Course:
PHP from scratch

by Sergey Podgornyy

Introduction and basics of networking

W E B
ACADEMY
PROGRAMMING
COURSES

About me

Sergey Podgornyy

Full-Stack Web Developer

WebbyLab

and

ignite
software outsourcing

zend CERTIFIED PHP ENGINEER

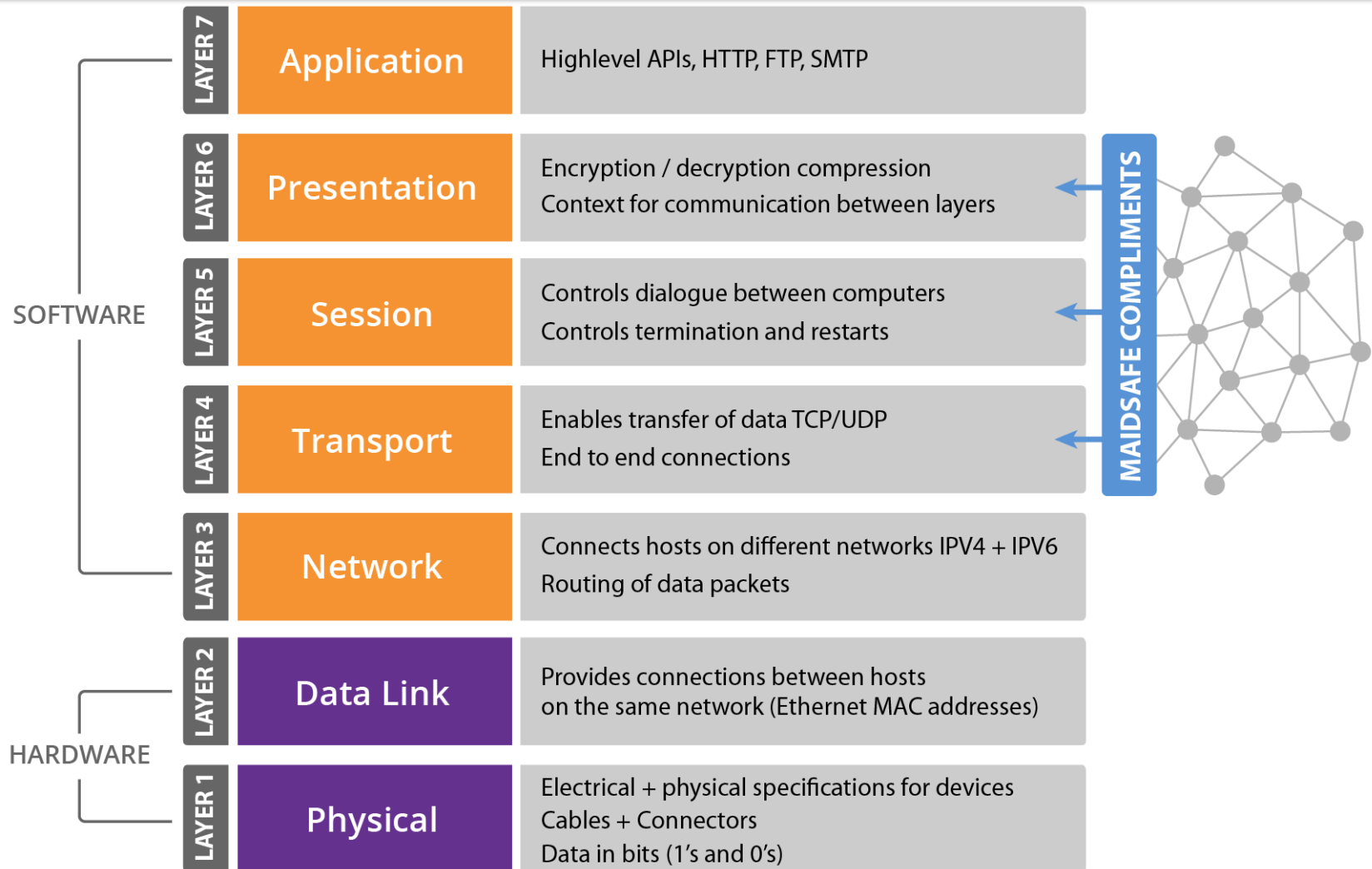Linux Professional Institute LPIC-1

Agenda

1. Internet, network basics. How web browsers work?
2. Client-Server Architecture. Local and remote servers
3. DNS, IP, ports and Sockets
4. Hostings
5. Useful utilities: `ipconfig`, `ping`, `tracert`, `netstat`
6. HTTP basics. Headers, methods, responses. Most common response statuses
7. What do we need for work?
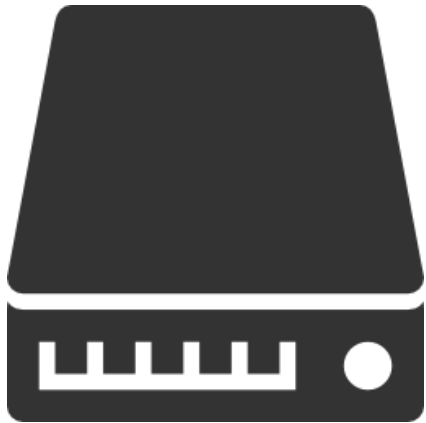
Client–Server model
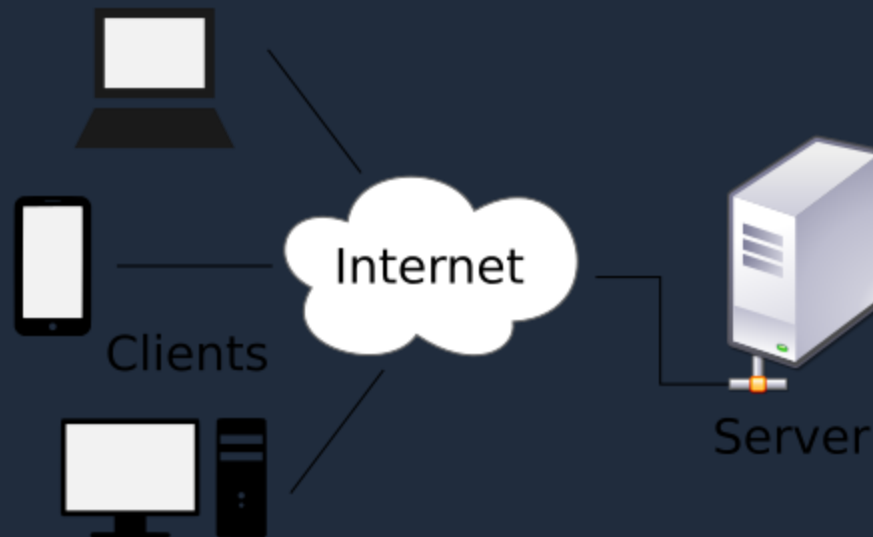
The *Client-Server* characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services
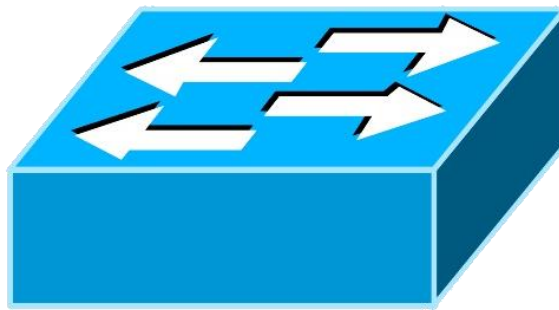
OSI model

SOFTWARE	LAYER 7	**Application**	Highlevel APIs, HTTP, FTP, SMTP
	LAYER 6	**Presentation**	Encryption / decryption compression Context for communication between layers
	LAYER 5	**Session**	Controls dialogue between computers Controls termination and restarts
	LAYER 4	**Transport**	Enables transfer of data TCP/UDP End to end connections
	LAYER 3	**Network**	Connects hosts on different networks IPV4 + IPV6 Routing of data packets
HARDWARE	LAYER 2	**Data Link**	Provides connections between hosts on the same network (Ethernet MAC addresses)
	LAYER 1	**Physical**	Electrical + physical specifications for devices Cables + Connectors Data in bits (1's and 0's)

MAIDSAFE COMPLIMENTS

Low Level of OSI model

Hub

physical layer

Switcher

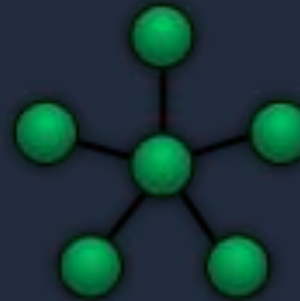data link layer

Router

network layer

Network topologies

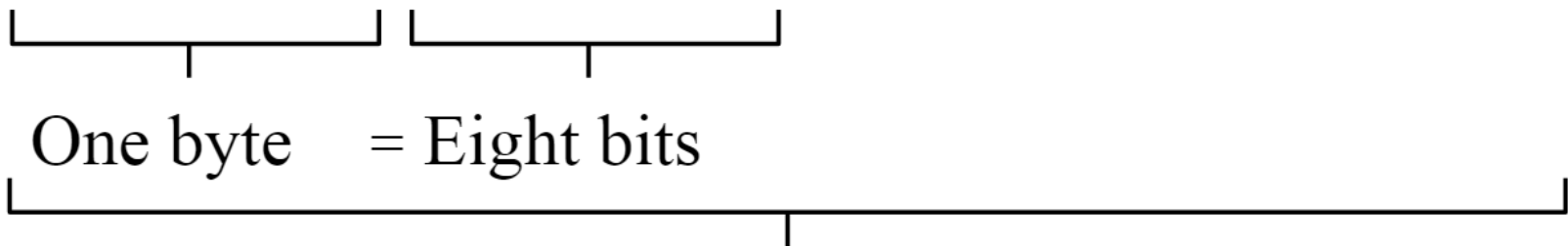

Ring

Mesh

Star

Fully Connected

Line

Tree

Bus

IP (IPv4)

An IPv4 address   (dotted-decimal notation)

172 . **16** . **254** . **1**

↓   ↓   ↓   ↓

10101100 .00010000 .11111110 .00000001

One byte   = Eight bits

Thirty-two bits (4 x 8), or 4 bytes

Subnet mask

|		
IP address	11000000 10101000 0000000**1 00000010**	(192.168.1.2)
Subnet mask	11111111 11111111 1111111**0** 00000000	(255.255.254.0)
Network address	11000000 10101000 0000000**0** 00000000	(192.168.0.0)

Private network

| RFC1918 name	IP address range	number of addresses	largest CIDR block (subnet mask)	host id size	mask bits
24-bit block	10.0.0.0 - 10.255.255.255	16,777,216	10.0.0.0/8 (255.0.0.0)	24 bits	8 bits
20-bit block	172.16.0.0 - 172.31.255.255	1,048,576	172.16.0.0/12 (255.240.0.0)	20 bits	12 bits
16-bit block	192.168.0.0 - 192.168.255.255	65,536	192.168.0.0/16 (255.255.0.0)	16 bits	16 bits

DNS

Domain Name System
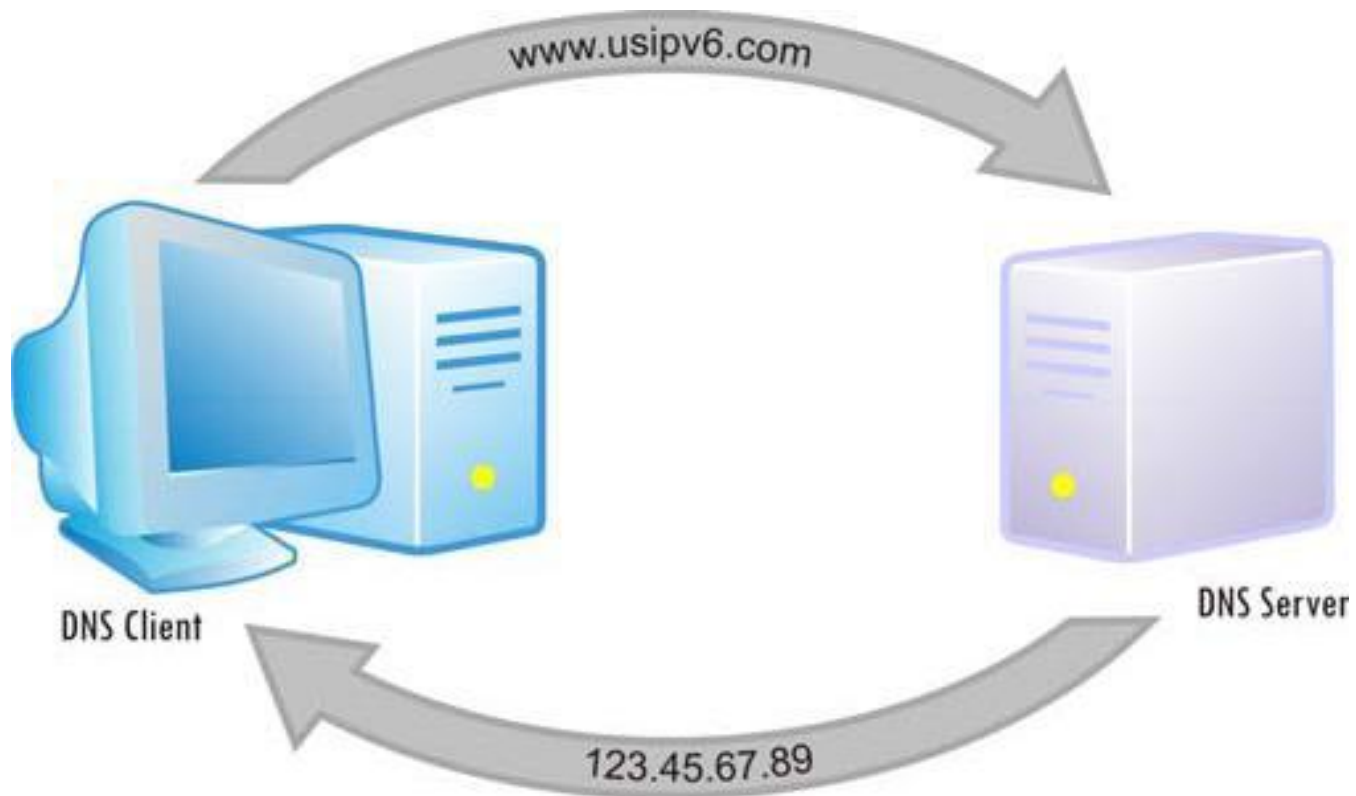
Protocols

- **HTTP** – Hyper Text Transfer Protocol

- **TCP** - Transmission Control Protocol

- **UDP** - User Datagram Protocol

- **IP** - Internet Protocol

http://

Domains
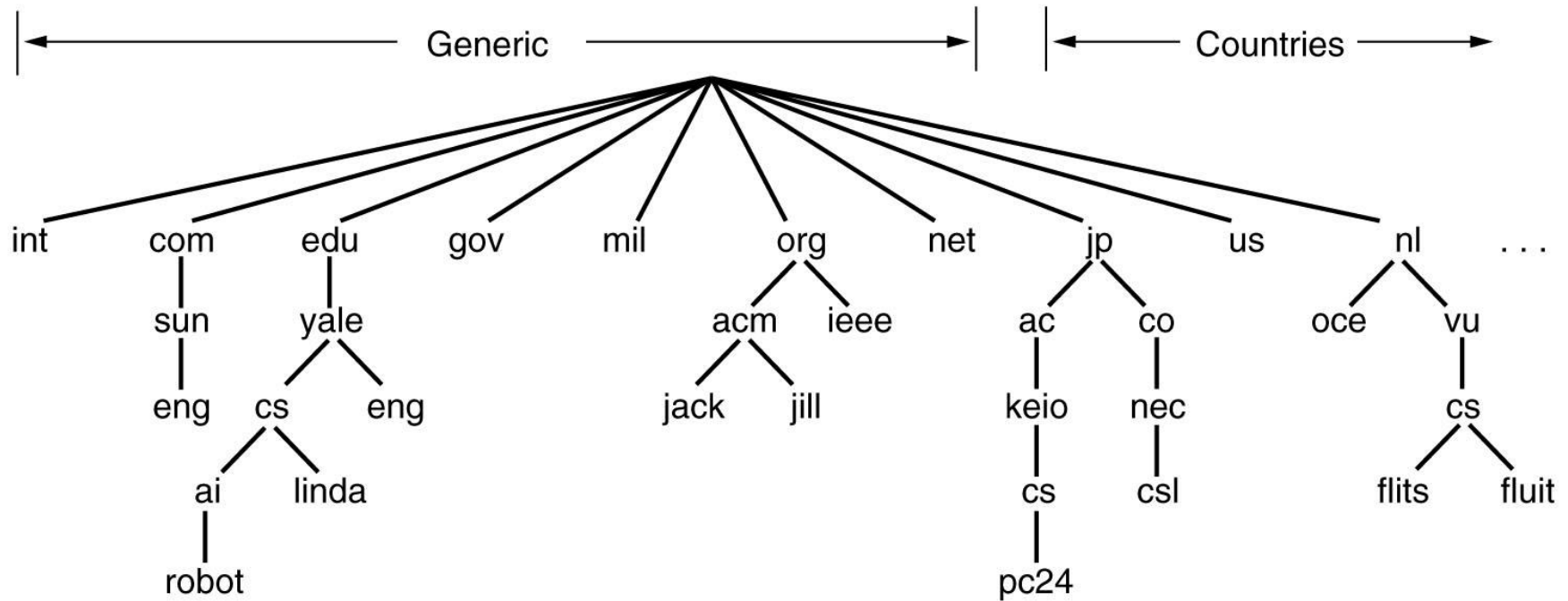
Host & Hosting

Web Server

A web server is a computer system that processes requests via **HTTP**, the basic **network protocol** used to distribute information on the **World Wide Web**
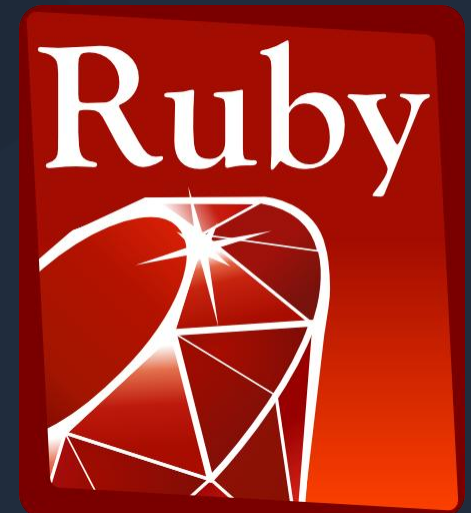
Server-Side languages

Utilities

- **`ipconfig`**

console application that displays all current TCP/IP network configuration values

- **`ping`**

utility used to test the reachability of a host on an Internet Protocol (IP) network

- **`tracert (traceroute)`**

tool for displaying the route (path) and measuring transit delays of packets across an Internet Protocol (IP) network

- **`netstat`**

network utility tool that displays network connections for the TCP, routing tables, and a number of network interface and network protocol statistics

HTTP Request

Every HTTP Request consist:

- Request string with method and HTTP version specified

- Request Headers

- An empty line

- Request Body

<Method> <URI> HTTP / <version>

GET http://example.com/index.html HTTP/1.1

Request methods

HTTP defines methods to indicate the desired action to be performed on the identified resource

- **GET**
- **HEAD**
- **POST**
- **PUT**
- **DELETE**
- **TRACE**
- **OPTIONS**
- **CONNECT**
- **PATCH**

Request

```
GET /wiki/HTTP HTTP/1.1
Host: uk.wikipedia.org
User-Agent: firefox/5.0 (Linux; Debian 5.0.8;
en-US; rv:1.8.1.7) Gecko/20070914
Firefox/2.0.0.7
Connection: close

Request body
```

Response statuses

- **1xx** Informational

- **2xx** Success

- **3xx** Redirection

- **4xx** Client Error

- **5xx** Server Error

Most popular statuses

- **200 OK**

- **301 Moved Permanently**

- **400 Bad Request**
- **401 Unauthorized**
- **403 Forbidden**
- **404 Not Found**
- **405 Method Not Allowed**
- **408 Request Timeout**

- **500 Internal Server Error**
- **502 Bad Gateway**
- **504 Gateway Timeout**

HTTP Response

The response message consists of the following:

- A status line which includes the status code and reason message

- Response header fields

- An empty line

- An optional message body

Most common Response fields

- **Server** - A name for the server

  `Server: Apache/2.4.1 (Unix)`

- **Last-Modified** - The last modified date for the requested object

  `Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT`

- **Content-Length** - The length of the response body in octets

  `Content-Length: 348`

- **Content-Type** - The MIME type of this content

  `Content-Type: text/html; charset=utf-8`

- **Date** - The date and time that the message was sent

  `Date: Tue, 15 Nov 1994 08:12:31 GMT`

Start built-in server


```
php -S  localhost:8000 [ -t public/ ]
```

IDE vs text editor

Development environment

Highly not recommended

Useful resources

- https://habrahabr.ru/post/307252/

- https://en.wikipedia.org/wiki/OSI_model

- https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods

- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

- https://en.wikipedia.org/wiki/List_of_HTTP_header_fields#Request_fields

Thanks for your attention

Q & A