



# Základy tvorby interaktivních aplikací

Ing. Peter Drahoš, PhD.

LS 2012 - 2013

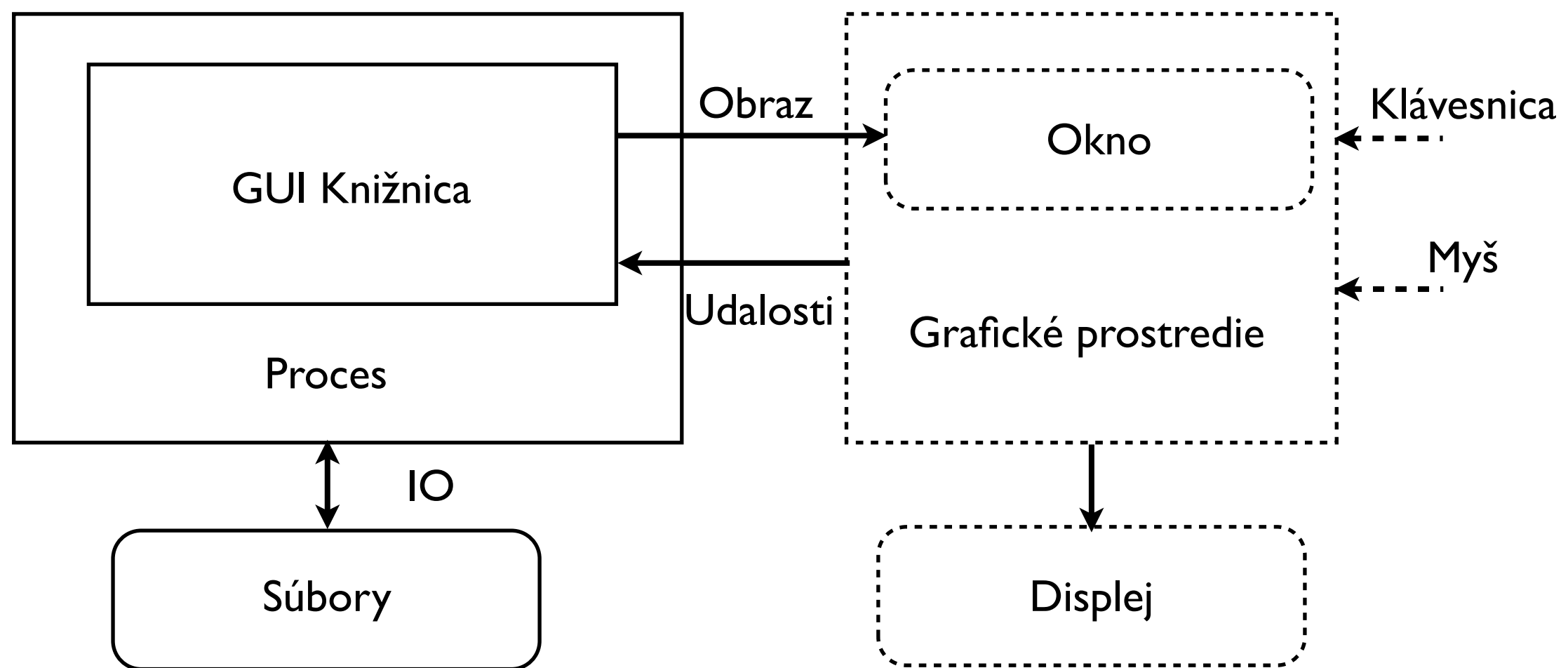
# Obsah

- Zápočtová skúška, správne odpovede
- Moderné interaktívne Web aplikácie
  - SOAP a REST
  - AJAX
  - JSON
- jQuery

# Zápočtová skúška

1. (2b) Nakreslite typickú štruktúru grafickej aplikácie s ohľadom na jej vstupy a výstupy.
2. (2b) Slovné opíšte čo je *MVC* a stručne vysvetlite jeho základné časti.
3. (2b) Vysvetlite čo je vzor *observer* a uveďte aspoň jedno jeho typické použitie v kontexte interaktívnych aplikácií.
4. (2b) Napíšte jednoduchý program v HTML5+JavaScript ktorý po načítaní dokumentu vypíše do *console* všetky parametre objektu *document*.
5. (2b) Definujte v jaz. JavaScript objekt *account* s parametrom *balance* a metódou *add* ktorá zvýši *balance* o hodnotu danú jej parametrom. Objekt musí obsahovať konštruktor ktorý nastaví počiatočnú hodnotu parametra *balance*.

1. (2b) Nakreslite typickú štruktúru grafickej aplikácie s ohľadom na jej vstupy a výstupy.



2. (2b) Slovné opíšte čo je **MVC** a stručne vysvetlite jeho základné časti.

- Architektonický návrhový vzor pre interaktívne aplikácie oddeľujúci dáta a logiku od ovládania a zobrazenia.
- *Model* - Aplikačná logika, dáta a hlavné algoritmy aplikácie
- *View* - Grafické či iné zobrazenie dát používateľovi
- *Controller* - Spracovanie ovládania aplikácie, ovláda model a view

3. (2b) Vysvetlite čo je vzor *observer* a uveďte aspoň jedno jeho typické použitie v kontexte interaktívnych aplikácií.

- Vzor observer je návrhovým vzorom pre tvorbu softvéru ktorý definuje objekt *observer* ktorý je schopný reagovať na zmeny v objekte *subject* s využitím zoznamu pozorovateľov ktorým sú zasielané správy.
- Typické použitie je najme v kontexte GUI aplikácií kde sa vzor využíva na propagáciu správ a udalostí.

4. (2b) Napíšte jednoduchý program v HTML5+JavaScript ktorý po načítaní dokumentu vypíše do *console* všetky parametre objektu *document*.

### HTML5 + JavaScript

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>HTML5+JavaScript</title>
    <meta charset="UTF-8">
    <script>
      window.onload = function() {
        for (var i in document) { console.log(document[i]) }
      }
    </script>
  </head>
  <body>
  </body>
</html>
```

5. (2b) Definujte v jaz. JavaScript objekt *account* s parametrom *balance* a metódou *add* ktorá zvýši *balance* o hodnotu danú jej parametrom. Objekt musí obsahovať konštruktor ktorý nastaví počiatočnú hodnotu parametra *balance*.

### JavaScript

```
function Account(balance) {  
    this.balance = balance  
    this.add = function(sum) {  
        this.balance += sum  
    }  
}
```

### JavaScript

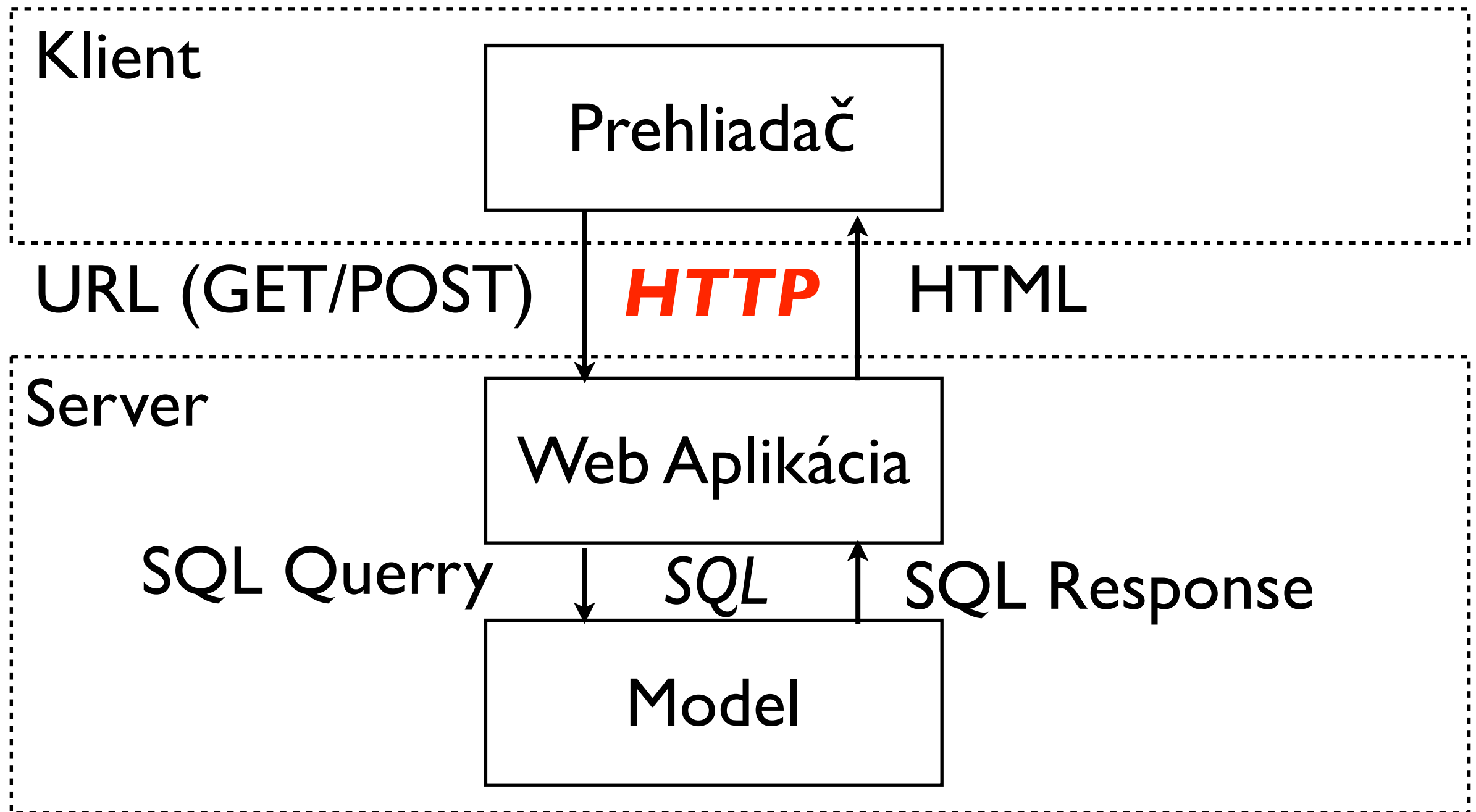
```
function Account(balance) {  
    this.balance = balance  
}  
  
Account.prototype.add = function(sum) {  
    this.balance += sum  
}
```



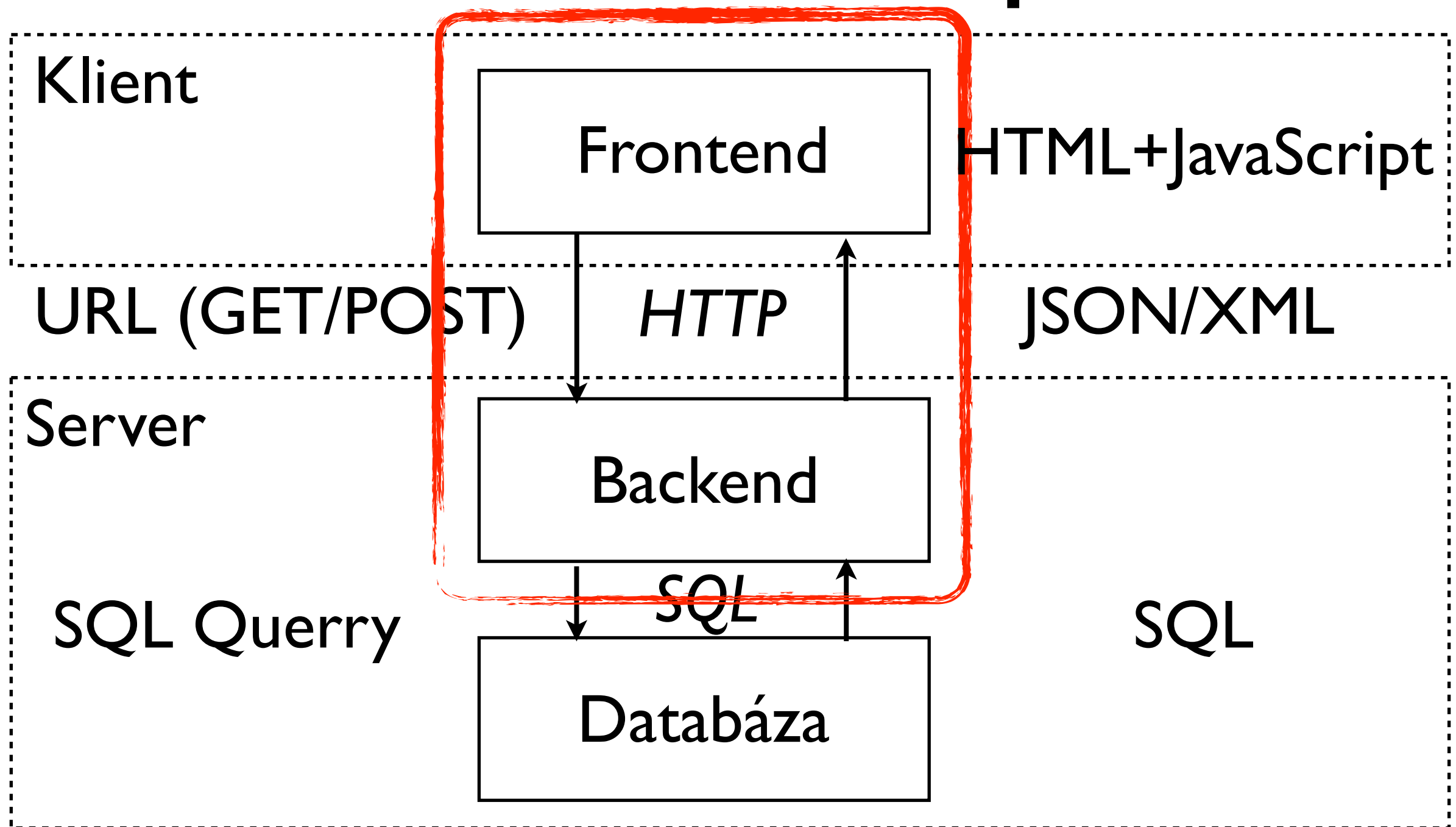
# Moderné Web aplikácie

- Minimalizácia odozvy pri práci s aplikáciou
  - Odstránenie nutnosti generovania stránky na strane servera po každej akcii
  - Zmenšenie množstva prenesených dát
- Vyššia zložitosť, klient už nie je len *presenter* ale je integrálnou súčasťou aplikácie
- Ide o klient server architektúru ktorá stavia na Web technológiách s

# Klasické Web aplikácie



# Moderné Web aplikácie



# Moderné Web aplikácie

- *Frontend* - Časť aplikácie ktorá ma za úlohu poskytovať rozhranie pre používateľa.
- Typicky spája funkcionality *view* a *controller*
- *Backend* - Aplikačná časť ktorá obsahuje funkcionality očakávanú od časti *model*
- Komunikácia je realizovaná zasielaním HTTP požiadaviek na rozhranie, API backend-u

# Moderné Web aplikácie

- *API* - Application Programming Interface
- V kontexte servera je to rozhranie za pomoci ktorého komunikuje *frontend* a *backend*
- Pre Web aplikácie sa často krát používa označenie WebAPI
- Prenos dát je zväčša realizovaný pomocou XML či JSON formátu

# AJAX

- *AJAX* - Asynchronous JavaScript and XML
- Umožňuje zaslať HTTP dopyt za pomoci JavaScript-u
- Odozva typicky neobsahuje žiadne HTML ale len dáta ktoré budú pomocou manipulácie DOM zobrazené
- Dáta prichádzajú asynchrónne po zaslaní dopytu a zvečša obsahujú údaje vo formáte XML

# XML a SOAP

- SOAP - Simple Object Access Protocol
- Protocol na výmenu informácií v kontexte web aplikácií za pomoci XML
- Správa je vložená do HTTP protokolu
- XML Information set definuje obsah a štruktúru XML správy

# XML a SOAP

## HTTP + XML

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: 299

SOAPAction: "http://www.w3.org/2003/05/soap-envelope"

```
<?xml version="1.0"?>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
```

```
  <soap:Header>
```

```
  </soap:Header>
```

```
  <soap:Body>
```

```
    <m:GetStockPrice xmlns:m="http://www.example.org/stock">
```

```
      <m:StockName>IBM</m:StockName>
```

```
    </m:GetStockPrice>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```



# XML a SOAP

- Výhody
  - Možno komunikovať i cez iný protokol ako HTTP
  - Komunikácia je obojstranná a len v XML
- Nevýhody
  - XML je pomerne neefektívny formát a vyžaduje spracovanie na strane klienta i servera
  - Problém s prenosom binárnych dát

# JSON a REST

- REST- Representative State Transfer
- Dominantný komunikačný model aplikácií v prostredí webu
- Využíva HTTP/1.1 na prenos stavu surovín(resource)
- Pod surovinou rozumieme čokoľvek čo môžeme adresovať a manipulovať v kontexte aplikácie

# JSON a REST

- Rozhranie využíva dopyt a odozvu poztavenú na HTTP protokole
- Dopyt môže byť typu GET, PUT, POST, DELETE
- GET - Vyžiadanie stavu suroviny
- POST - Nový záznam
- PUT - Zmena záznamu
- DELETE - Odstránenie záznamu

# JSON a REST

Shell

```
curl -i https://api.github.com/users/drahosp/orgs
```

- Príklady
  - GitHub API
  - Twitter API
  - ...

# JSON

- Využíva syntax jaz. JavaScript bez možnosti definovania funkcií
- Jednoduchý a ľahko čitateľný formát ktorý nie je nutné spracúvať ako XML
- Podpora v prehliadačoch cez `JSON.parse()`

# JSON

## JSON

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}
```

# JSON a AJAX

## JavaScript

```
var my_JSON_object = {};  
var http_request = new XMLHttpRequest();  
http_request.open("GET", url, true);  
http_request.onreadystatechange = function () {  
    var done = 4, ok = 200;  
    if (http_request.readyState == done && http_request.status == ok) {  
        my_JSON_object = JSON.parse(http_request.responseText);  
    }  
};  
http_request.send(null);
```

# Prestávka



# jQuery

- jQuery je knižnica určená na rýchlu manipuláciu s DOM
- Radikálne redukuje čas vývoja web aplikácií
- Umožňuje tvorbu rôznych špeciálnych efektov a animácií
- Poskytuje funkcionality pre tvorbu AJAX dopytov a zjednodušuje ich spracovanie

# jQuery

- Typicky ukladáme knižnicu k vyvíjanej aplikácii

## HTML + JavaScript

```
<html>
<head>
<script type="text/javascript" src="js/jquery-1.9.1.min.js"></script>
....
```

- Môžeme použiť i odkaz na google projekt

## HTML + JavaScript

```
<html>
<head>
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1/
jquery.min.js"></script>
....
```

# DOM manipulácia

## HTML + JavaScript

```
...  
<script type="text/javascript">  
    $(function(){  
        $("#p1").text("Hello World");  
    });  
</script>  
  
<body>  
    <p id="p1"></p>  
</body>
```

## HTML + JavaScript

```
$(document).ready(function(){  
    $("#p1").text("Hello World");  
});
```

# Volanie Funkcionality

- Volanie pomocou funkcie jQuery

```
jQuery
```

```
jQuery("#p1");
```

- Volanie skrátenou formou

```
jQuery
```

```
$("#p1");
```

# Selektor

- Volanie nad jediným elementom

```
jQuery  
$("#p1");
```

- Nastavenie viacerých elementov

```
jQuery  
$("p").text("Hello World");  
  
<body>  
  <p id="p1"></p>  
  <p id="p2"></p>  
</body>
```

# Selektor

- Hľadanie na základe parametra

jQuery

```
$("#p[important=true]").text("This is an important paragraph.");
```

```
<p important="true"></p>
```

```
<p id="p2"></p>
```

- Hľadanie na základe tried

jQuery

```
$(function(){  
    $(".c1").text("You have class c1");  
});
```

```
<p class="c1">Hello World</p>
```

```
<p class="c2">Hello Moon</p>
```

# Selektor

- Pokročilé hľadanie

## jQuery

```
$(function(){
    $("tr:even").css("background-color", "grey");
    $("tr:odd").css("background-color", "yellow");
});
```

```
<table border="1">
  <tr>
    <td>John Doe</td>
    <td>19</td>
  </tr>
  <tr>
    <td>Jane Doe</td>
    <td>21</td>
  </tr>
  <tr>
    <td>Mary Doe</td>
    <td>22</td>
  </tr>
</table>
```

John Doe	19
Jane Doe	21
Mary Doe	22

# jQuery Objekty

- jQuery neposkytuje rovnaké rozhranie pre manipuláciu s DOM objektami, má vlastne
- Je však možné ľahko previesť akýkoľvek DOM objekt na jQuery objekt

```
jQuery
```

```
$("#p1")
```

```
$(document.getElementById("p1"))
```



# Udalosti

- Ľahko možno definovať spracovanie akcií nad viacerými elementami

## jQuery

```
$("#p").click(  
    function() {  
        alert($(this).text());  
    }  
);
```

```
<p>  
Hello world!  
</p>  
<p>  
Hello moon!  
</p>
```

# Ret'azenie

- Ľahko možno definovať spracovanie akcií nad viacerými elementami

jQuery

```
$("#button").click(function(){  
    $("#p1").css("color","red").slideUp(2000).slideDown(2000);  
});
```

```
<p id="p1">jQuery is fun!!</p>  
<button>Click me</button>
```

# jQuery a AJAX

- Pohodlné spracovanie zasielania a spracovania udalostí

## jQuery

```
$("#button").click(function(){  
    $.get("test_json.php",function(data,status){  
        var json = JSON.parse(data)  
        console.log(json)  
    });  
  
<button>get json</button>
```

# Prehľad MVC knižníc

- Existuje veľké množstvo knižníc pre JavaScript ktoré výrazne zjednodušujú tvorbu komplexných aplikácií. vid. prehľad pre 2013
- Za spomenutie stojí najme knižnica Cappuccino ktorá modeluje tvorbu Web aplikácií v podobe známej z tvorby Desktop aplikácií pre Apple OSX