DCGI

DEPARTMENT OF COMPUTER GRAPHICS AND INTERACTION

WA1 (TW1)

cvičení 8
Smarty a MVC

Osnova cvičení

1. Organizace, seznámení s prostředím, HTTP, HTML
2. CSS
3. Skriptování na straně klienta
4. Jazyk PHP
5. Obsluha fomulářů, udržení stavu aplikace (sezení)
6. Přístup k databázi, PDO
7. OOP v PHP
8. **MVC, Smarty**
9. **MVC - pokračování**
10. Ajax a PHP
11. Autentizace a autorizace
12. Pear a Zend
13. Odevzdávání semestrálních úloh, zápočet

Osnova přednášek

- **Organizace, seznámení s prostředím, HTTP, HTML**
- **CSS**
- **Skriptování na straně klienta**
- **Jazyk PHP**
- **Obsluha fomulářů, udržení stavu aplikace (sezení)**
- **Přístup k databázi, PDO**
- **OOP v PHP**
- **MVC, Smarty**
- **MVC - pokračování**
- Ajax a PHP
- Autentizace a autorizace
- Pear a Zend

Užitečné odkazy

- http://www.php.net/manual/en/
- http://smarty.php.net/

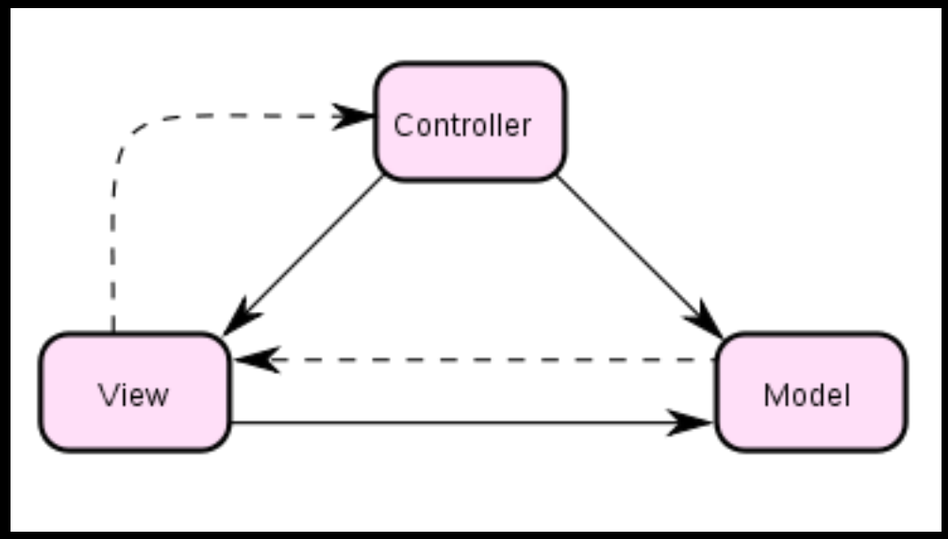
DCGI

Opakování

- Co je to MVC?
- K čemu je to dobré
- Jak to vypadá na webu?
- Jak vypadá varianta Front Controller?
- Co jsou to šablony?
- Co je to SMARTY?

DCGI

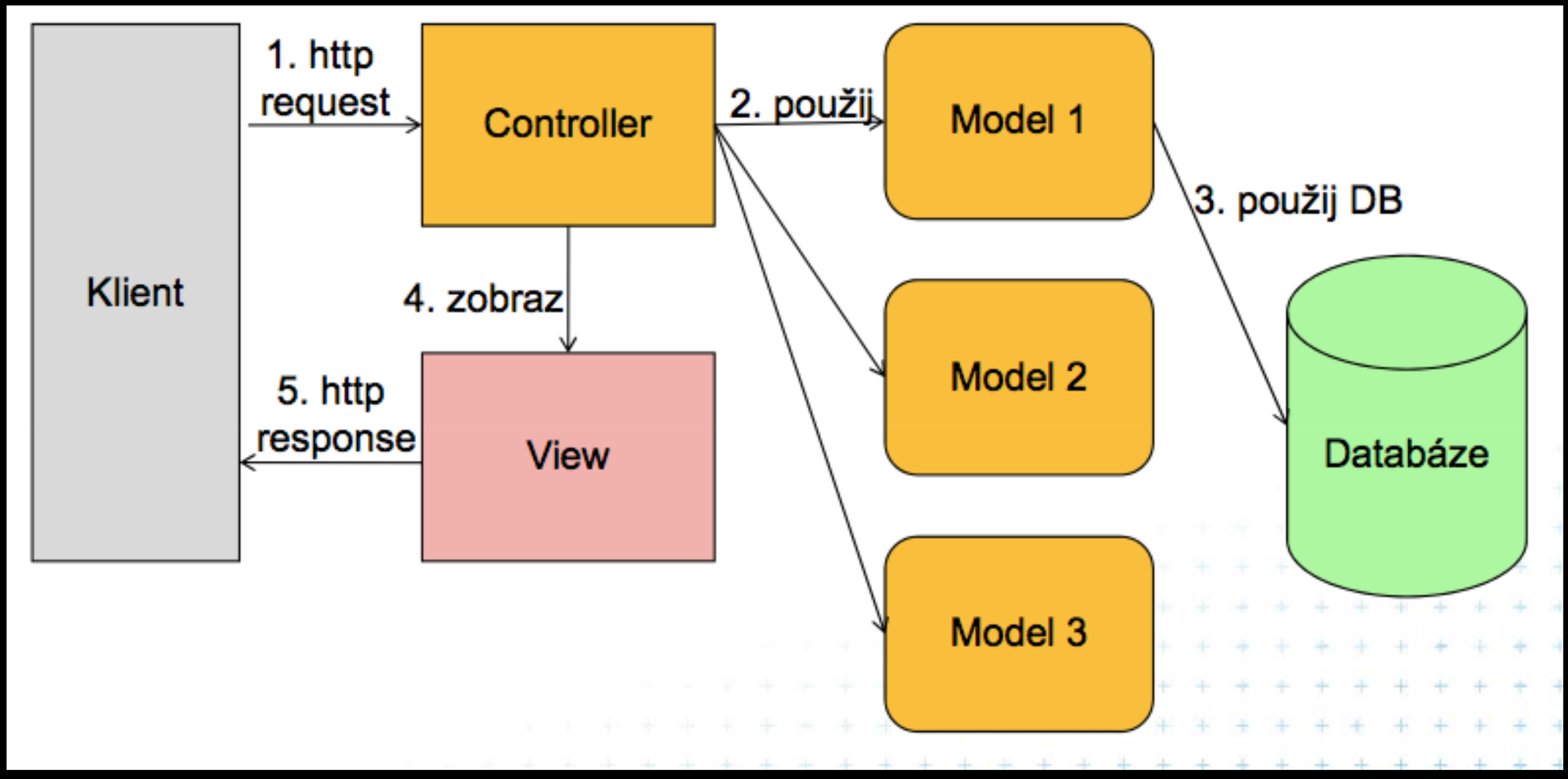
MVC



- Model - chování a data v aplikace
- View - reprezentuje to co je v modelu uživateli prostřednictvím UI
- Controller - řídí aplikaci - přijímá uživatelský vstup a volá model a view k provedení patřičných akcí

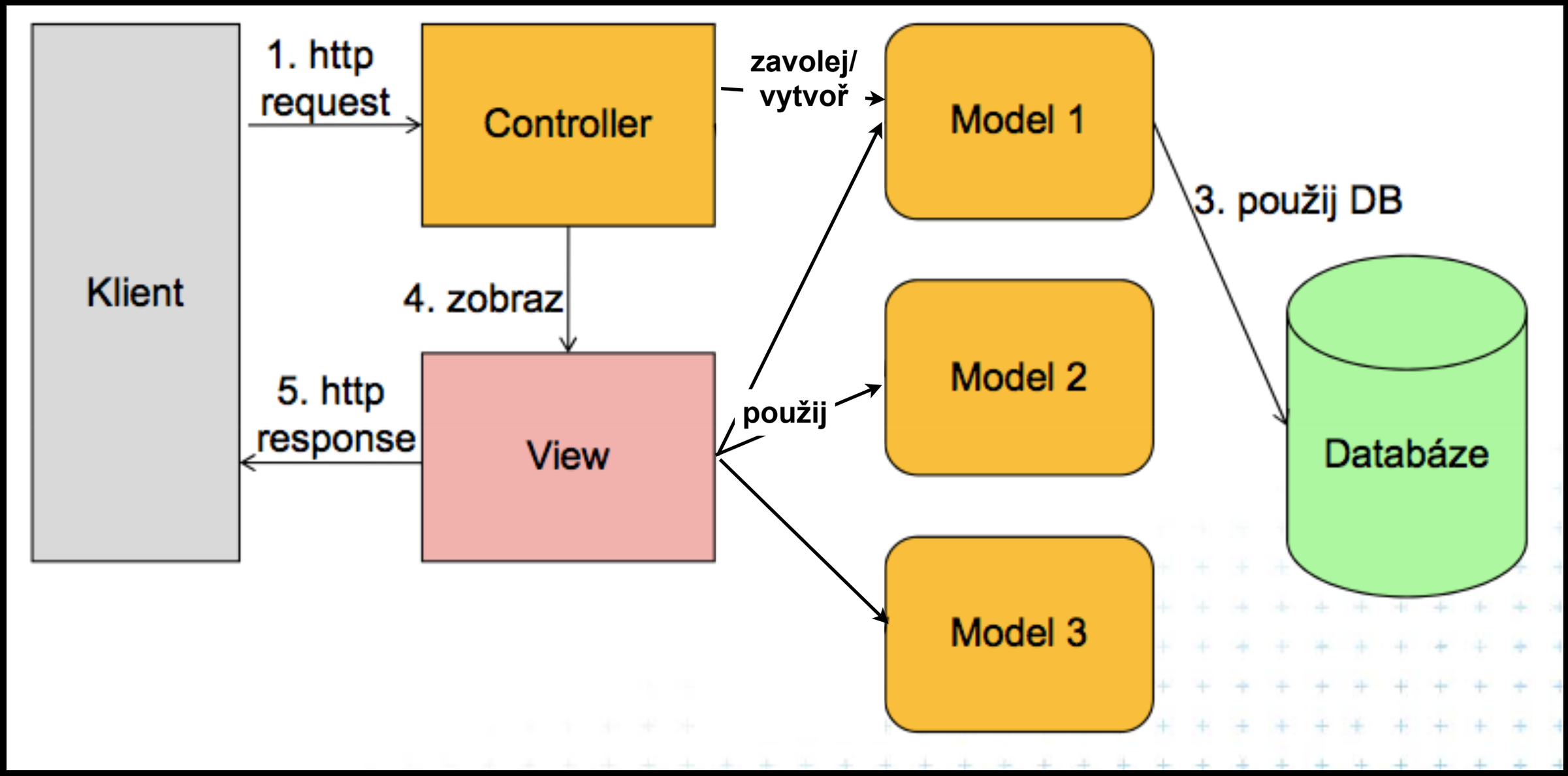
DCGI

MVC web


1. http request
Controller
2. použij
Model 1
3. použij DB
Klient
4. zobraz
Model 2
Databáze
5. http response
View
Model 3

DCGI

MVC web


Klient
1. http request
Controller
zavolej/ vytvoř
Model 1
3. použij DB
4. zobraz
5. http response
View
použij
Model 2
Model 3
Databáze

DCGI

Smarty

- Oddělení aplikační a prezentační logiky
- V podstatě podpora tvorby view

PHP Skript

```php
<?php
// inicializace smarty engine
require_once("init_smarty.php");

// vyrobim si data, v tomto pripade info o
datumu (to je model)
$datum = date ("d.m.Y");

// vytvor sablonu
$templatovaci_objekt = new T_Template();

//prirad data do sablony
$templatovaci_objekt->assign("datum",
$datum);

// nech to zobrazit
$templatovaci_objekt-
>display('hello_world.html');

?>
```

Template

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html>
  <head>
    <title>
      Hello world
    </title>
  </head>
  <body>
    Dnes je: {$datum}
  </body>
</html>
```

Zadání
Instalace Smarty

- Stáhněte zadani.zip z Eduxu.
- Balíček rozbalte do Vámi zvoleného adresáře viditelnéno z webu.
- Pro správnou funkci je nutné, aby měl Smarty právo zápisu do adresářů templates_c a cache.
  - Toto zajistíte příkazem chmod 777 templates_c cache z shellu. Prostudujte ukázkovou aplikaci v souboru index.php a index.xhtml. Identifikujte jednotlivé části MVC.

DCGI

Zadání Svátky

- Doplňte kód controlleru svatky.php a view svatky.xhtml tak, aby aplikace měla následující funkcionalitu:
  1. kontrola správného vyplnění datumu (datum musí existovat, správný tvar)
  2. při chybném vyplnění datumu zobrazte chybovou hlášku
  3. při prvním načtení formuláře hlášku nezobrazujte
  4. při chybě vyplňte formulářová pole tak, aby obsahovala uživatelem vyplněné hodnoty

DCGI

Zadání Články

- Mějme články načtené z DB (zajišťuje model). Datová struktura je dvourozměrné pole.
  1. Vypište seznam článků ve formě Titulek, Autor, Náhled textu.
  2. Výpis formátujte tak, že liché řádky budou mít jinou barvu pozadí než sudé.
  3. Zapněte cache takovou, že články se budou aktualizovat jen jednou za 1 minutu.
     - Dejte pozor na to, aby se nenačítaly články z modelu, pokud to není nutné
  4. V GET parametru isadmin předávejte hodnotu 1 nebo 0 podle toho, zda daný uživatel má či nemá adminová práva. Administrátoři vidí vždy aktuální stav článků (bez cache), ostatní vidí články se cache 1 minuta.

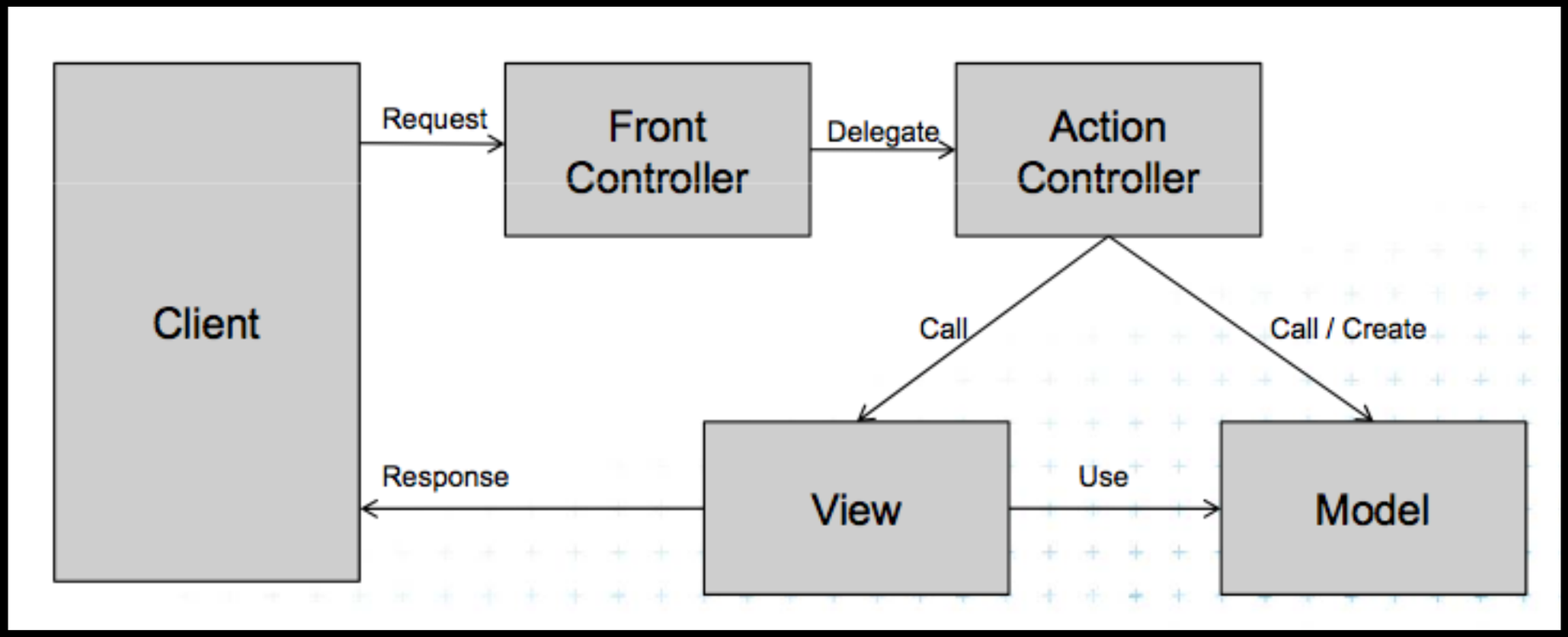
Opakování

- Jak vypadá varianta MVC Front Controler?

DCGI

MVC

Varianta Front Controller



- Jeden Controller pro všechny stránky
- Front Controller udělá globální práci a deleguje dotaz konkrétní implementaci (action controller)

DCGI

MVC
Úkol 1

- Přesměrování všech dotazů na jeden skript

```
RewriteEngine on

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule ^(.*)$ index.php?rt=$1 [L,QSA]
```

- logika: V URL bude zakódováno jméno action controlleru a v rámci tohoto controlleru také jméno metody, kterou v rámci daného ActionControlleru zavoláme.

DCGI

MVC
Úkol 2

- Připravte následující strukturu projektu
  - /classes
  - /controllers
  - /views
  - /index.php
  - /.htaccess
  - /bootstrap.php

DCGI

MVC

Úkol 3 - založení routeru

- implementace základní logiky pro předávání řízení.
- Tuto logiku implementujte v třídě Router (soubor classes/router.php).
- Kostra třídy viz Edux

DCGI

MVC

Úkol 4 - implementace metody route()

- Implementujte metodu route() tridy Controller.
- Kostra viz Edux

DCGI

MVC

Úkol 5 - implementace ActionControlleru

- Nyní umíte zavolat potřebné controllery a jejich metody.
- Struktura každého controlleru je naznačena na Eduxu
- Imlementujte jeden action controller s názvem indexController v souboru /controllers/index.php který obslouží všechna volání. Přidejte potřebné metody podle vlastního uvážení, které obslouží volání jiné než defaultní akce.
- Pro případ volání controlleru se jménem, které nemá implementaci vložte do frameworku také controller s názvem error404Controller v souboru /controllers/error404.php.

DCGI

MVC

Úkol 6 - implementace view

- View jsme zatím neimplementovali. Zavolání view je starostí jaždého jednotlivého ActionControlleru.
- Jaké máme požadavky na třídu implementující view:
  1. ke view je možné přiřadit libovolná data ve formě KLIC - HODNOTA
  2. view má metodu show(„nazev_view“), která zobrazí soubor umístěný v adresáři views /views/nazev_view.php
  3. pro soubor „nazev_view“ před jeho zavoláním připraví všechna data přiřazená jak klíč-hodnota tak, aby byla viditělná jako globální proměnná.
- Implementujte třídu Template v souboru /classes/ template.php , která zajistí předchozí funkcionalitu

DCGI

MVC

Úkol 7 - orchestrace

- Nyní máte implementovány všechny komponenty MVC. Vraťte se k front controlleru, tedy k index.php.
- Můžeme využít registry k tomu, abychom předávali informace mezi komponentami MVC a to včetně view.
- Ukázka výsledného kódu viz Edux.

DCGI

DEPARTMENT OF COMPUTER GRAPHICS AND INTERACTION

Q&A

macikmir@fel.cvut.cz