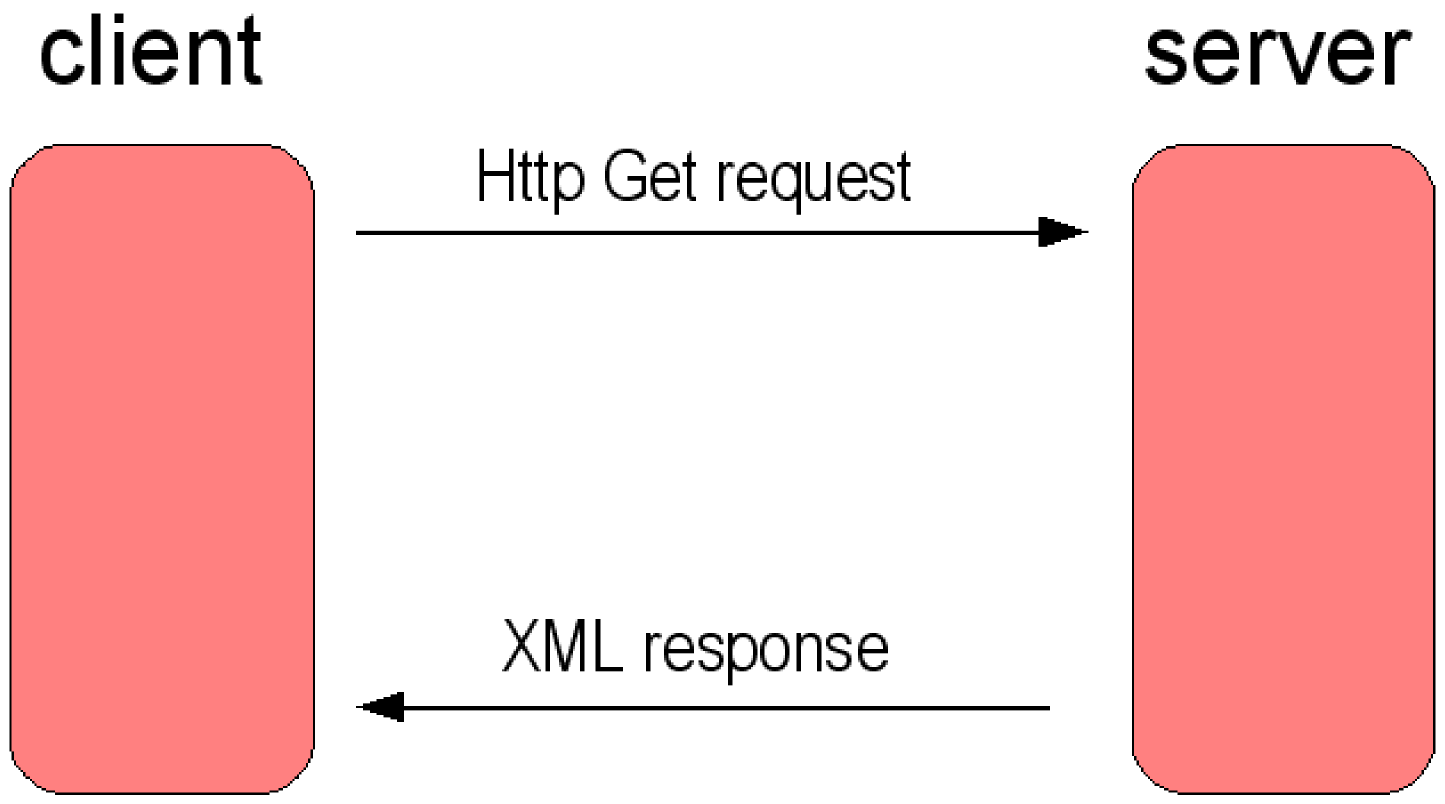
JSP - Java Server Pages

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; . . . >
<title>Java Server Pages (JSPs)</title>
</head>
    <body style="font-family:Comic Sans MS; color:darkblue">
    <form action="http://localhost/example.jsp" method="GET">
        <table border="0">
            <tr>
                <td>First name:</td>
                <td><input type="text" size="10" name="First"></td>
            </tr>
                        ...
        </table>
        <p><input type="submit" value="Send"> </p>
    </form>
</body>
</html>
```

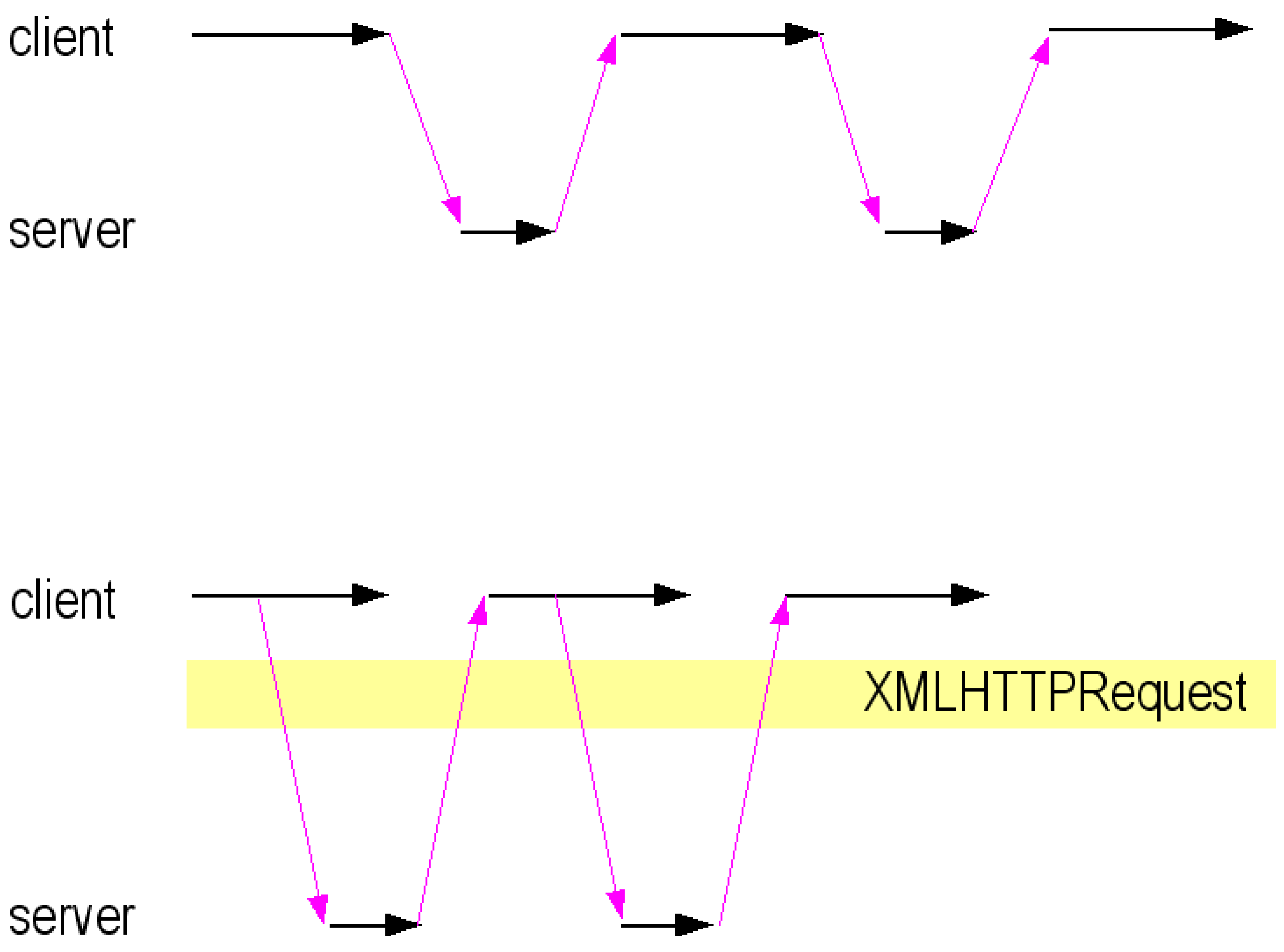
JSP - Java Server Pages

```
<%@ page contentType="text/xml" import="java.util.*" %>
<%
String firstName = request.getParameter("First").toString();
String familyName = request.getParameter("Family").toString();
%>

<ConfirmationMessage>
    <Name> <%= firstName + " " + familyName %> </Name>
    <When> <%= (new Date()).toLocaleString() %> </When>
</ConfirmationMessage>
```

AJAX - Asynchronous JavaScript And XML



AJAX - Asynchronous JavaScript And XML



AJAX - Asynchronous JavaScript And XML

```
<?xml version="1.0" encoding="UTF-8"?>
    <root>
    <data>
        This is some example data stored in an XML file
        and retrieved by JavaScript.
    </data>
    </root>
```

AJAX - Asynchronous JavaScript And XML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
    <html lang="en" dir="ltr">
        <head>
            <meta http-equiv="Content-Type" content="text/html; ...>
            <title>Developing with Ajax</title>
        </head>
        <body>
            <p>This page . . . </p>
            <p id="xmlObj">
            This is a default text for this web page.
            <a href="data.xml" title="View the XML data."
                onclick="ajaxRead('data.xml');
                this.style.display='none'; return false">
                View XML data.</a> </p>
        </body>
    </html>
```

AJAX - Asynchronous JavaScript And XML

```
<script type="text/javascript">
<!- -
  function ajaxRead(file) {
    var xmlObj = null;
    if(window.XMLHttpRequest) {
      xmlObj = new XMLHttpRequest();
    } else if(window.ActiveXObject) {
      xmlObj = new ActiveXObject("Microsoft.XMLHTTP");
    } else {
      return;
    }
```

dcse ctu prague

AJAX - Asynchronous JavaScript And XML

```
        xmlObj.onreadystatechange = function() {
          if(xmlObj.readyState == 4) {
            updateObj(’xmlObj’,
            xmlObj.responseXML.getElementsByTagName(’data’)[0].
                  firstChild.data);
          }
        }
          xmlObj.open (’GET’, file, true);
          xmlObj.send (”);
        }
        function updateObj(obj, data) {
          document.getElementById(obj).firstChild.data = data;
        }
      }
    - ->
    </script>
```