Съвременно web прогрмаиране с PHP 2013

Protocols;
HTTP;
C-S architecture;

Владимир Алексиев

@phpfmi
@v_aleksiev
github.com/valeksiev

Start with a joke?

Днес ще:

- Уеднаквим понятията
- Разберем как работи интернет (или поне ще добием представа)
- Разлика между desktop и web среда

WEB?

- URI = URL + URN
- Ресурси
- HTTP - протокол за комуникация
- Client-Server архитектура

Ресурси?

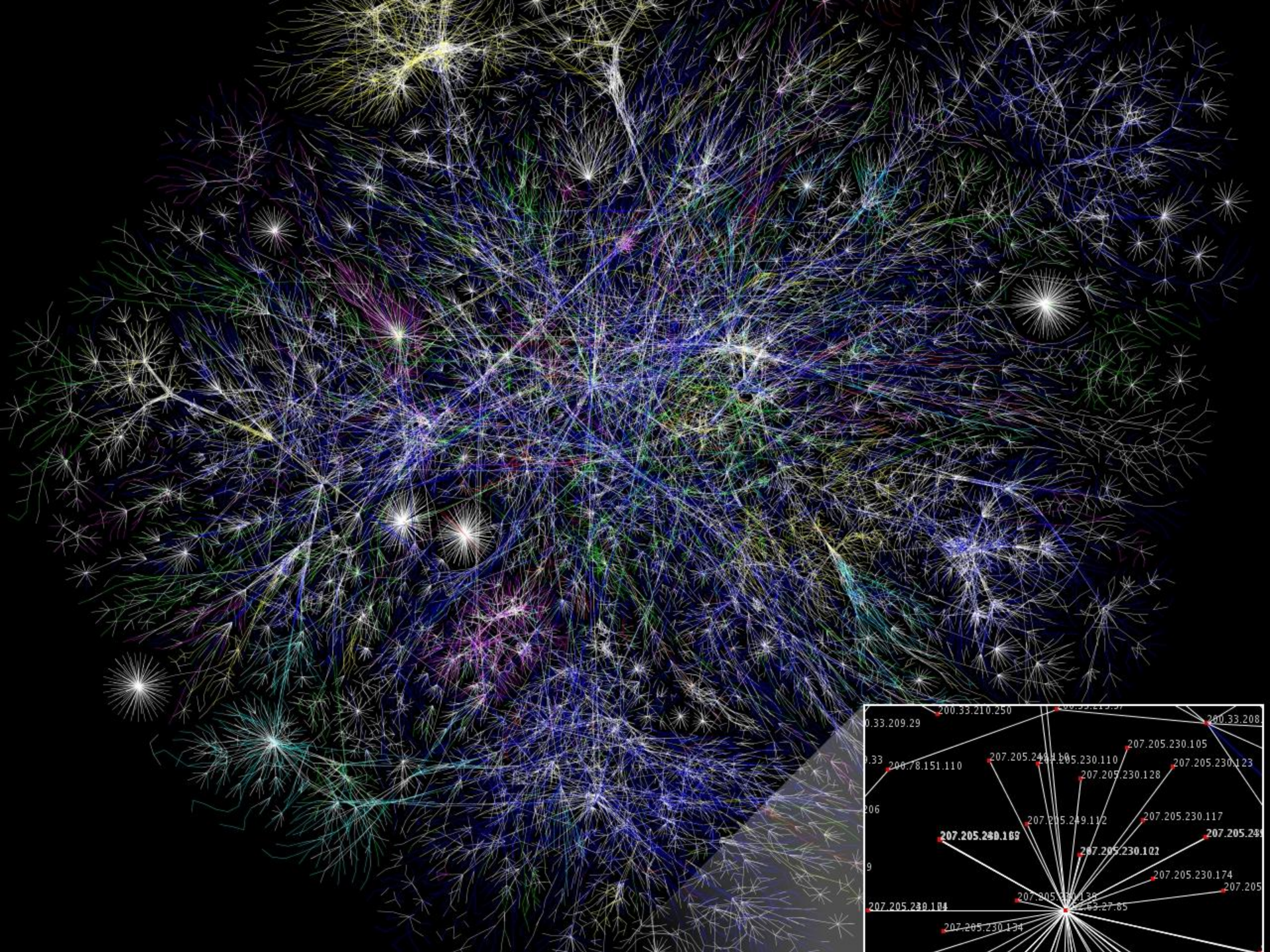
- Всичко, което виждаме в браузъра
  - HTML документи
  - Картинки {.png|.jpg|.gif|etc}
  - JavaScript файлвое
  - Видео/флаш файлове

```
$webPage = array($res1, $res2);

$webSite = array($page1, $page2);

$webApplication = new App(
   array(
      'use_database' => true,
      'implement_logic' => true,
   )
);
```

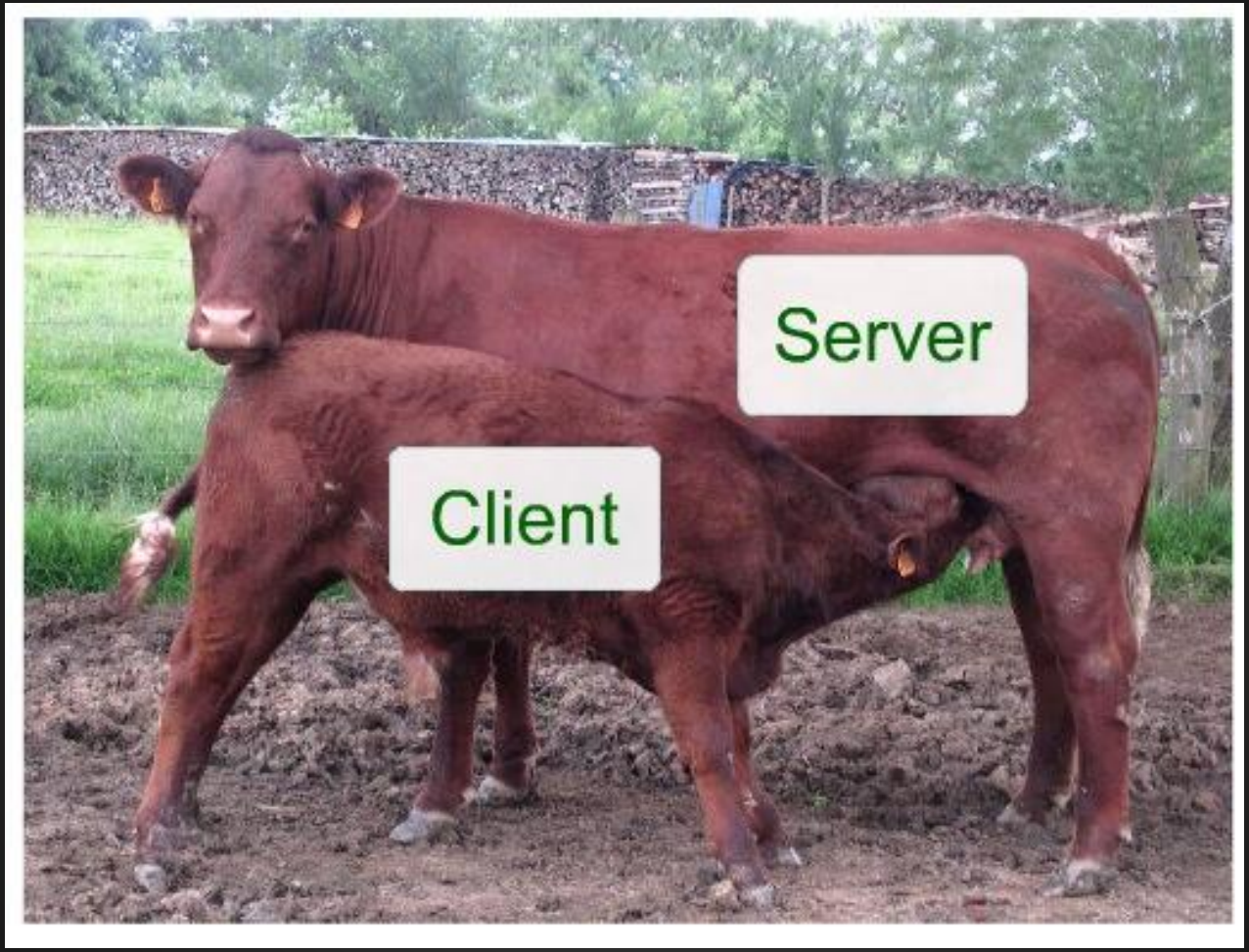
Проткол?
Кому е нужно!?

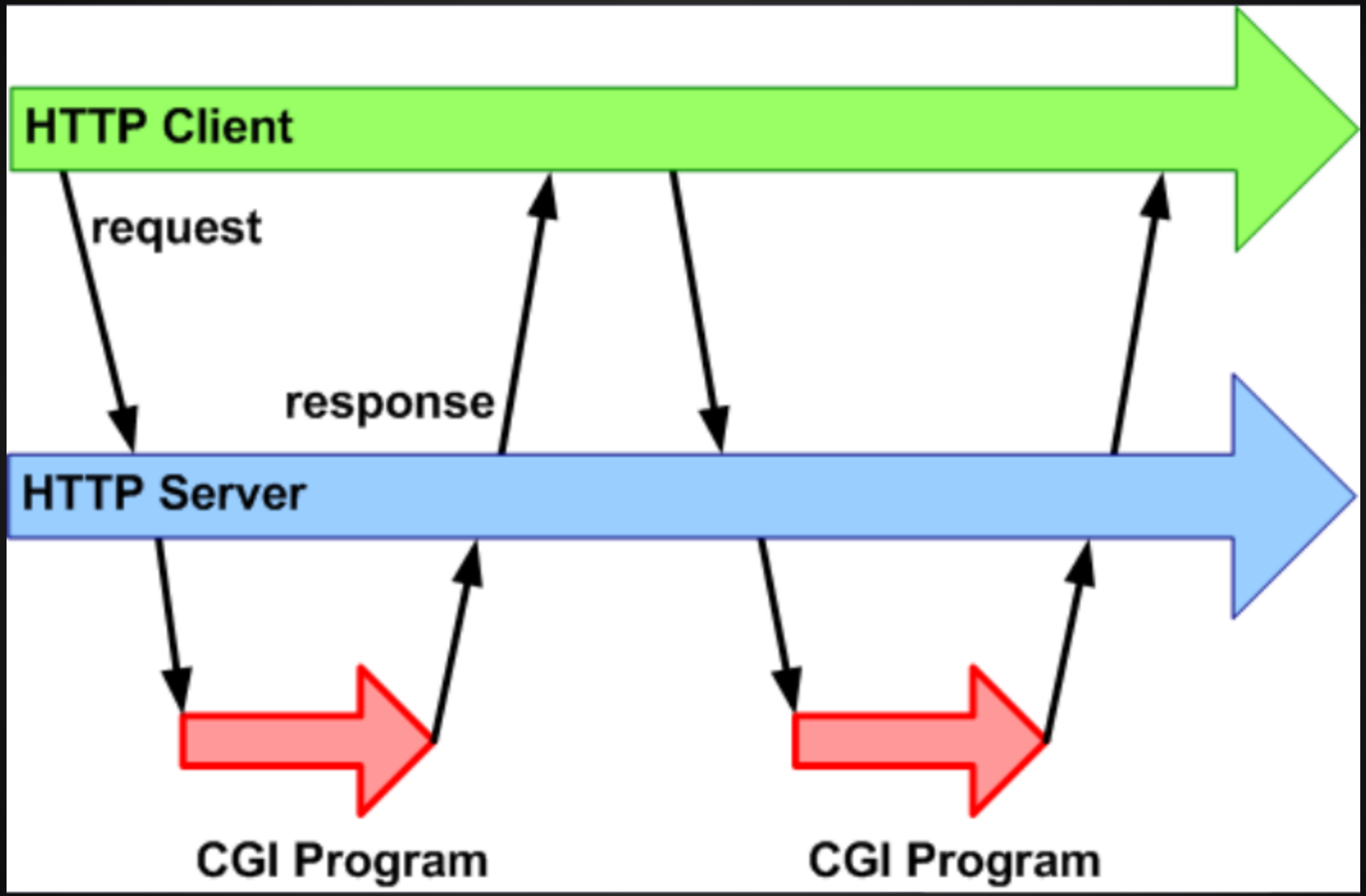
Client - Server

- Алтернативи - p2p
- Централизирана и широко застъпена в web
- 1 сървър, n клиенти

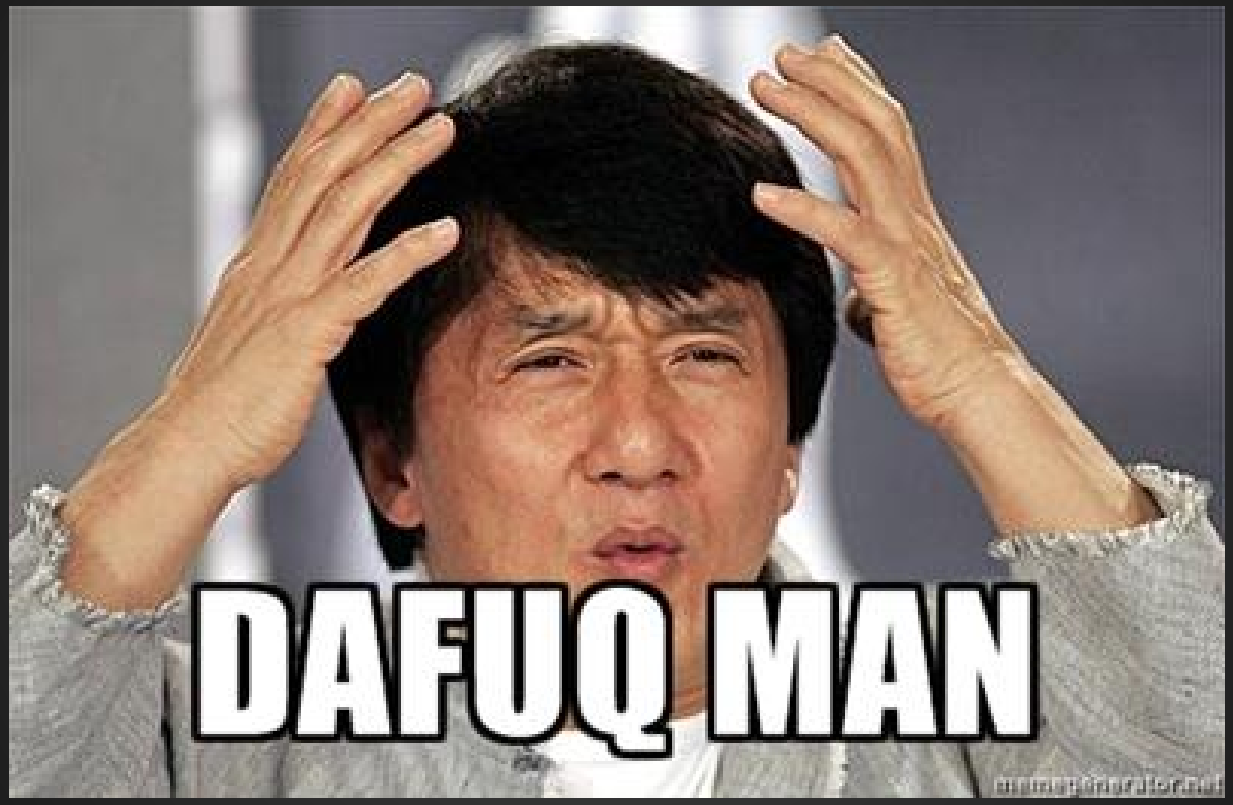

200.33.210.250
0.33.209.29
200.33.208
207.205.230.105
207.205.230.110
207.205.230.123
200.78.151.110
207.205.230.128
207.205.249.112
207.205.230.117
207.205.230.174
207.205.230.134

Формат и правила за комуникация между устройства.


Server
Client


HTTP Client
request
response
HTTP Server
CGI Program
CGI Program

CGI Program!?



Common Gateway Interface

- Страниците са статични!
- CGI прави магията

Client - Server

- HTTP (browser <-> apache)
- Email (gmail <-> sendmail)
- MySQL (phpMyAdmin <-> mysqld)
- FTP (FileZilla <-> vsftpd)
- SSH (Putty <-> SSH server)
- IRC
- WoW
- etc.

Сървър != Сървър

- Физически сървър

- Сървърен софтуер
  - има представка
    - HTTP Server
    - FTP Server
    - Mail Server

Къде живее HTTP?

- Internet protocol suite - TCP/IP
  - TCP
    - осигурява пристигането на пакетите(тея пък к'ви са!?)
    - TCP Ports
    - http://localhost:8080
  - IP
    - IP адрес
    - There is no place like 127.0.0.1

За любознателните

- OSI model
- TCP/IP model
- Vint Cerf
- Bob Kahn

facebook.com
<->
173.252.110.27

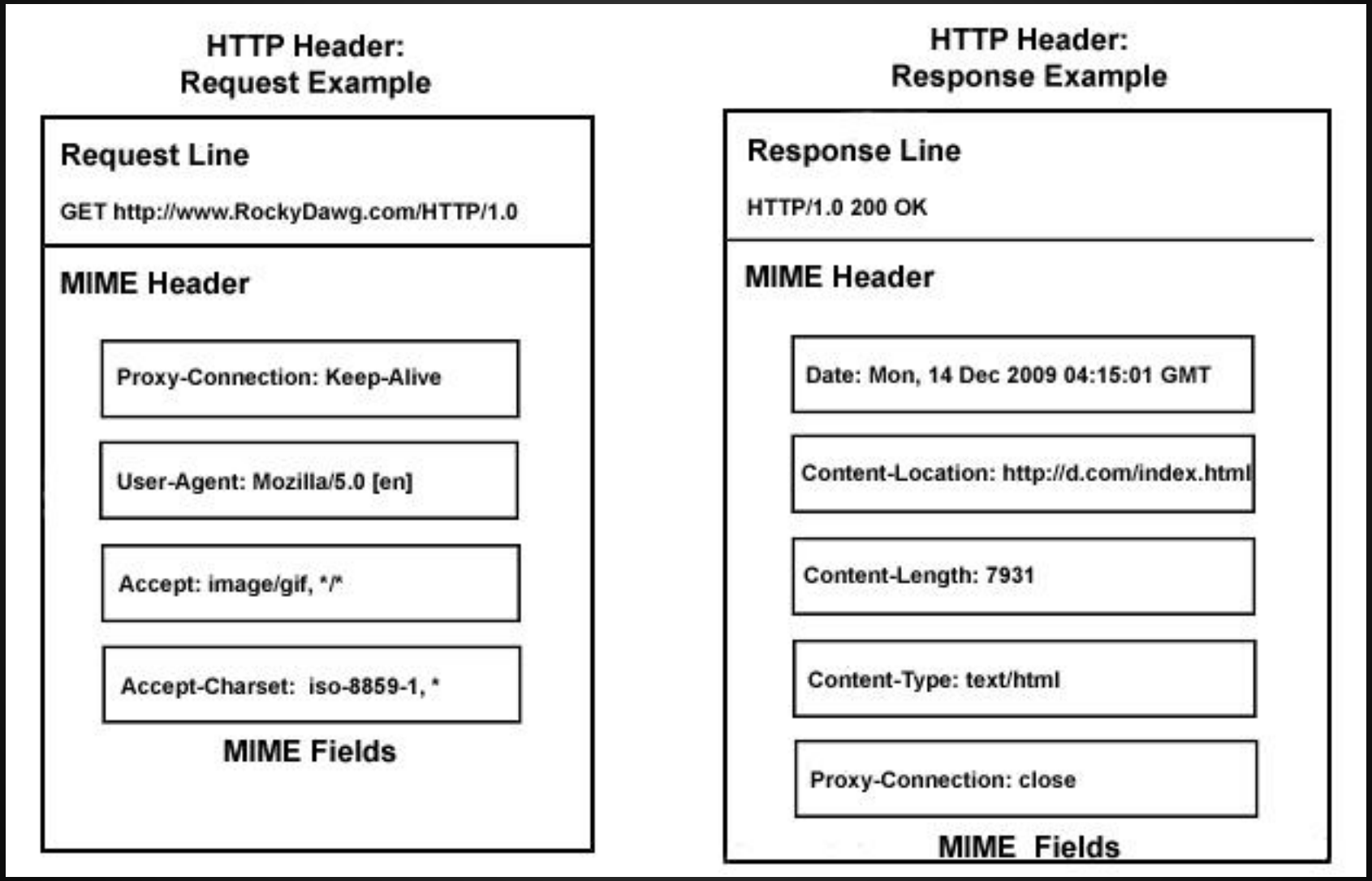
DNS

- От human readable име до IP адрес
- Browser-а първо се обръща към DNS сървър.
- Пример?

HTTP

- “Brain-dead simple protocol”
- Текстово-базиран протокол
- Request <-> Response по TCP конекция
- Основните му части са :
- HTTP Methods {GET, POST, PUT, DELETE, ..}
- HTTP Headers
- HTTP Request/Response body

Headers


HTTP Header:
Request Example
Request Line
GET http://www.RockyDawg.com/HTTP/1.0
MIME Header
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/5.0 [en]
Accept: image/gif, */*
Accept-Charset: iso-8859-1, *
MIME Fields
HTTP Header:
Response Example
Response Line
HTTP/1.0 200 OK
MIME Header
Date: Mon, 14 Dec 2009 04:15:01 GMT
Content-Location: http://d.com/index.html
Content-Length: 7931
Content-Type: text/html
Proxy-Connection: close
MIME Fields

A flash from the past

http://internal.network.address:8082/index.php?token=U05JQ0tFUlM=&fname=Lamond&lname=Cranston&redirect=/

Примери

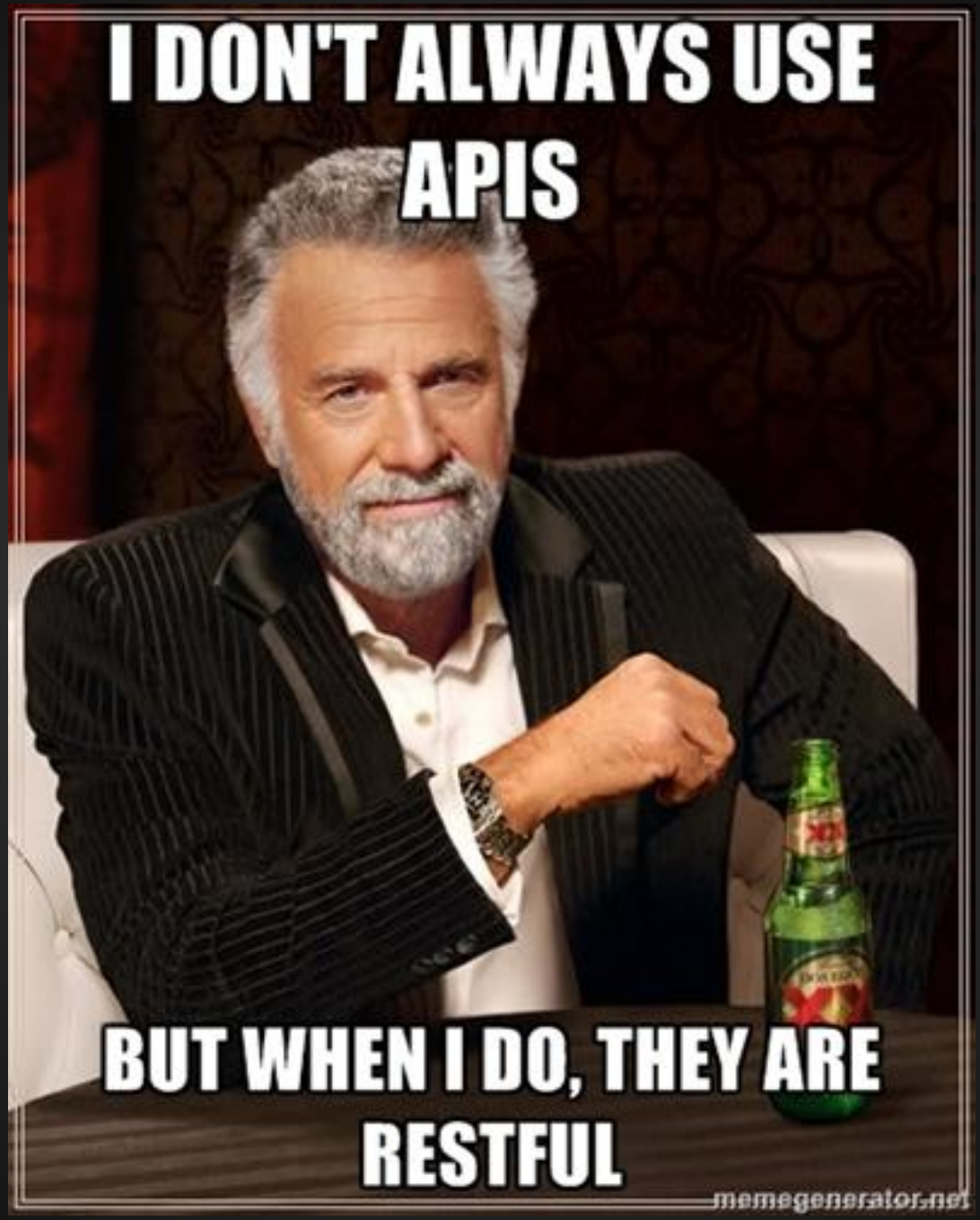
- netcat
- curl

Клиенте и Сървари

- HTTP Clients
  - Le Browser!
  - cURL
  - lynx
  - wget
  - AJAX
  - etc.
- HTTP Servers
  - Apache
  - NodeJS
  - Nginx
  - IIS
  - etc.

Методи

- GET /resource/
  - Give me everything you have about this URL
- POST /resource/
  - I am sending some information in my request body for this URL
- PUT /resource/
  - I am sending changes about this resource in my request body
- DELETE /resource/
  - Delete everything you have about this URL
- Etc.(There are others, less used)


I DON'T ALWAYS USE APIS
BUT WHEN I DO, THEY ARE RESTFUL
memegenerator.net

Status codes

- Трицифрено число, което показва дали дадена заявка е успешна или има проблем

- **1xx – Information**
- **2xx – Success**
- **3xx – Redirection**
- **4xx – Client Error**
- **5xx – Server Error**

Малко примери в Хромето

И още...

- HTTP е stateless протокол.
- Всеки request е независим и не е свързан с предният
- **Плюсовете:**
  - Улеснява протоколната имплементация
- **Минуси:**
  - Как тогава работят всички страници в web ?
- HTTP Server-ите имплементират session management


CutestPage.com

Въпроси?