

< Teach
Me
Skills />

Занятие 16. web приложения

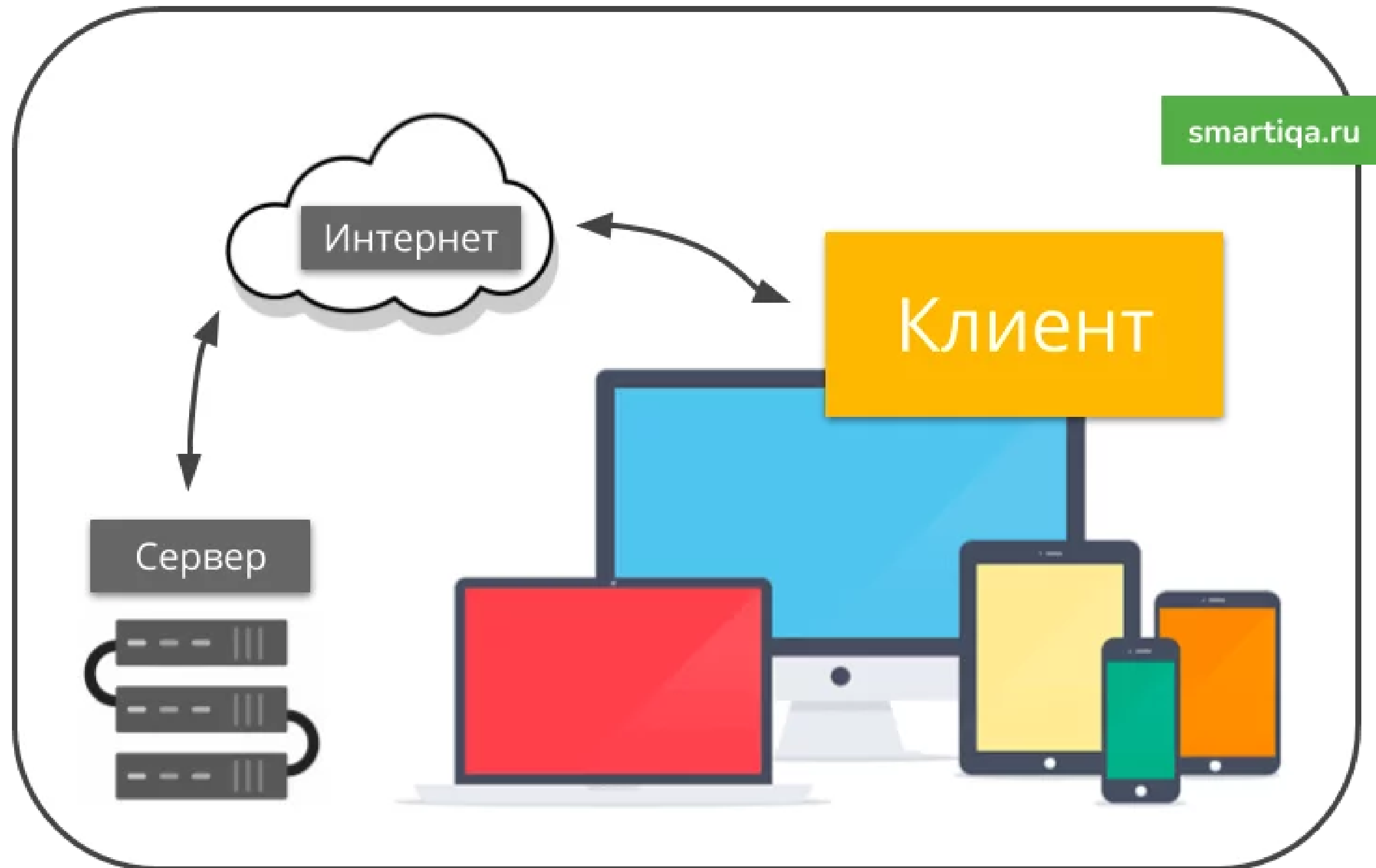
1

Как они работают

Что такое веб-приложения и как они работают

- **Веб-приложение** это клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.
- **Клиент** - приложение, которое запрашивает данные с сервера по сети Интернет через браузер
- **Сервер** - наше приложение, которое умеет обрабатывать запросы и возвращать результат

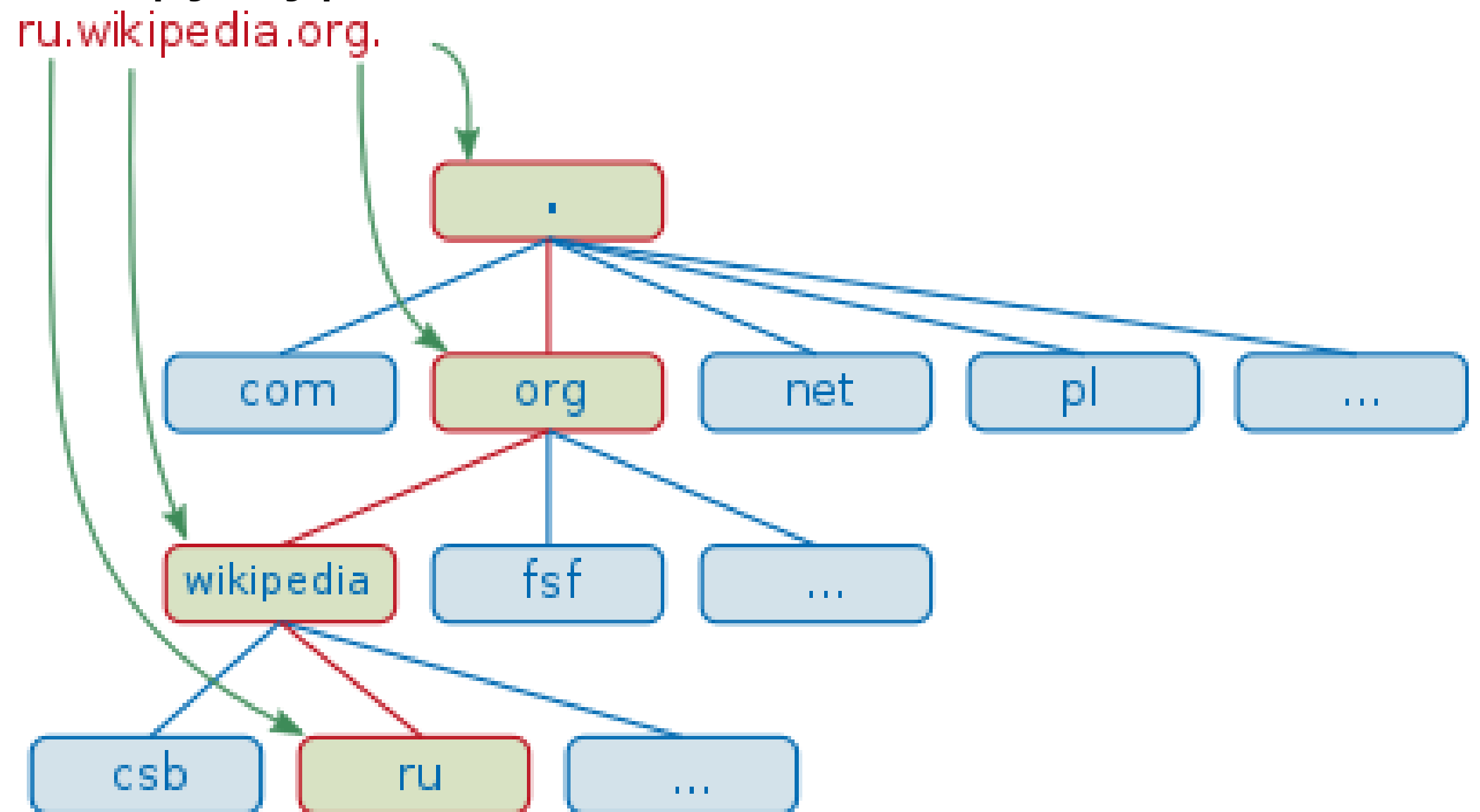
Что такое веб-приложения и как они работают



DNS

Что это такое?

- **domain name system** - система доменных имен
- человекочитаемые доменные имена
- можем получить ip адрес по имени хоста
- тут есть зоны и иерархическая структура имени



OSI - Open Systems Interconnection model

Открытая сетевая модель (7 уровней) - описывает как передаются данные по сети

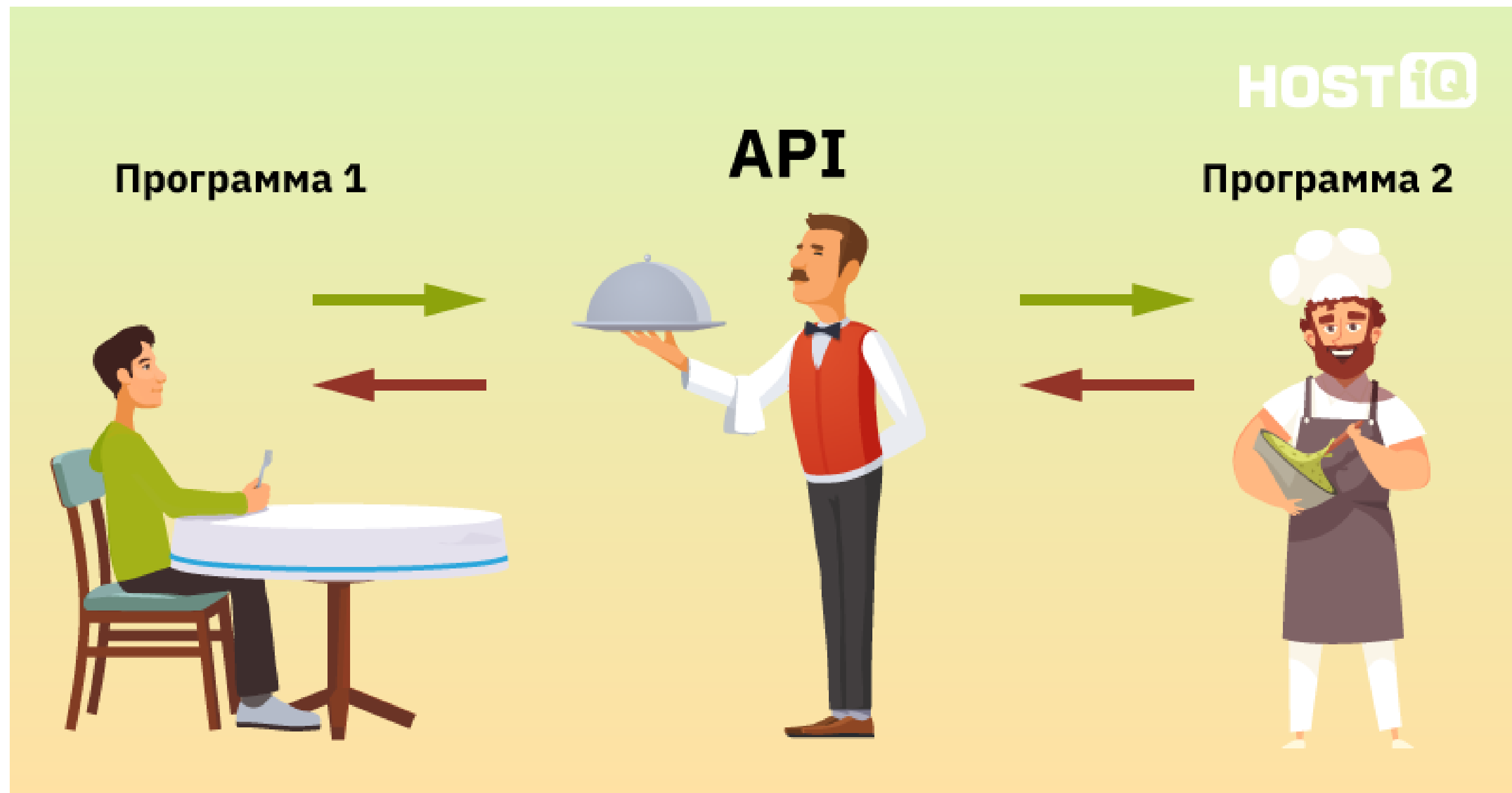
HTTP

7	Прикладной уровень (application layer)
6	Уровень представления (presentation layer)
5	Сеансовый уровень (session layer)
4	Транспортный уровень (transport layer)
3	Сетевой уровень (network layer)
2	Канальный уровень (data link layer)
1	Физический уровень (physical layer)

биты

API - Application Programming Interface

- посредник между двумя программами, который задает правила общения



API - Пример

Разработчик сделал сайт с онлайн-конвертером валют. Он хочет, чтобы конвертером пользовались не только на исходном сайте, но и в банковских приложениях.

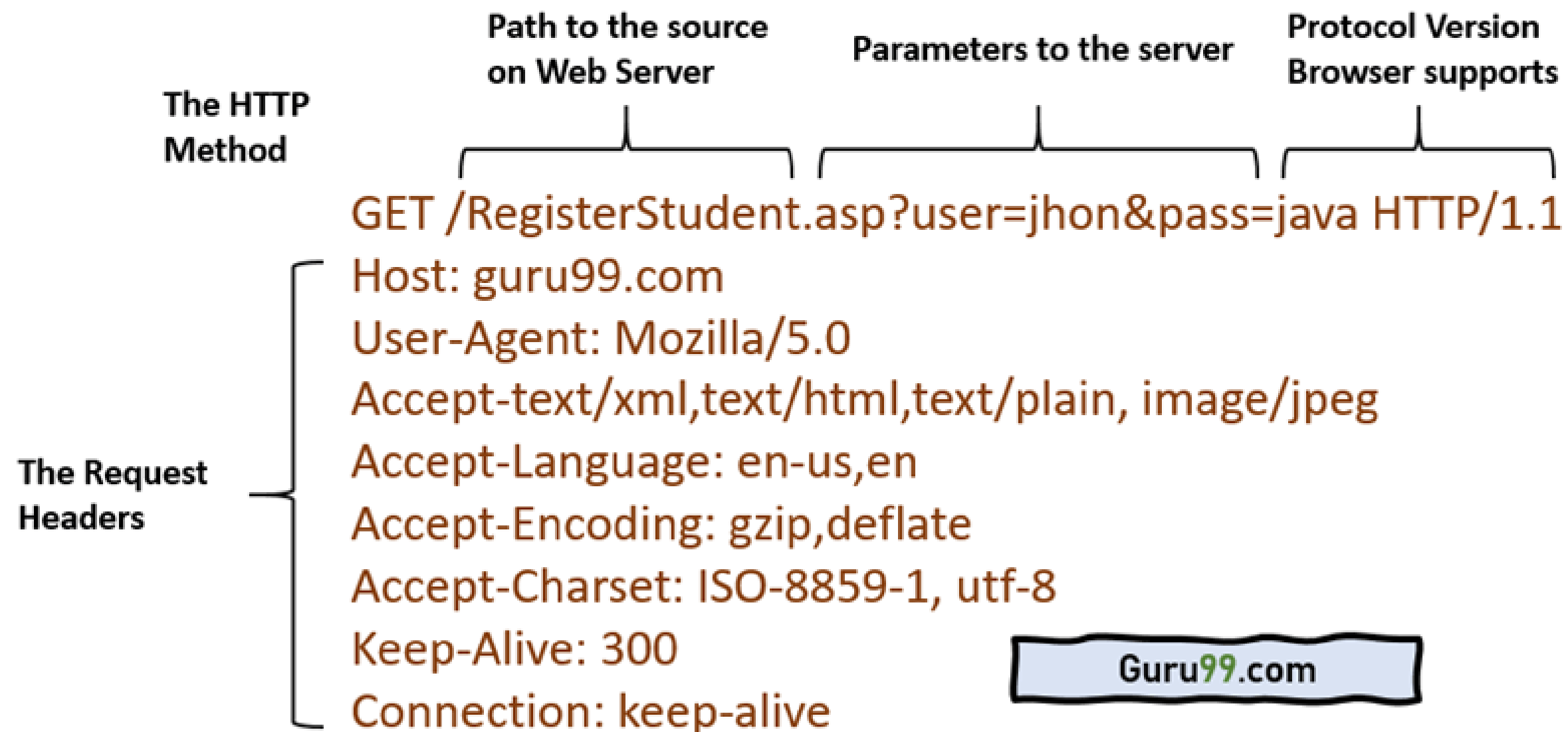
Как это сделать:

- Разработчик конвертера создает API и выкладывает его на своем сайте. Описывает в документации, как работает API и какие есть функции.
- Разработчики банковских приложений берут API конвертера и добавляют его в свой код, если хотят дать своим пользователям функции этого конвертера.
- Пользователь приложения вводит сумму, которую хочет «обменять», и выбирает валюты.
- Приложение посылает на сервер конвертера специальный запрос с суммой, которую нужно перевести в выбранную валюту.
- Конвертер переводит сумму и отправляет ответ с результатом в нужной валюте.
- Приложение получает ответ и показывает его пользователю.

В итоге все довольны: пользователь перевел цифры в другую валюту в приложении, конвертером пользуются не только на сайте, а банковские приложения не писали конвертер с нуля, но улучшили свой продукт.

HTTP - HyperText Transfer Protocol

- широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам)



HTTP - HyperText Transfer Protocol

- пример ответа

```
HTTP/1.1 200 OK
```

```
Date: Sun, 08 Feb xxxx 01:11:12 GMT
```

```
Server: Apache/1.3.29 (Win32)
```

```
Last-Modified: Sat, 07 Feb xxxx
```

```
ETag: "0-23-4024c3a5"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 35
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<h1>My Home page</h1>
```

→ Status Line

} Response Headers

} Response
Message
Header

→ A blank line separates header & body

} Response Message Body

HTTP - HyperText Transfer Protocol

Состоит из:

- request line
- headers
- body

Виды запросов:

- **get** - запрос на получение данных
- **post** - запрос на обновление данных
- **put** - запрос на создание данных
- **patch** - запрос на частичное обновление
- **delete** - запрос на удаление

2

**Из каких технологий
состоит сайт**

Технологии

frontend (внешний вид)

HTML - разметка

CSS - стили

JS - логика на стороне
клиента (в браузере)
и другие (**React** и тд)

backend (логика)

Python- логика

DB - уровень данных

Django - фреймворк
для создания веб-
приложений

Практика - пишем наше первое веб-приложение

Порядок действий:

- `pip install Django`
- `django-admin startproject tms`
- `cd tms`
- `python manage.py runserver`

Создаем первое приложение:

- `python manage.py startapp posts`
- в папке `posts` создаем файл `urls.py` и добавляем туда `home url`
- добавляем `posts` в `settings.py` -> `INSTALLED_APPS`
- в папке `posts` в файле `views.py` создаем вьюху

3

Домашнее задание

Практика

- Написать джанго-приложение для отображения наших Todos
 - Логику по получению todos берем из 13 дз
 - URL для получения todos пока можно записать в settings.py
 - В приложении должно быть 2 url
 - /todos/home- возвращает html со списком todo (желательно использовать синтаксис html, например, чтобы список дел был в виде таблицы)
 - /todos - возвращает список todo в формате json
- В качестве дополнения можно реализовать:
 - /todos/<id> - получение по id
 - /todos/page/<page_number> - реализовать пагинацию, например, page_number=1 будет возвращает первые 10 записей, page_number=2 - вторые 10 и тд

The background is a vibrant yellow color, covered with a dense, repeating pattern of various geometric shapes. These shapes include circles, squares, triangles, and lines, some of which are filled with a fine grid pattern. The shapes are scattered across the entire surface, creating a dynamic and modern aesthetic.

Спасибо за внимание!