



# Web karkasas (Web framework)

# Kas yra karkasas?

- Karkasas (Framework) yra programinis kodas turintis:
  - Suplanuotą veiksmų seką (control flow)
  - Sukurtą pagrindinį funkcionalumą
  - Dokumentaciją kaip jį panaudoti
- Pavyzdžiai:
  - JavaEE, CakePHP, RubyOnRails, .NET framework, CodeIgniter, ...

# Kokia karkaso nauda?

- Logiškai viskas sutvarkyta →
  - Aišku kur kas yra “padėta”
  - Mažiau klaidų programuojant
  - Greičiau klaidos ištaisomos
  - Didesnis saugumas
  - Gražu pažiūrėti į kodą ☺
  
- Kas nors sako: Tai – nereikalingas dalykas
  - Tai gal programuokime Assembler, kam ta JAVA

# Pavyzdžiai



# Pavyzdžiai (2)



# PHP karkasai

- CodeIgniter
- CakePHP
- Symfony
- ...
  
- PHP frameworks
  - PHP karkasų palyginimas, bei trumpas kiekvieno aprašymas
- 20 Great PHP frameworks for developer
  - Blog'o straipsnis

# Nemaišyti su TVS!

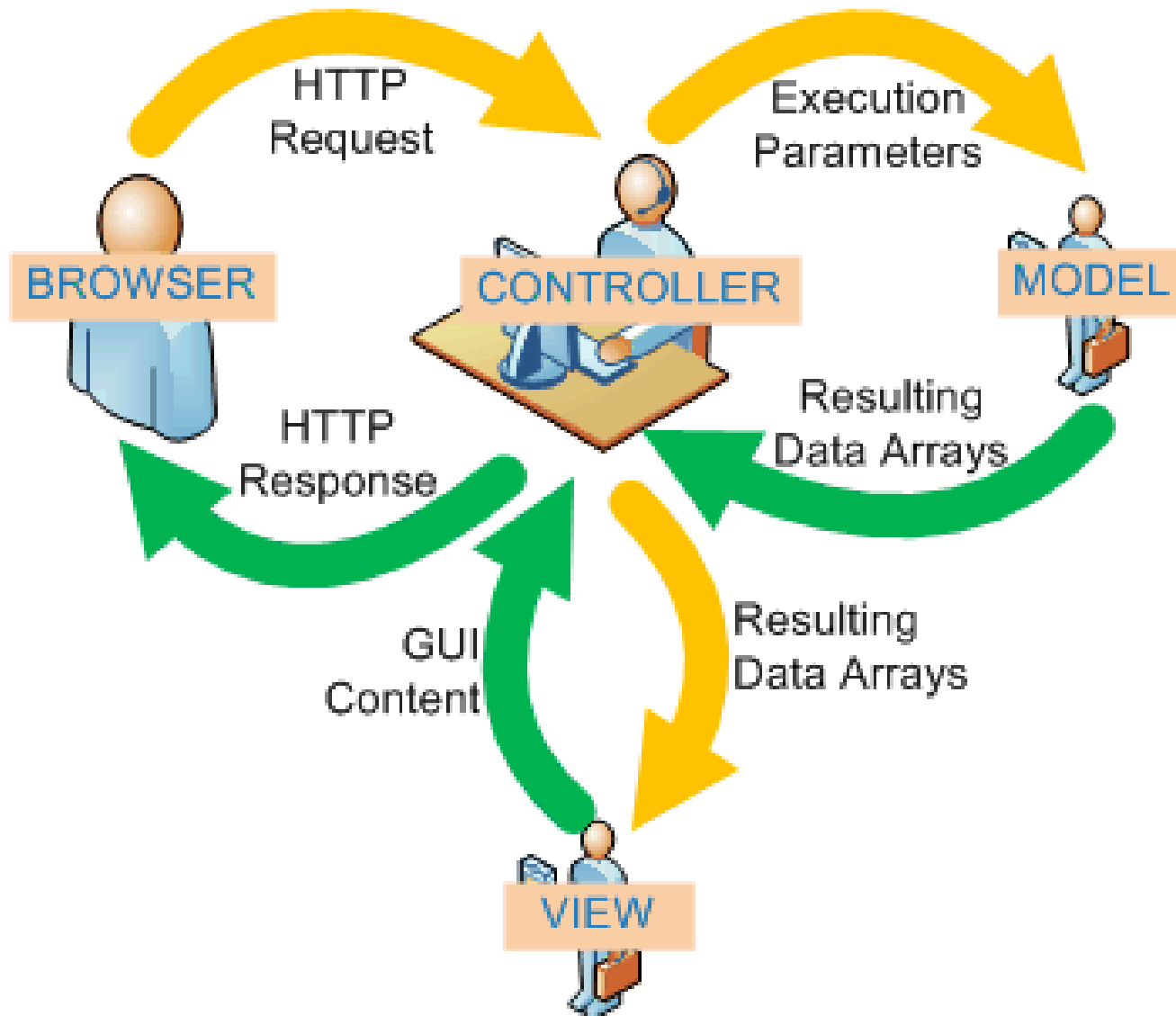
- Karkasas tai ne TVS (turinio valdymo sistema)
- TVS yra:
  - Pilnai programiškai veikiantis produktas
  - Skirtas kurti svetaines žmonėms, nemokantiems programuoti

- 3-tier (aukštų/pakopų)
  - Presentation, Bussines Logic, Data
  - Pvz.: .NET 2.0 framework
- MVC – Model-View-Controller
  - CodeIgniter, CakePHP, Zend fram.
  - WPF (subsystem .NET 3.0),
- Module based
  - ...
- Kelios architektūros kartu (dažniausiai)



# MVC (Model-View-Controller)

[1]



# 3-tier (lygių)

[2]

## Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

>GET SALES  
TOTAL

>GET SALES  
TOTAL  
4 TOTAL SALES

## Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

GET LIST OF ALL  
SALES MADE  
LAST YEAR

ADD ALL SALES  
TOGETHER

QUERY

SALE 1  
SALE 2  
SALE 3  
SALE 4

## Data tier

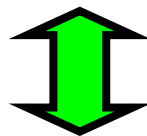
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Database

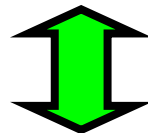
Storage

# 3-tier (lygių) (2)

**Visualization / Presentation**



**Program / Logic tier / Business**



**Data model / Data tier**

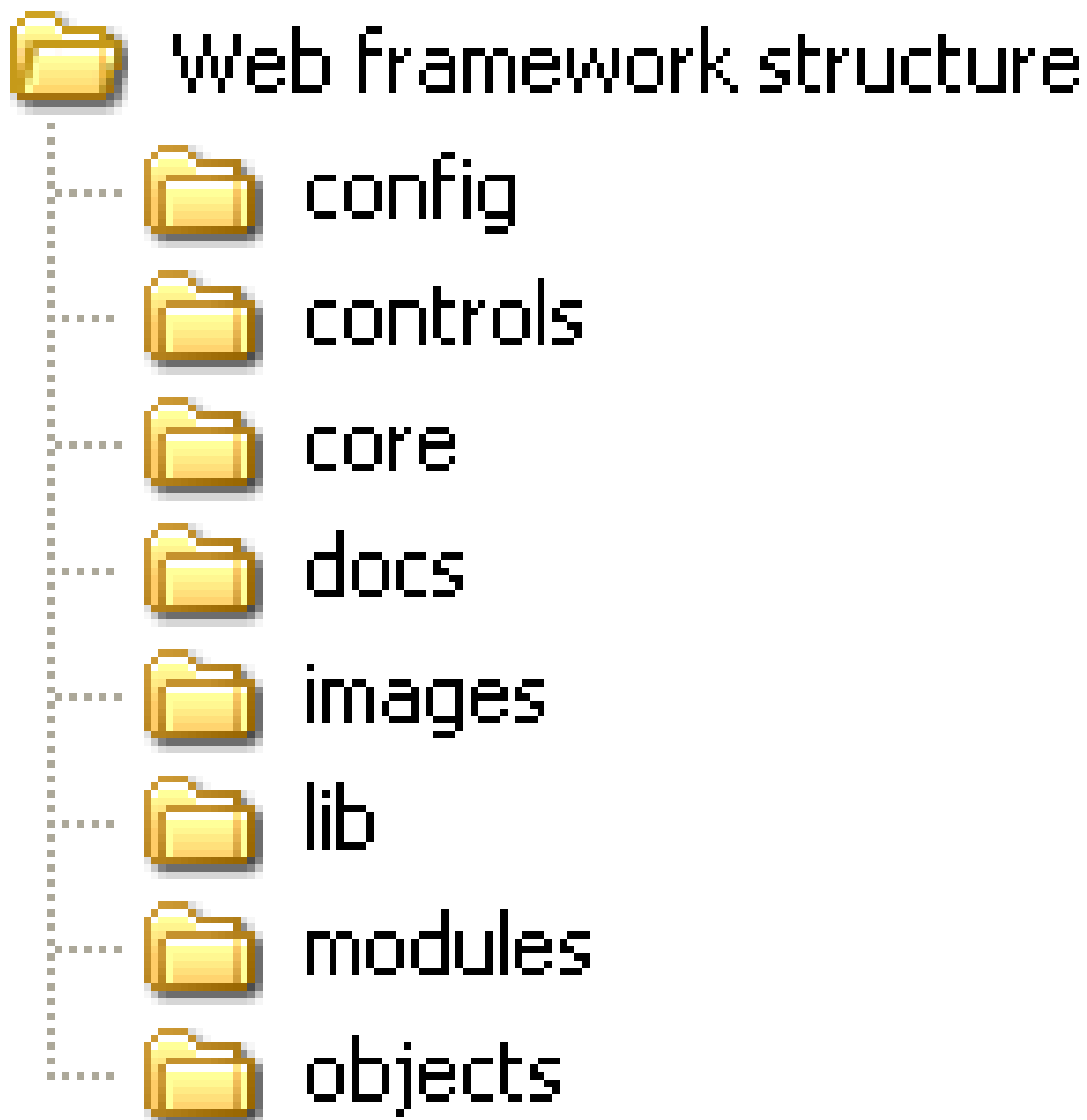
# 3-tier (lygių/pakopų)

- Presentation
  - Views
  - Templates
  - Controls
- Business logic – Core
  - Main controller
  - Session
  - Cookie
  - Error handling
- Data
  - DB/XML/data manager

# Modulinis Karkasas

- Toliau skaidrėse pateikiamas **modulinio** karkaso aprašymas
  - Katalogų struktūra
  - Veikimo principai

# Struktūra





## Web framework structure



config

// konfiguracioniai failai



controls

// vizualūs komponentai



core

// pagrindiniai failai



docs

// dokumentacija



images

// media (dėl saugumo)



lib

// bendros bibliotekos



modules

// moduliai



objects

// objektai

- Pagrindinis valdymo failas
- Turi:
  - Pakrauti visus/reikiamus modulius/failus (include)
  - Pakrauti konfigūraciją
  - Saugumo procedūros
  - Atlikti visas reikalingas pirmines funkcijas (preprocess)
    - Iškviesti modulių pirmines funkcijas (preprocess)
    - ...
  - Iškviesti reikiamą modulį



# Automatinis sudėjimas

- Core sudedami manualy
- Module failai turi:
  - Įsidėti save į globalų masyvą
  - Didesnis saugumas → Galima sudėti į masyvą modulio vardą ir iškvietimo žodį
- Parametras adrese yra modulio pavadinimas/iškvietimo žodis
- Index tikrina parametrus – jei nėra → į default puslapį

# /core principai

- Čia sudėti visi bendri moduliai
- Turi būti nekeičiamas bet kuriai svetainei
  - Neturi būti jokių spec. nustatymų ar kodo konkrečiai svetainei

- Managers:
  - Database
  - Language
  - Session
  - Cookie
  - Objects
  - ...

- Konfiguracijos failai
- Dažniausiai:
  - config.php
  - config.js
  - language.xml (jei yra LanguageManager)
  - objects.xml (objektų duomenys)
- Gali būti “Language/” ir ten <modulio vardas>.xml

# Multi Language

```
<language>  
  <system> main </system>  
  <lt> pagrindinis </lt>  
  <en> Main </en>  
</language>
```

```
LanguageManager::print('main');  
LanguageManager::print('UserModule','main');
```

# /modules principai

- visi moduliai reikalingi konkrečiai web svetainei
- Moduliai turi bendrą struktūros principą
  - visi turi klasę standartiniu pavadinimu
    - <vardas>Module
  - klasė turi “main()” funkciją
  - apsaugą nuo tiesioginio priejimo
  - įsidėjimą į modulių masyvą

# /modules (specialūs)

- Template Module
- User Module

# Modulių prasmė

- Kas tas modulis?
- Aiškus funkcionalumo išskirstymas
- Kiekvienas modulis žino tai, su kuo dirba
  - Pvz. Session, User, Gallery, ...
- Lengvas klaidų taisymas
  - + Taisant vieną iš modulių kitur turi klaidų neatsirasti



# Parametrų perdavimas

- `index.php?a=1&b=2`
  - + Labai aiškus, jokių dviprasmybių
  - – Nemėgsta paieškos varikliai
- `index.php/a/1/b/2`
- `index.php/1/2`
  - – Reikalingas:
    - Papildomas “parsing” žingsnis
    - Dokumentacija kas kur yra
  - + Paieškos varikliams OK
  - + Daugumą parametrų reikia žinot tik vienam moduliui

# User module

- Labai specifinis
- Susijęs su teisiu skirstymu
  - Turi būti sukurtas mechanizmas, kaip sąveikaus įdiegti ir nauji vartotojai su naujais moduliais, bei kaip bus administruojamos teisės
- admin loginas

# Sąvybės (Features)

---

- BreadCrumb
- Paieškos sistema
- Meniu modulis
- Cronjob sistema

# Praktiniai patarimai

- Statinės klasės
  - Konfiguracija
  - Database, Session, Cookie
  - ...
- Klasės kurias pasiekiamo iš kitų klasių, bet nenorime kiekvieną kartą kurti klasę, kad panaudotume metodą
- PHP objektų gyvavimas:
  - Objektai gyvuoja neilgai, todėl statinių klasių panaudojimas yra geriau, nei 20 kartų kurti objektą dinamiškai

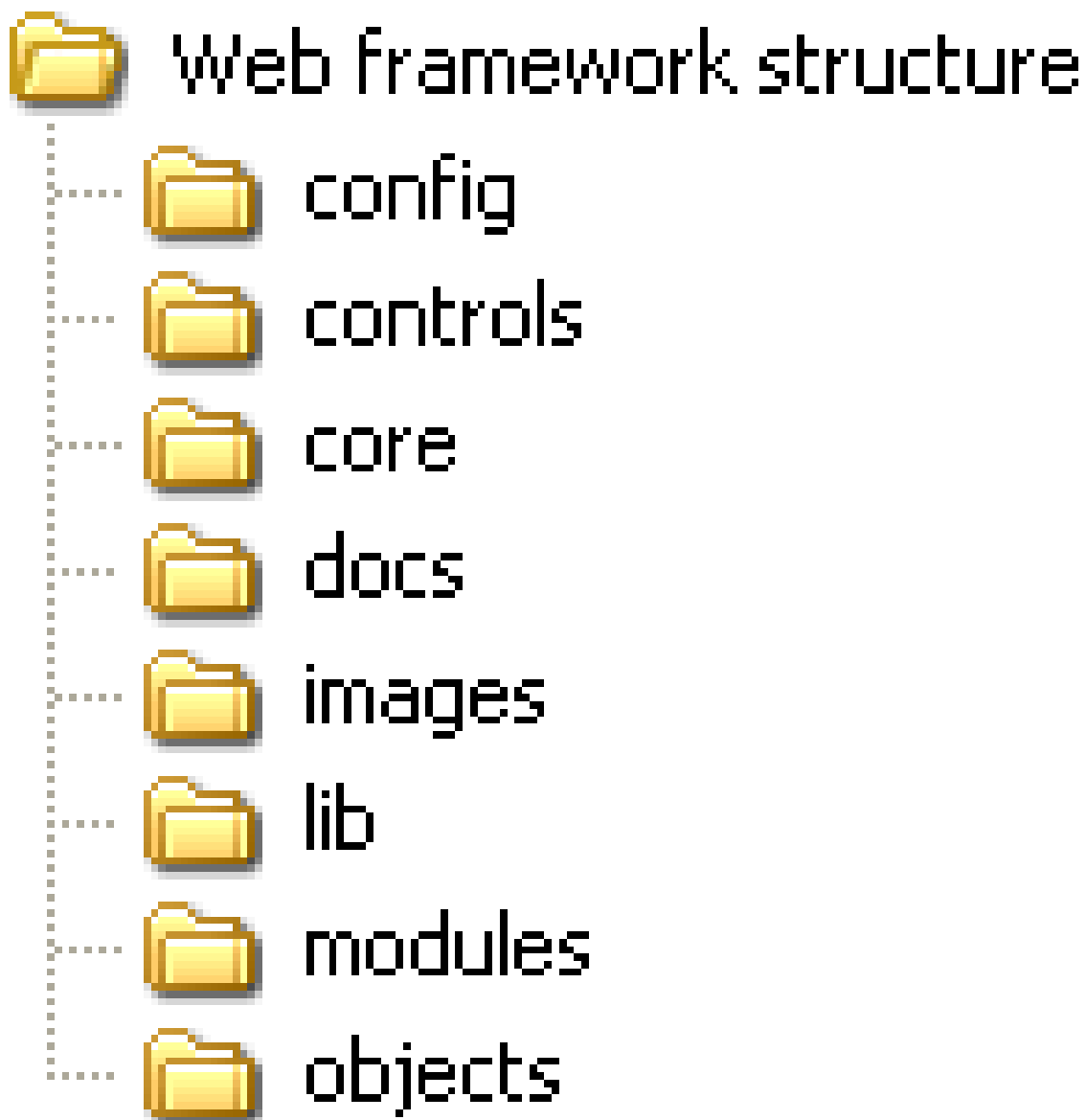
# Nuo ko pradėti?

- Susikurkite minimalią struktūrą
  - index.php
  - /modules/templateModule.php
  - /modules/mainModule.php
- Po to prilipdysite daugiau
- ...

# Failų sudėjimas ir apsauga

- Automatinis/manual sudėjimas
  - Automatinis reikalingas moduliams
  - Manual – branduoliui (core)
- Nuo tiesioginio pasiekimo apsauga
  - Konstantos apsibrėžimas

# Struktūra



# Literatura

- [1] ash.MVC <http://ash-mvc.org/website/framework/framework.html> (2008.11.06)
- [2] Wikipedia <http://en.wikipedia.org/wiki/3-tier> (2008.11.06)



# Pabaiga