



# Базовые компоненты интернет- технологий

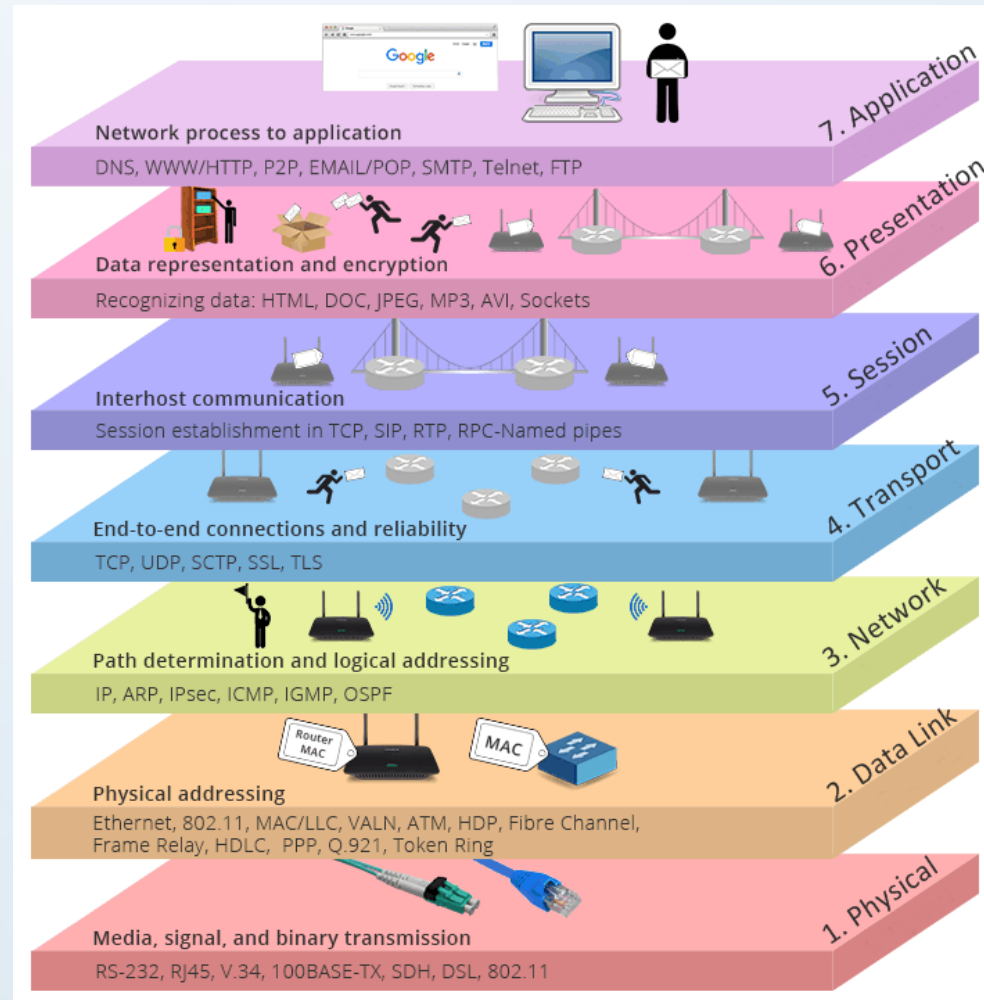


ИУ-5, бакалавриат, 3 семестр

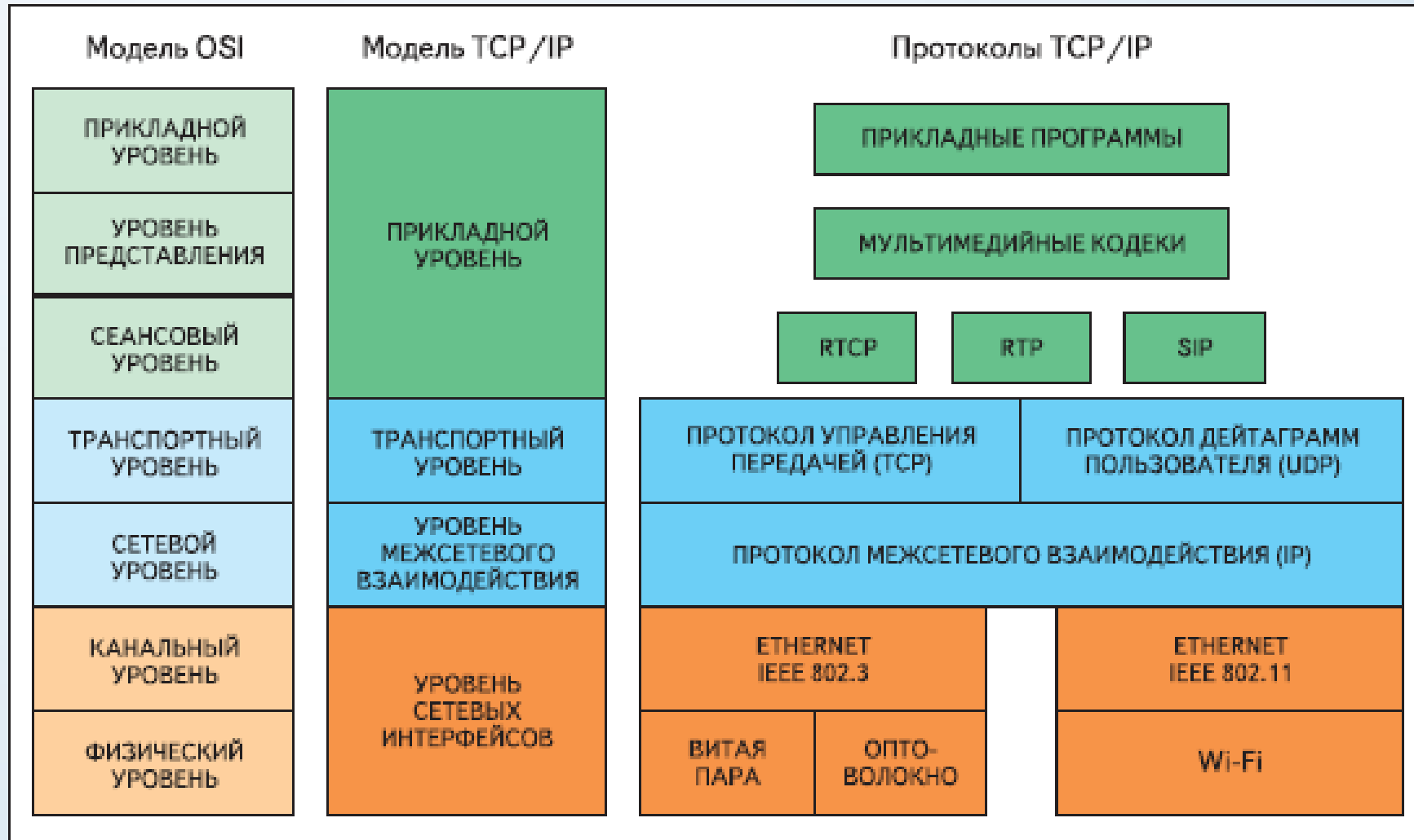


# **Введение в разработку веб- приложений на Python с использованием веб-фреймворка Flask**

# Сетевая модель OSI

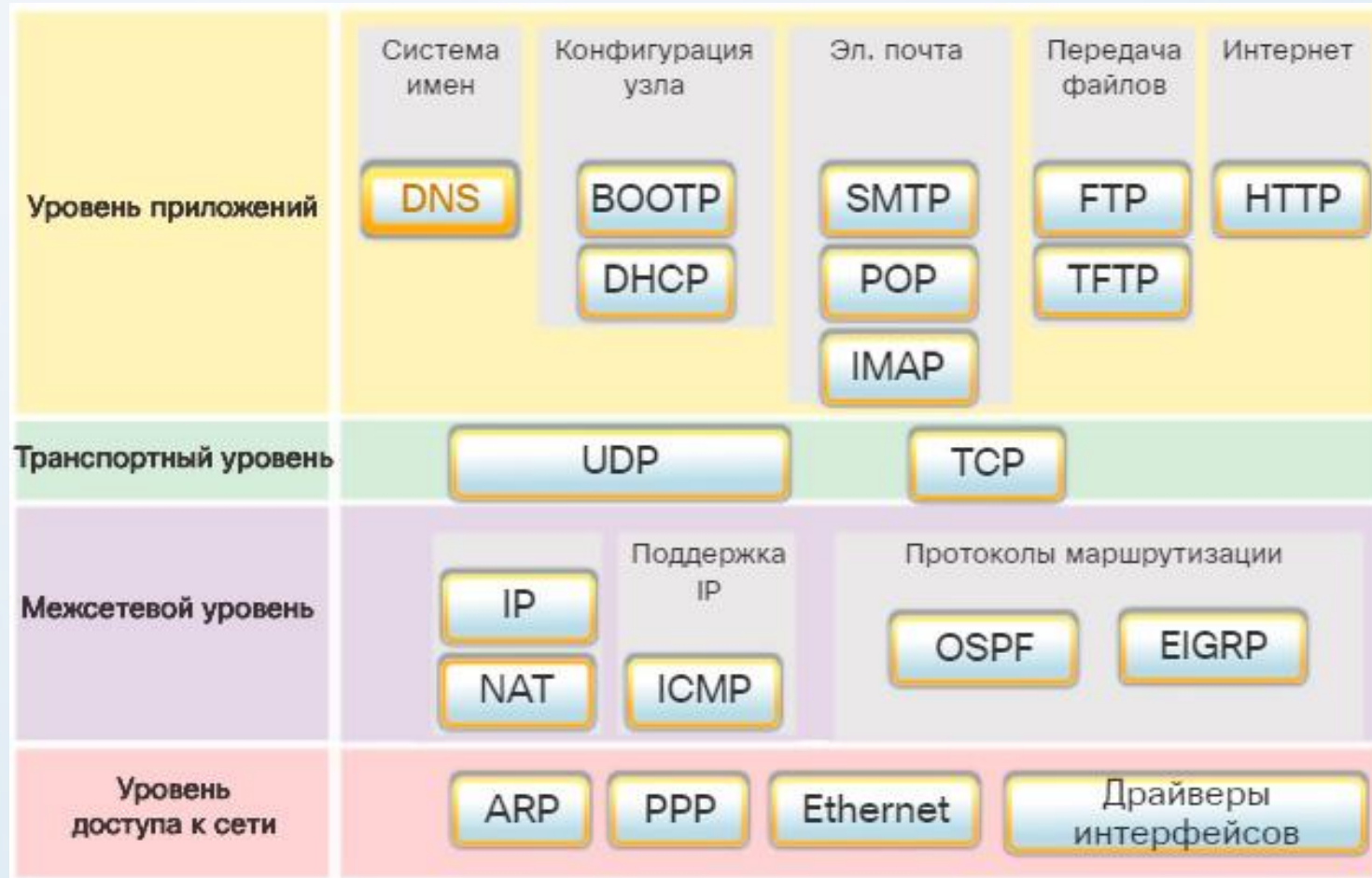


# Модель TCP/IP





# Модель TCP/IP (детально)



<https://artemsannikov.ru/cisco/ccnars/nabor-protokolov-tcp-ip/>

Для взаимодействия между клиентом и сервером используются сокеты.

# Основные компоненты технологии WWW

Тим Бернерс-Ли создал три основных компонента WWW:

- язык гипертекстовой разметки документов [HTML](#) (HyperText Markup Language);
- универсальный способ адресации ресурсов [URI](#) (Universal [Uniform] Resource Identifier);
- протокол обмена гипертекстовой информацией [HTTP](#) (HyperText Transfer Protocol – протокол передачи гипертекста).

Позже, при разработке Mosaic, [NCSA](#) добавила к этим трем компонентам четвертый:

- универсальный интерфейс шлюзов [CGI](#) (Common Gateway Interface) с помощью которого можно создавать динамические HTML-документы.

# Эмуляция HTTP с использованием telnet

```
osboxes@osboxes:~  
File Edit View Search Terminal Help  
[osboxes@osboxes ~]$ telnet iu5.bmstu.ru 80  
Trying 195.19.50.252...  
Connected to iu5.bmstu.ru.  
Escape character is '^]'.  
GET / HTTP/1.0  
  
HTTP/1.1 200 OK  
Server: nginx  
Date: Mon, 09 Nov 2020 08:53:01 GMT  
Content-Type: text/html; charset=UTF-8  
Content-Length: 985  
Connection: close  
Last-Modified: Fri, 12 Apr 2019 09:22:18 GMT  
ETag: "3d9-58651d6d73b52"  
Accept-Ranges: bytes  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"><head>  
  <title>hoster1.uimp.bmstu.ru &mdash; Coming Soon</title>  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>  
  <meta name="description" content="This is a default index page for a new domain."/>  
  <style type="text/css">  
    body {font-size:10px; color:#777777; font-family:arial; text-align:center;}  
    h1 {font-size:64px; color:#555555; margin: 70px 0 50px 0;}  
    p {width:320px; text-align:center; margin-left:auto;margin-right:auto; margin-top: 30px }  
    div {width:320px; text-align:center; margin-left:auto;margin-right:auto;}  
    a:link {color: #34536A;}  
    a:visited {color: #34536A;}  
    a:active {color: #34536A;}  
    a:hover {color: #34536A;}  
  </style>  
</head>
```

# Классификация веб-фреймворков

- Клиентские фреймворки

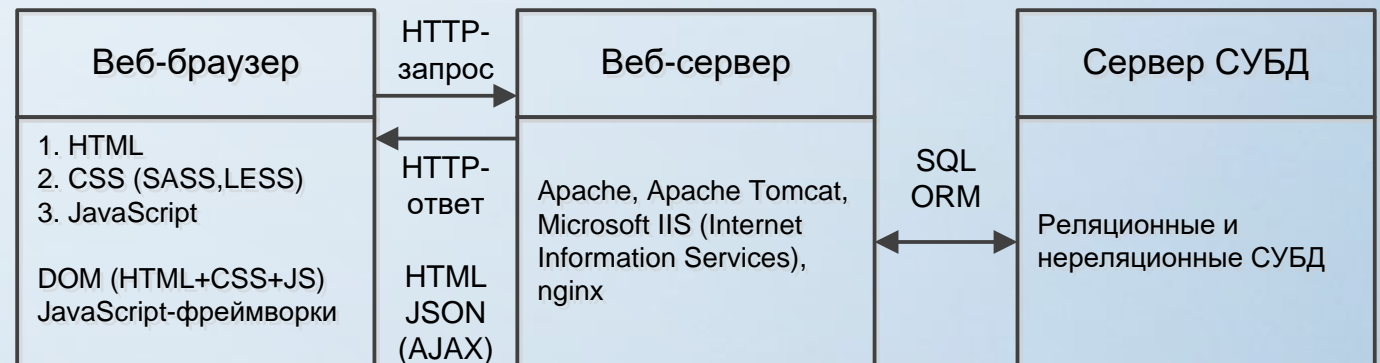
Предназначены для разработки SPA. Реализуют концепцию «толстого» клиента и «тонкого» сервера. Основная функциональность реализована с использованием JavaScript (и транспилируемых в него языков). Некоторые используют паттерн MVC или его разновидности. Примеры: React, Angular, Vue, Svelte.

- Серверные фреймворки

Предназначены для разработки приложений на стороне веб-сервера. Реализуют концепцию «тонкого» клиента и «толстого» сервера. Используют традиционные языки веб-разработки: Python, PHP, Ruby, C#, ...

Подразделяются на две категории:

- Микрофреймворки.
- Традиционные фреймворки с полной функциональностью (используют паттерн MVC или его разновидности).



- Универсальные фреймворки (пример Meteor).



# Веб-разработка на Python

- Интерпретаторы некоторых языков, изначально ориентированных на применение в WWW (например, PHP), обладают встроенным шаблонизатором HTML и могут непосредственно использоваться для веб-разработки.
- В отличие от таких языков, Python для веб-разработки использует исключительно фреймворки.
- Для интеграции с веб-серверами в Python используются спецификация WSGI, которая основана на CGI.
  - В частности, для интеграции с веб-сервером Apache разработан модуль Apache mod\_wsgi.
  - Спецификация WSGI включает такое важное понятие как «Middleware».
- Дальнейшим развитием спецификации WSGI является спецификация ASGI, которая ориентирована на разработку как синхронных, так и асинхронных веб-приложений.

# Традиционный серверный фреймворк. Шаблоны проектирования.

- Традиционный серверный фреймворк строится на шаблоне проектирования MVC или какой-либо из его разновидностей.
  - Шаблон проектирования [Model-View-Controller](#).
  - Шаблон проектирования [Model-View-Presenter](#).

# Фреймворк Django

- Используют паттерн MVP.
- Документация:
  - [Оригинальная документация \(на английском языке\)](#).
  - [Русская документация](#).
- Особенности терминологии:
  - Представление, view (MVC, MVP) – шаблон, template (django).
  - Контроллер (MVC) или Presenter (MVP) – представление, view (django).
- Проекты и приложения (реализация модульной концепции)
  - Проект – это набор настроек и приложений для определенного веб-сайта.
  - Приложение – это независимое веб-приложение, в большинстве случаев работающее с БД. Приложение можно перенести из одного проекта в другой.
  - Аналогичные концепции в других фреймворках:
    - В ASP.NET используется концепция [области \(area\)](#).
    - В технологии [веб-порталов](#) сам портал можно считать аналогом проекта. [Портлет](#) можно считать аналогом приложения, которому выделяется отдельная визуальная область в браузере.

# Микрофреймворк Flask

- Оригинальная документация (на английском языке).
- Статья на русском языке.
- Создание простого приложения:
  - Установим виртуальное окружение (windows cmd):
    - cd <каталог проекта>
    - python -m venv venv #создадим виртуальное окружение
    - venv\Scripts\activate #активируем окружение
    - pip install flask # установим flask
    - pip list
  - Создадим в каталоге проекта Python-файл с простейшим обработчиком URL - <https://flask.palletsprojects.com/en/2.2.x/quickstart/#a-minimal-application>
  - Запустим приложение:
    - flask --app hello run
    - Откроем в браузере адрес - <http://127.0.0.1:5000/>