

Сетевое взаимодействие

Ивановский Николай

Никонов Владимир



ТИНЬКОФФ



Сегодня в программе



Сетевые протоколы



TCP и UDP



HTTP



Параметры в HTTP



Безопасность в HTTP



REST

Сетевые протоколы

Протокол – набор соглашений

Сетевой протокол – набор соглашений, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.



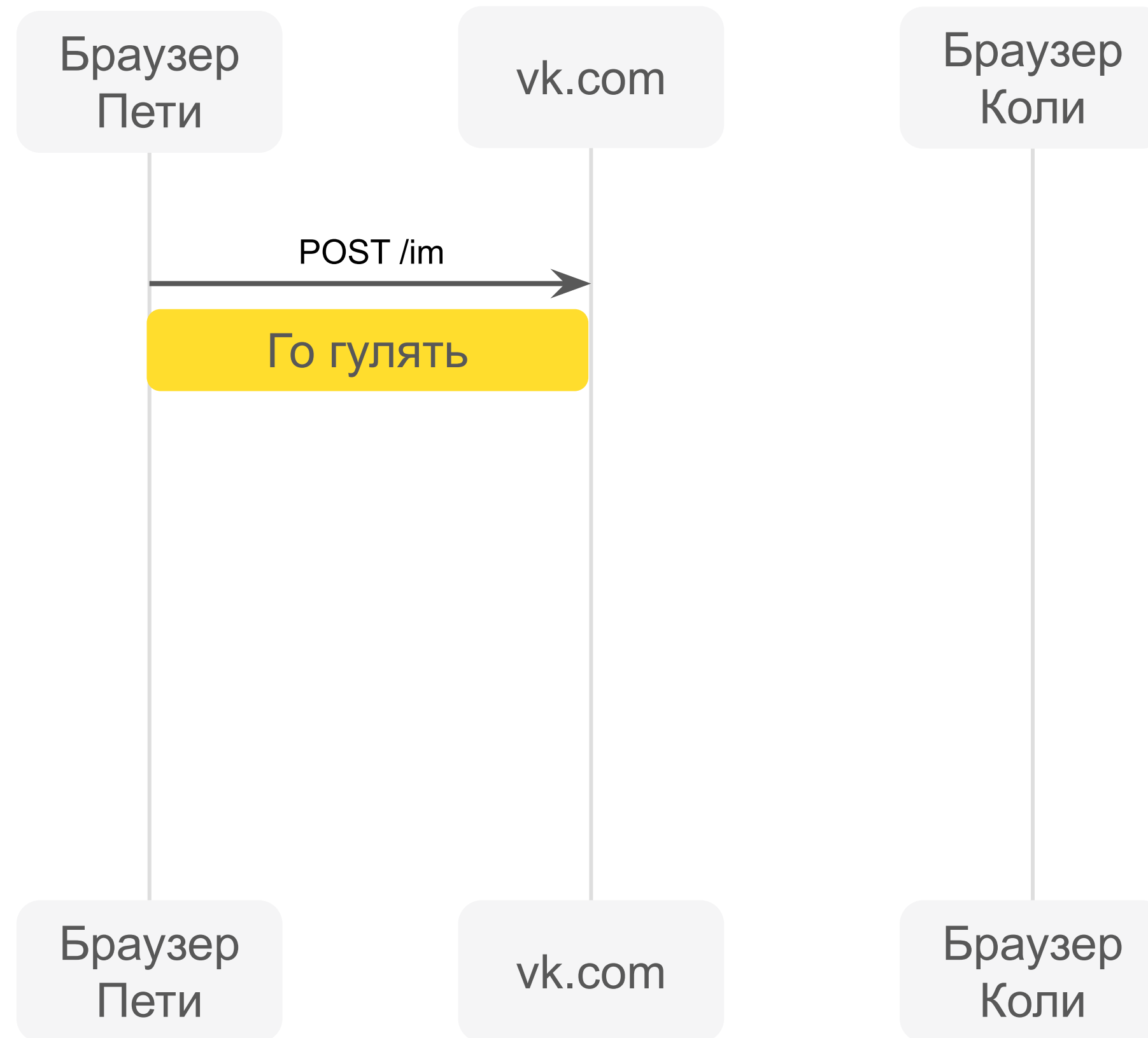
Сетевые протоколы

Набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.



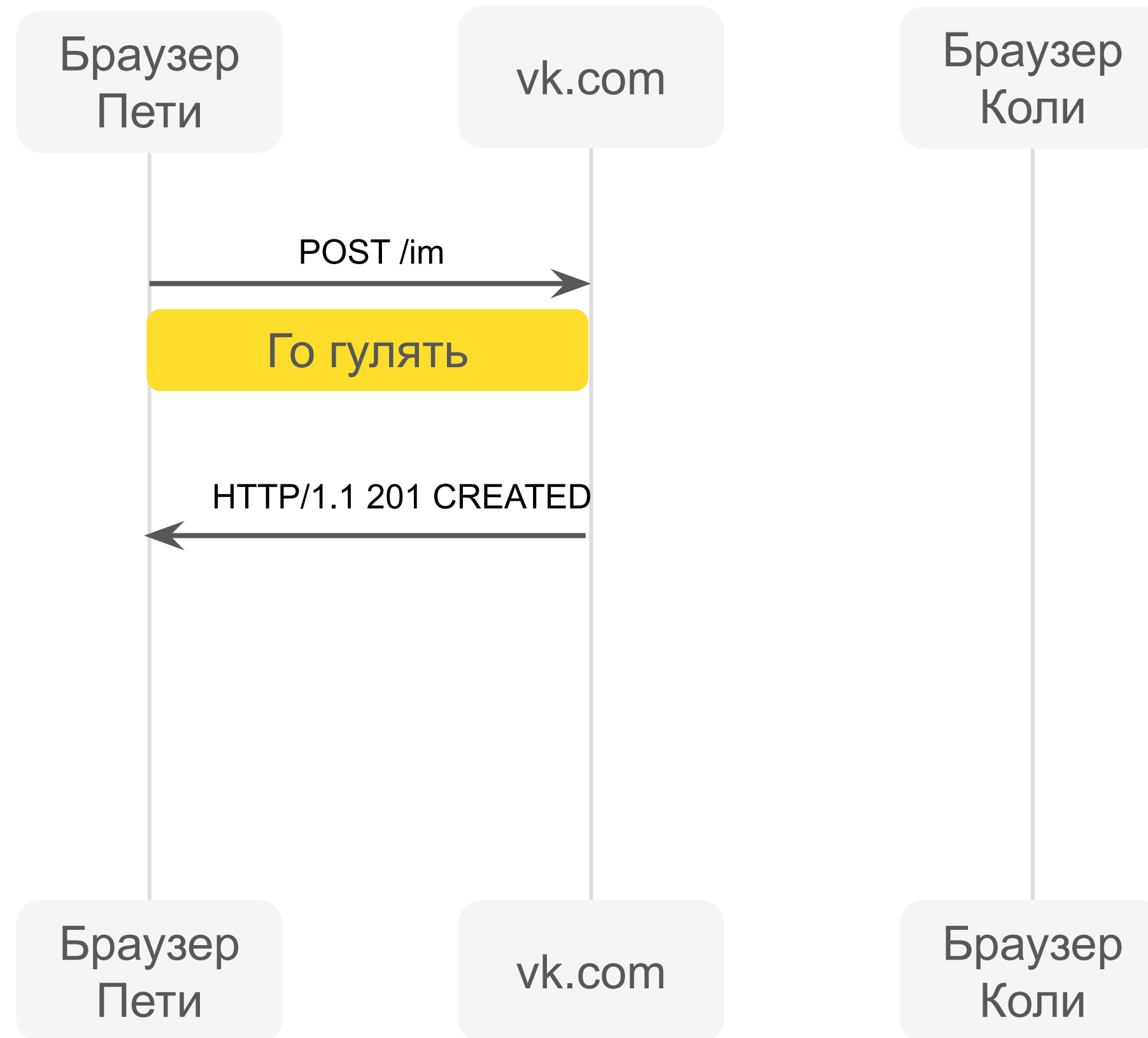
Сетевые протоколы

Набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.



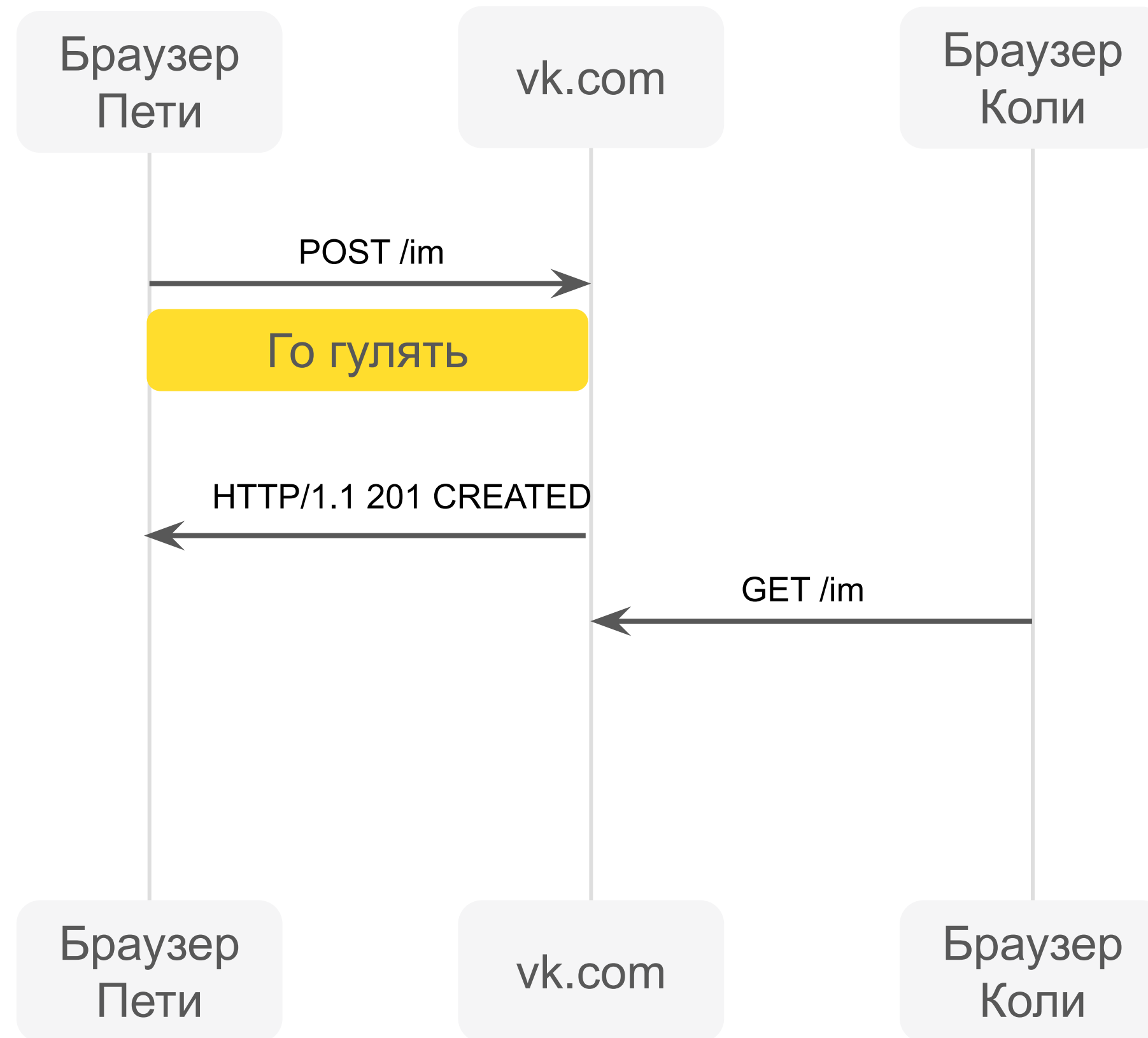
Сетевые протоколы

Набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.



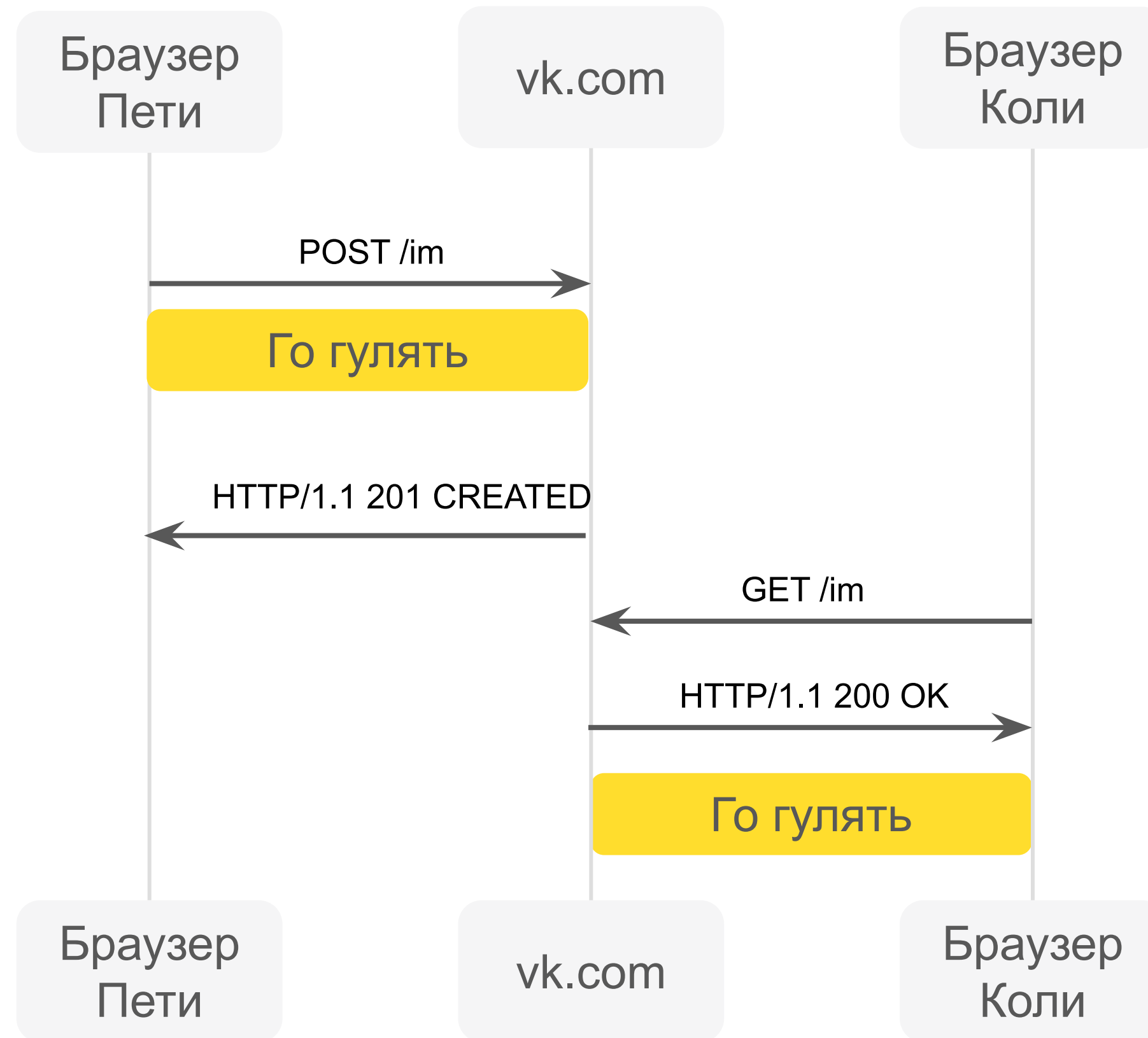
Сетевые протоколы

Набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.



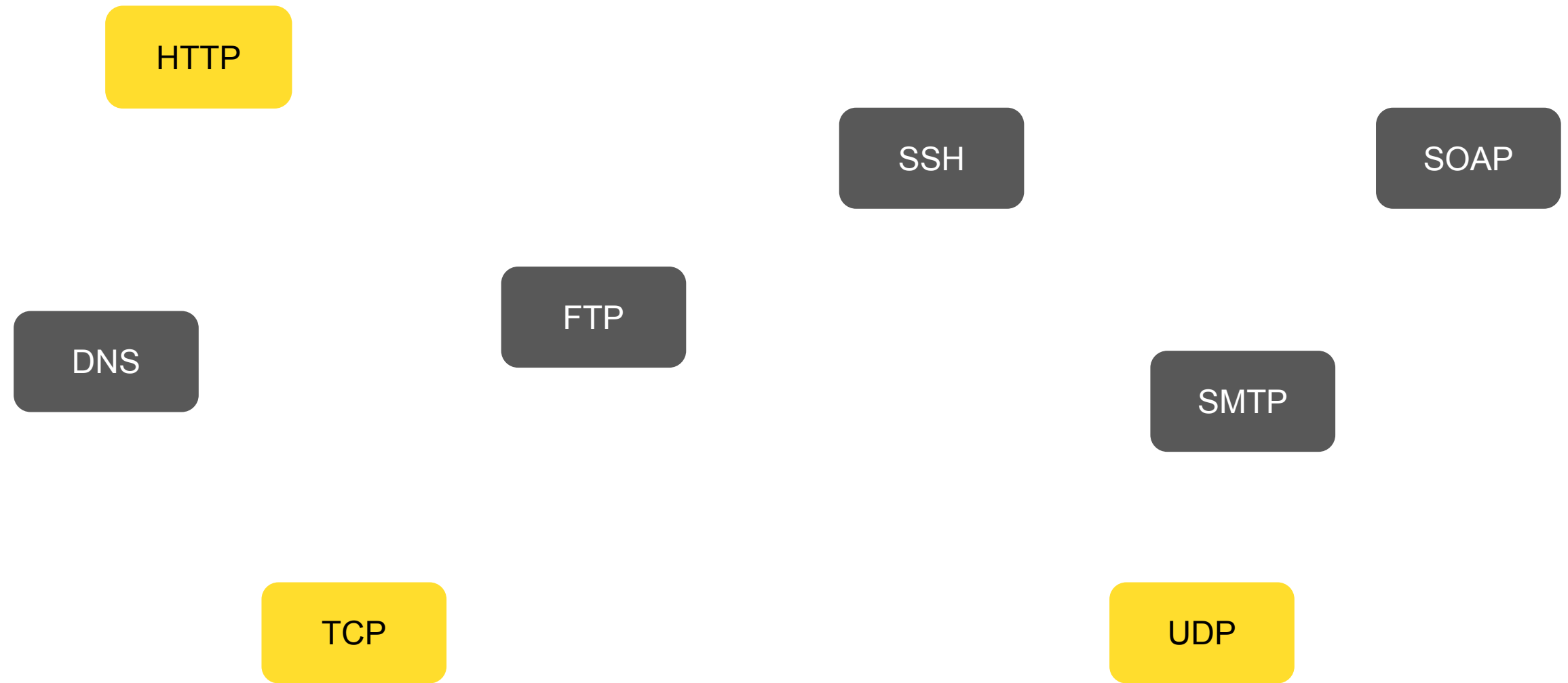
Сетевые протоколы

Набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.



Сетевые протоколы

Желтые – поговорим в лекции
Серые – устаревшие или не интересные нам



Вопросы



TCP - протокол

TCP (Transmission Control Protocol) – сетевой протокол, который используется для обеспечения надёжной доставки данных на транспортном уровне.



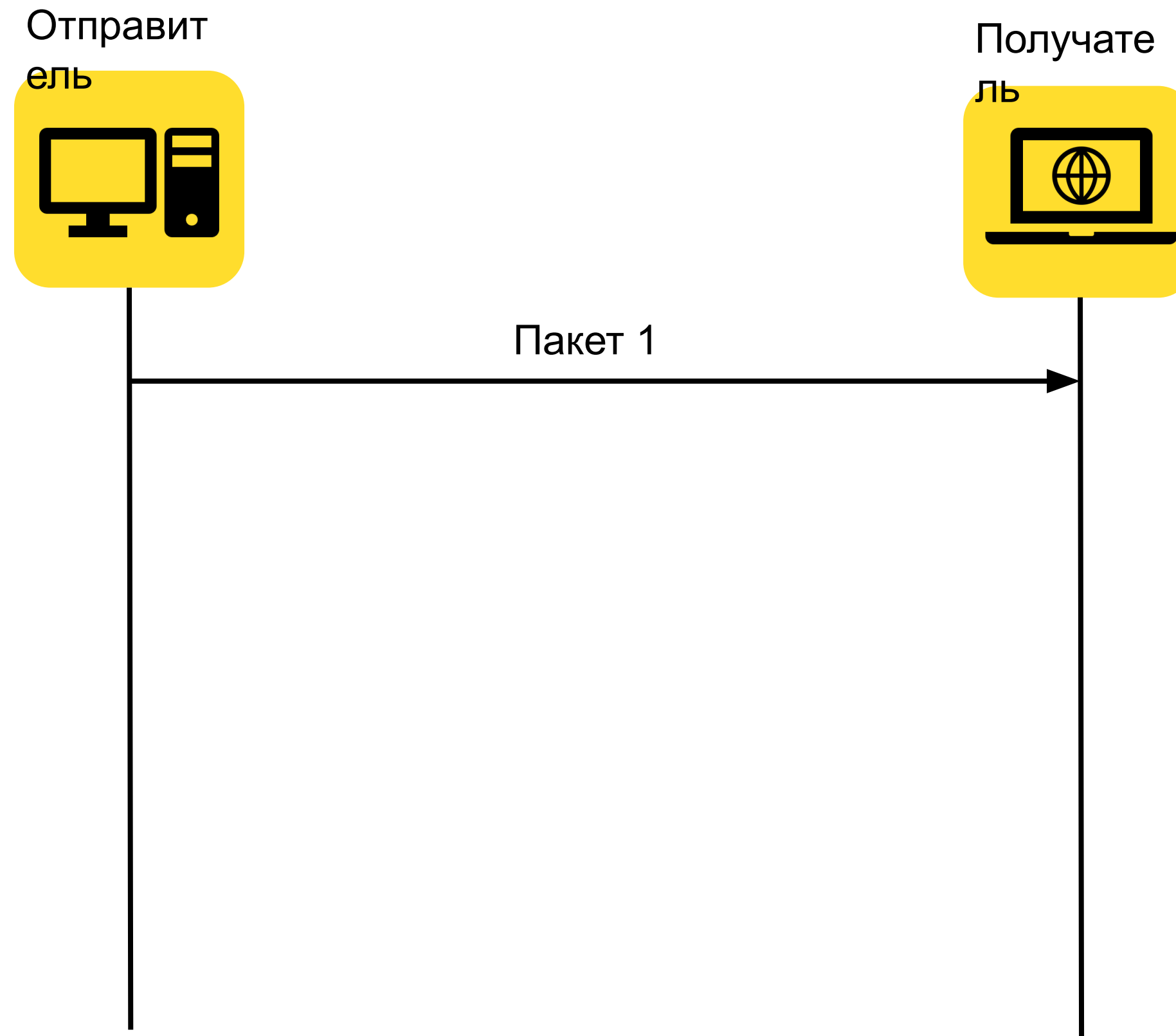
Пакеты

Пакет – минимальная единица данных, используемая при общении по ТСР



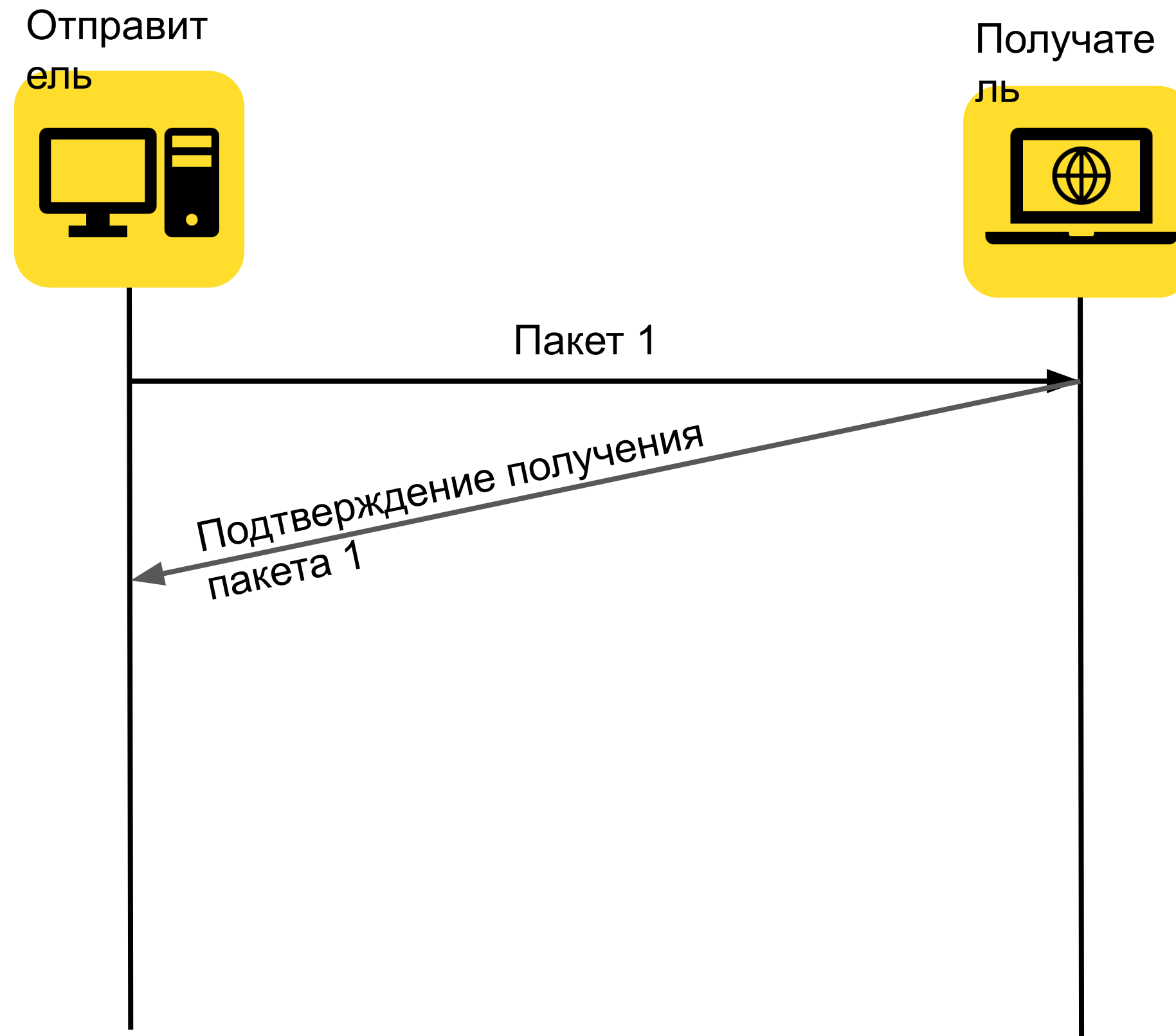
TCP - протокол

TCP (Transmission Control Protocol) – сетевой протокол, который используется для обеспечения надёжной доставки данных на транспортном уровне.



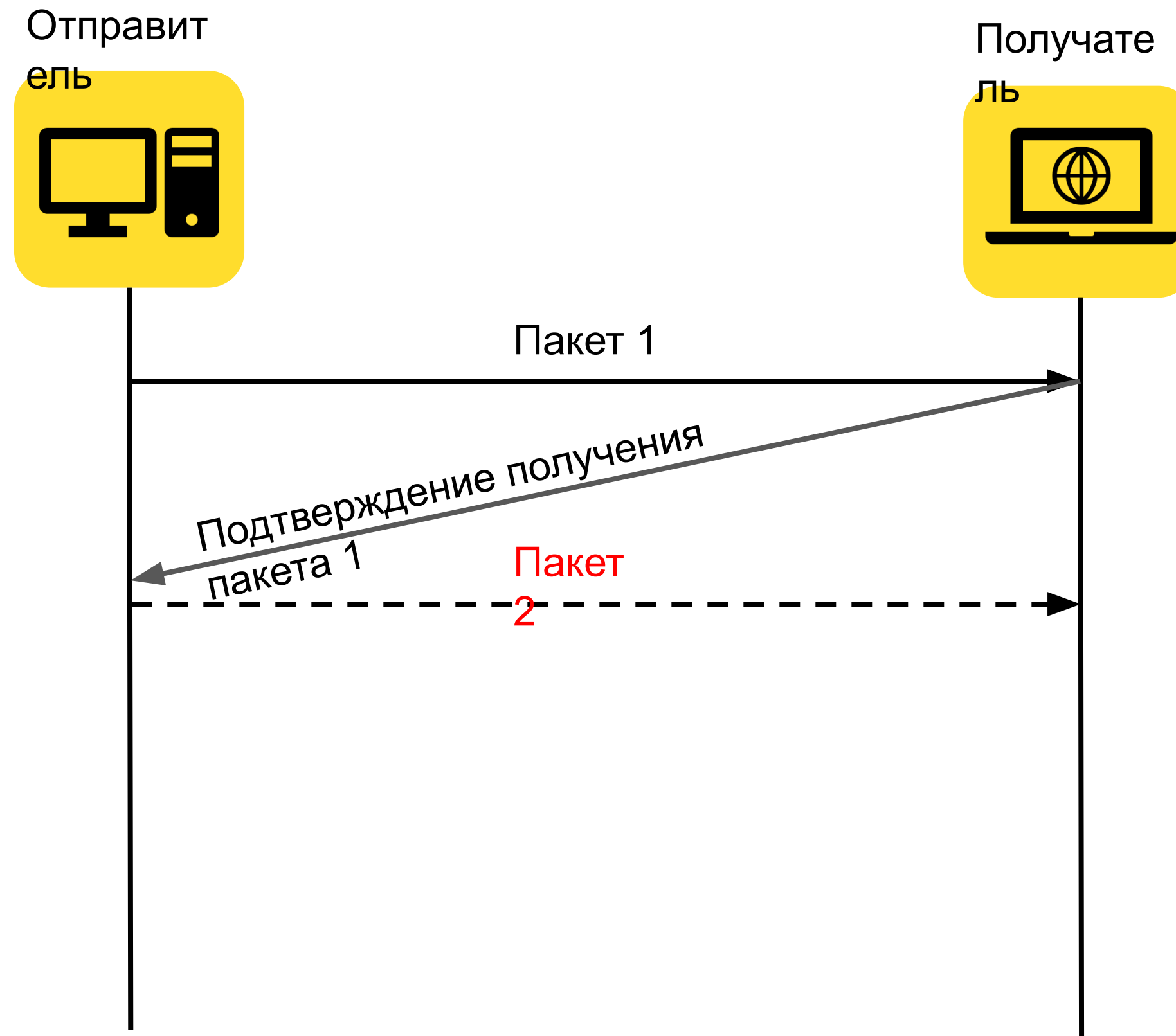
TCP - протокол

TCP (Transmission Control Protocol) – сетевой протокол, который используется для обеспечения надёжной доставки данных на транспортном уровне.



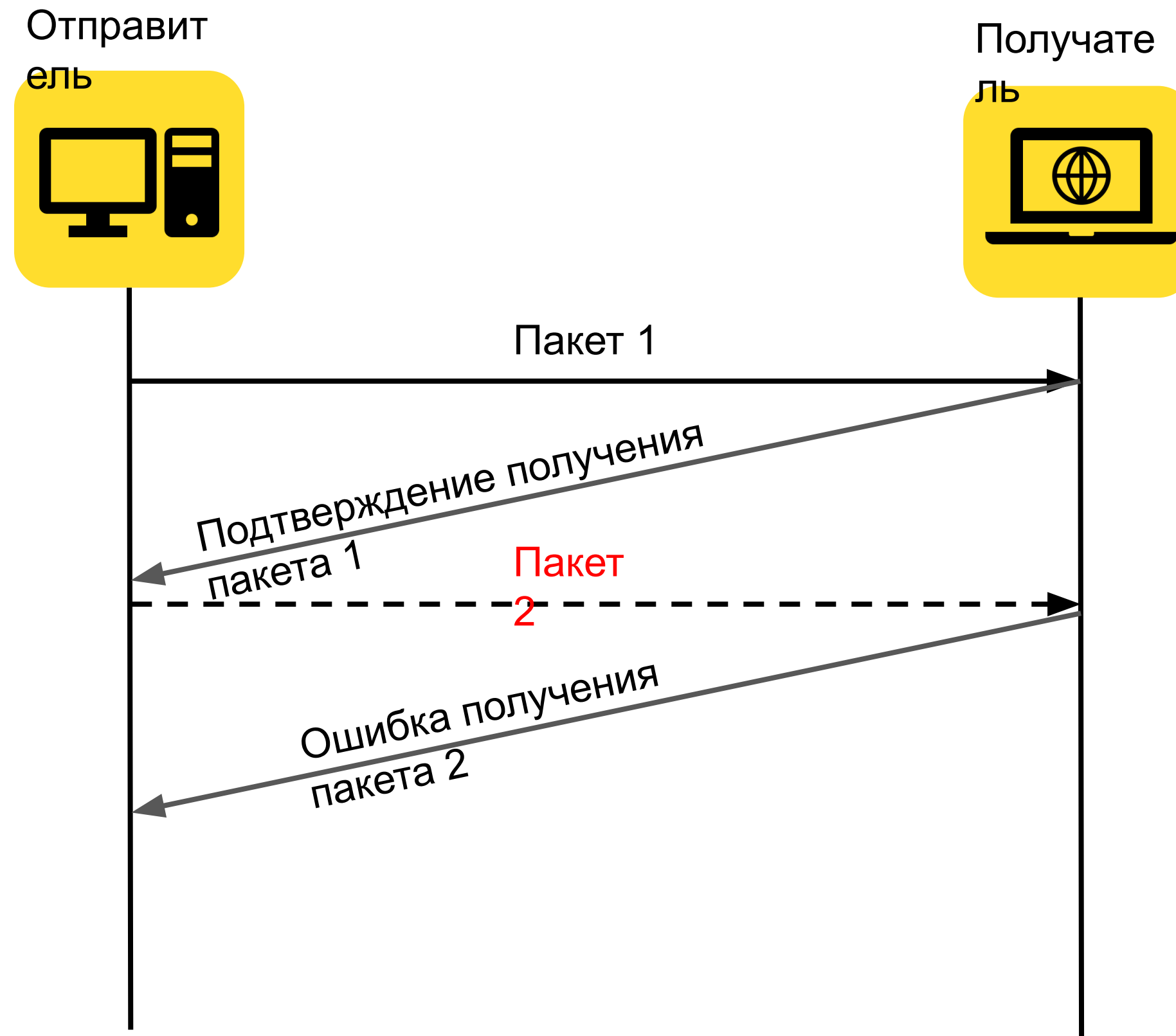
TCP - протокол

TCP (Transmission Control Protocol) – сетевой протокол, который используется для обеспечения надёжной доставки данных на транспортном уровне.



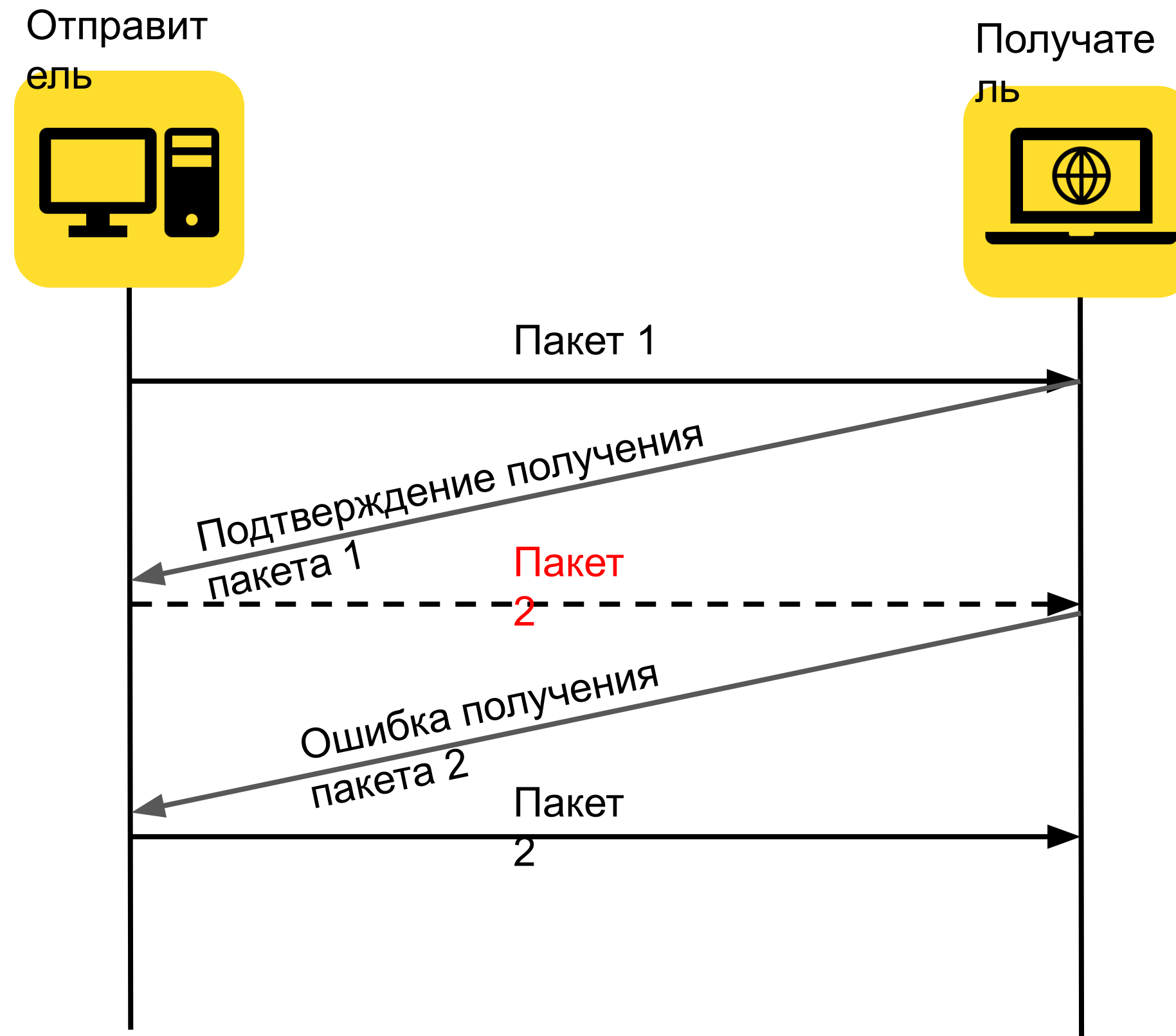
TCP - протокол

TCP (Transmission Control Protocol) – сетевой протокол, который используется для обеспечения надёжной доставки данных на транспортном уровне.



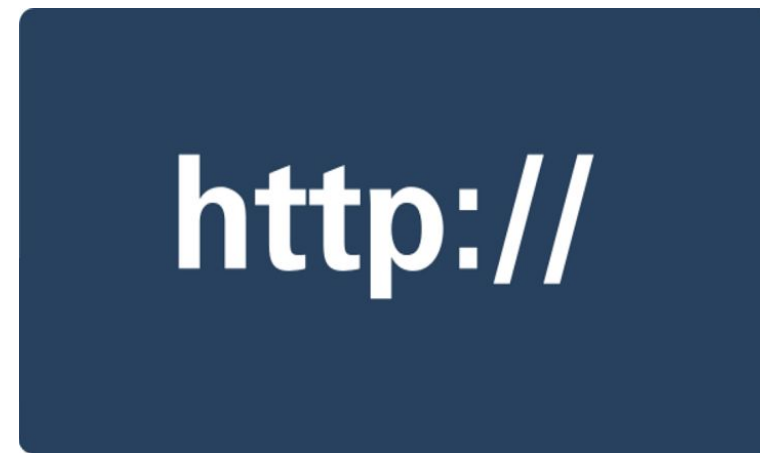
TCP - протокол

TCP (Transmission Control Protocol) – сетевой протокол, который используется для обеспечения надёжной доставки данных на транспортном уровне.



Где используется TCP протокол

- HTTP
- Базы данных
- SOAP
- Очереди
- GRPS
- ...



Вопросы



UDP - протокол

UDP (*User Datagram Protocol*) – более простой, основанный на сообщениях протокол без установления соединения. Доставка не гарантируется

UDP - протокол

UDP (*User Datagram Protocol*) – более простой, основанный на сообщениях протокол без установления соединения. Доставка не гарантируется



UDP - протокол

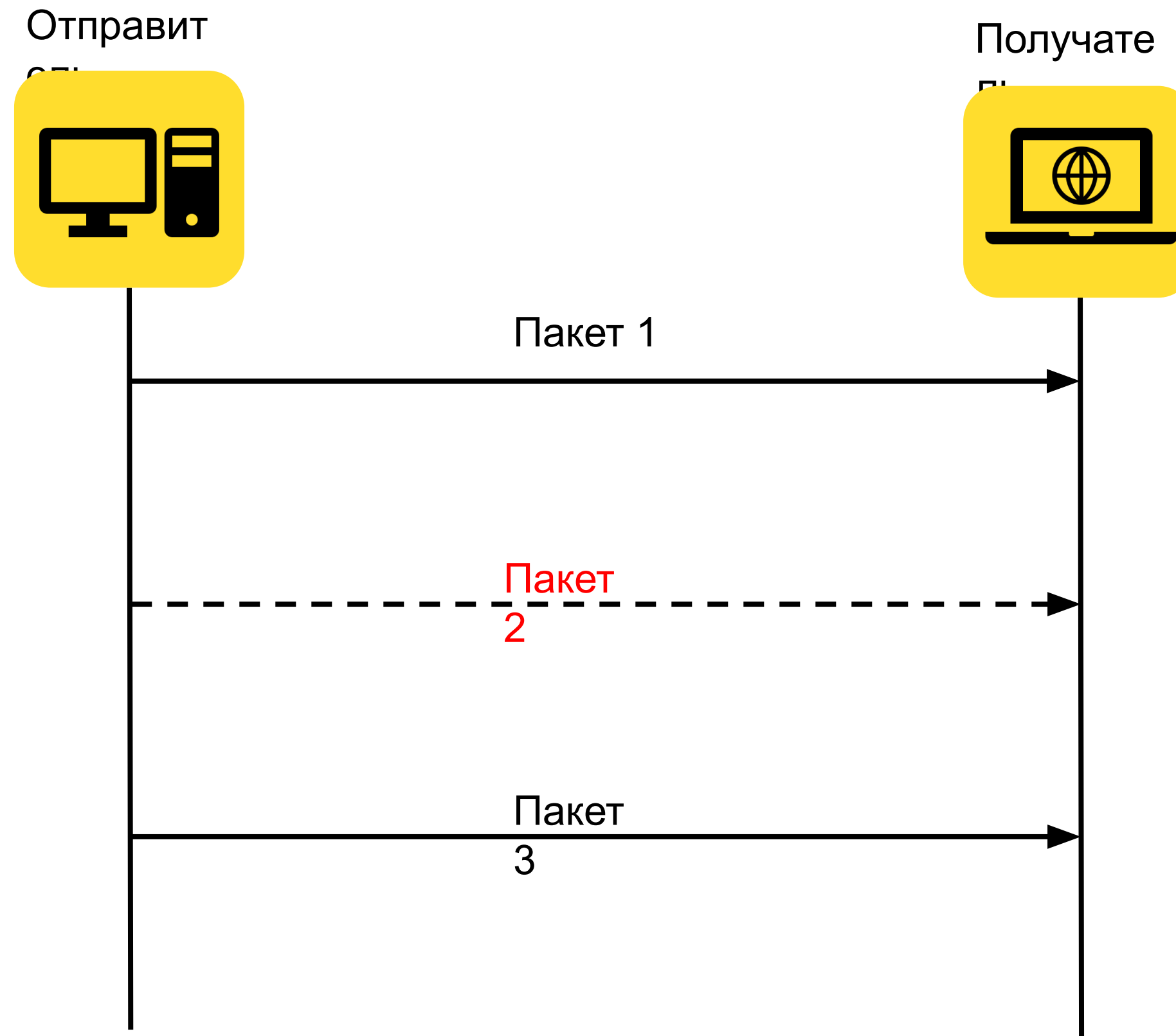
UDP (*User Datagram Protocol*) – более простой, основанный на сообщениях протокол без установления соединения. Доставка не гарантируется

UD
P



UDP - протокол

UDP (*User Datagram Protocol*) – более простой, основанный на сообщениях протокол без установления соединения. Доставка не гарантируется



UDP - протокол

UDP (*User Datagram Protocol*) – более простой, основанный на сообщениях протокол без установления соединения. Доставка не гарантируется



Вопросы



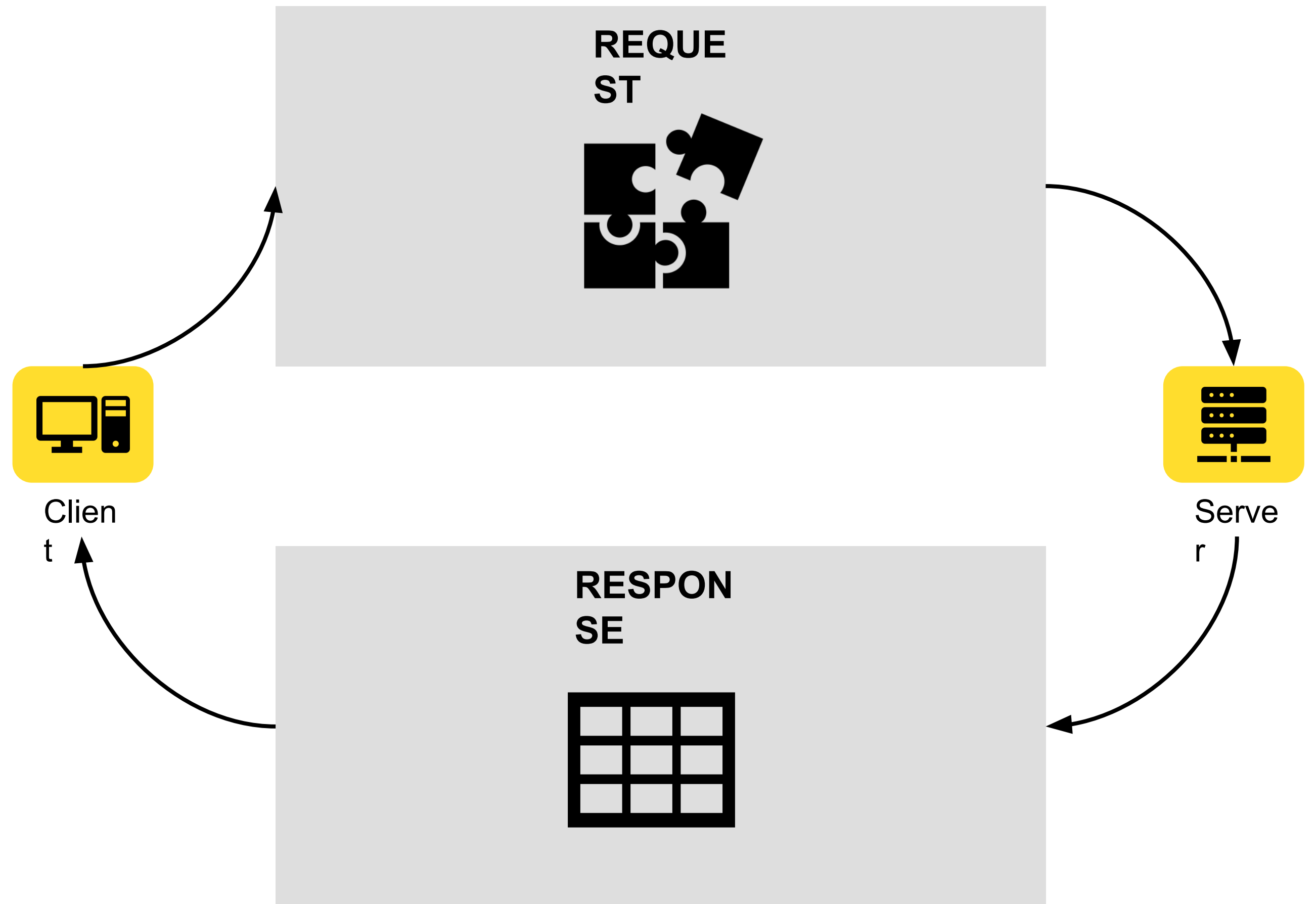
HTTP – протокол

Протокол передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.

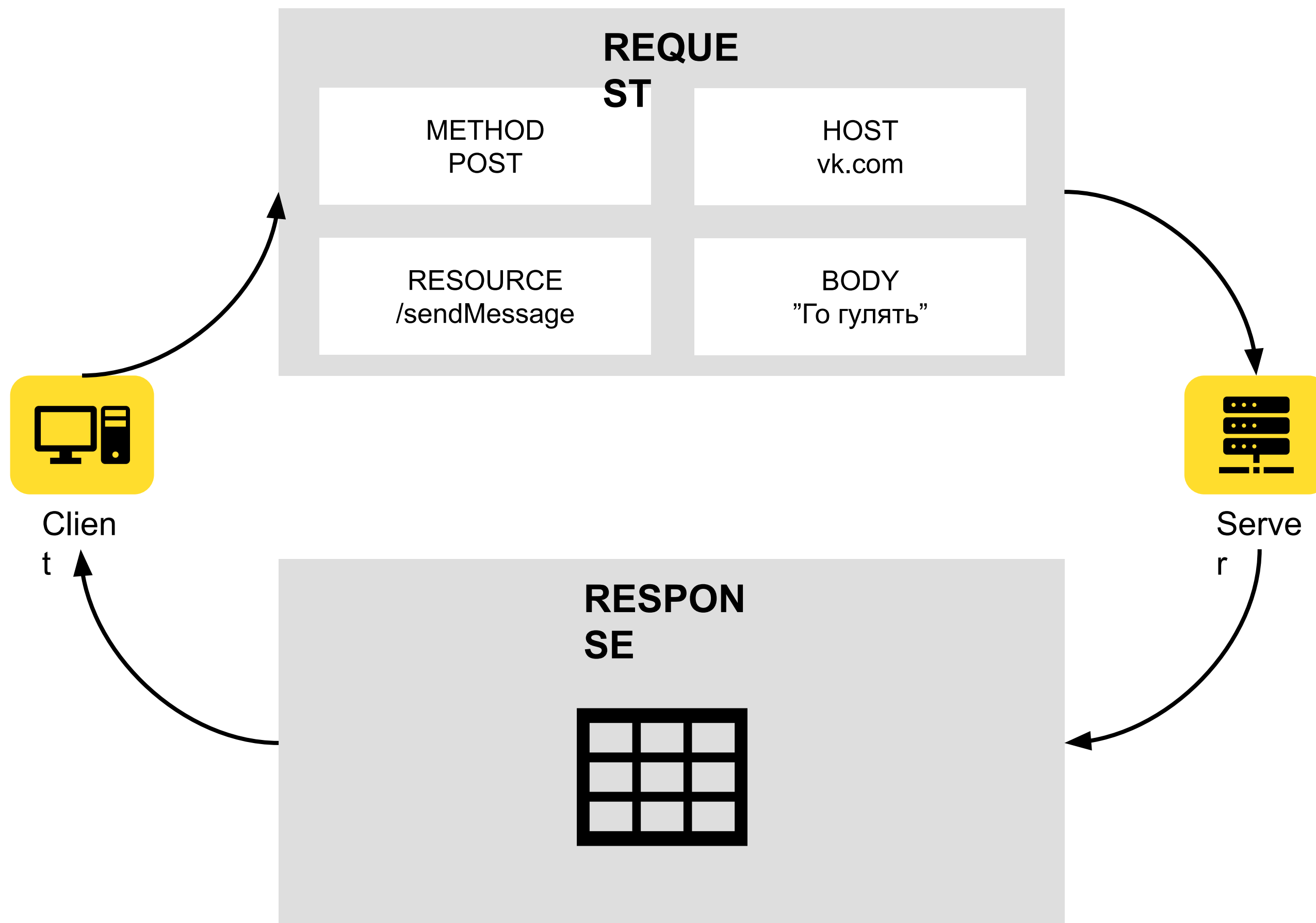
A 3D rendered graphic featuring the text 'http://' in a bold, black, sans-serif font. The text is mounted on a white rectangular panel that is part of a larger blue structure. The letters have a slight shadow, giving them a three-dimensional appearance. To the right of the text, there is a stylized starburst or explosion graphic with a black outline and a white fill, pointing towards the right. The background is a solid light blue color.

http://

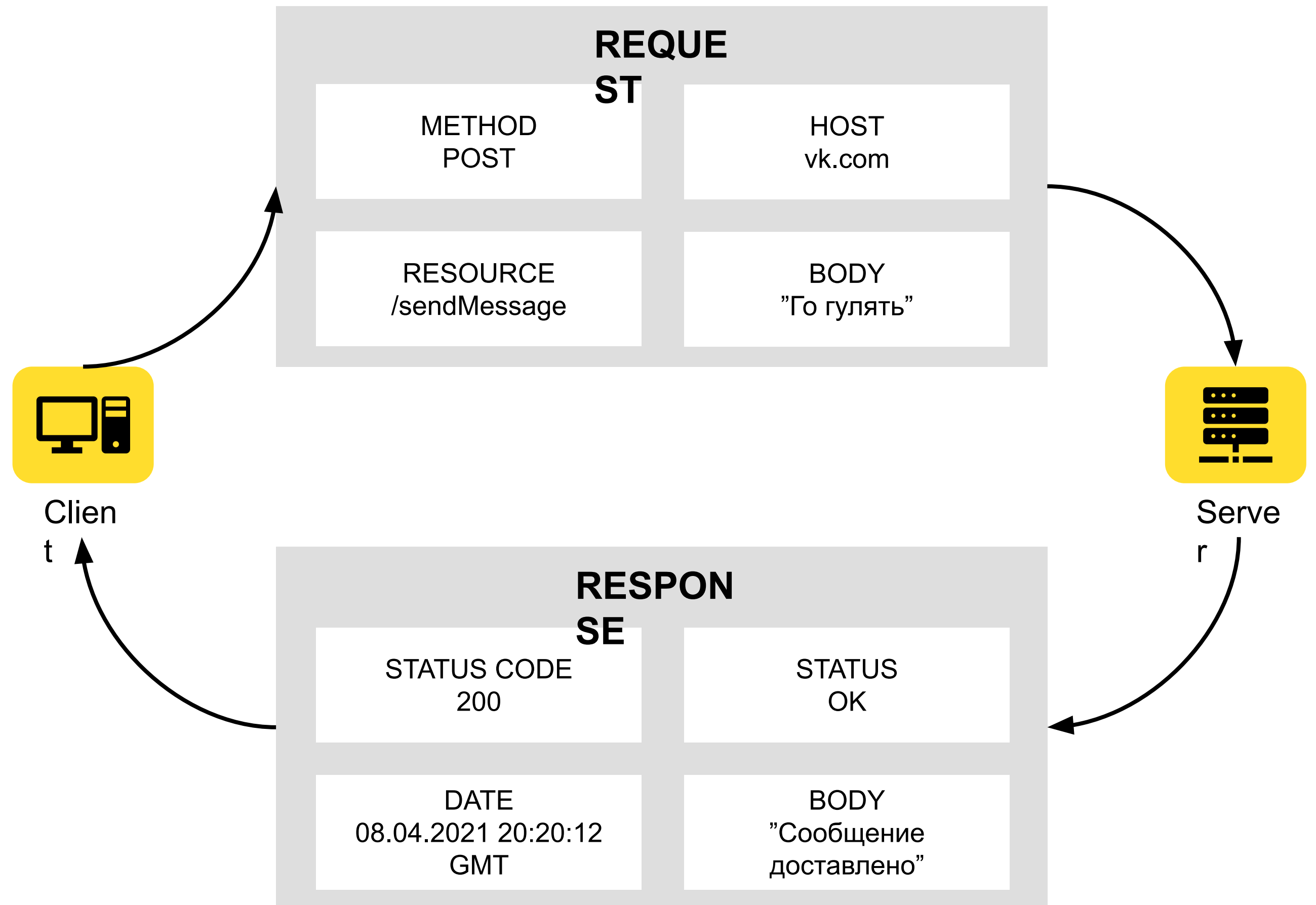
Из чего состоит HTTP запрос



Из чего состоит HTTP запрос



Из чего состоит HTTP запрос



Адрес запроса

HOST - адрес сервера, на который отправляются запросы

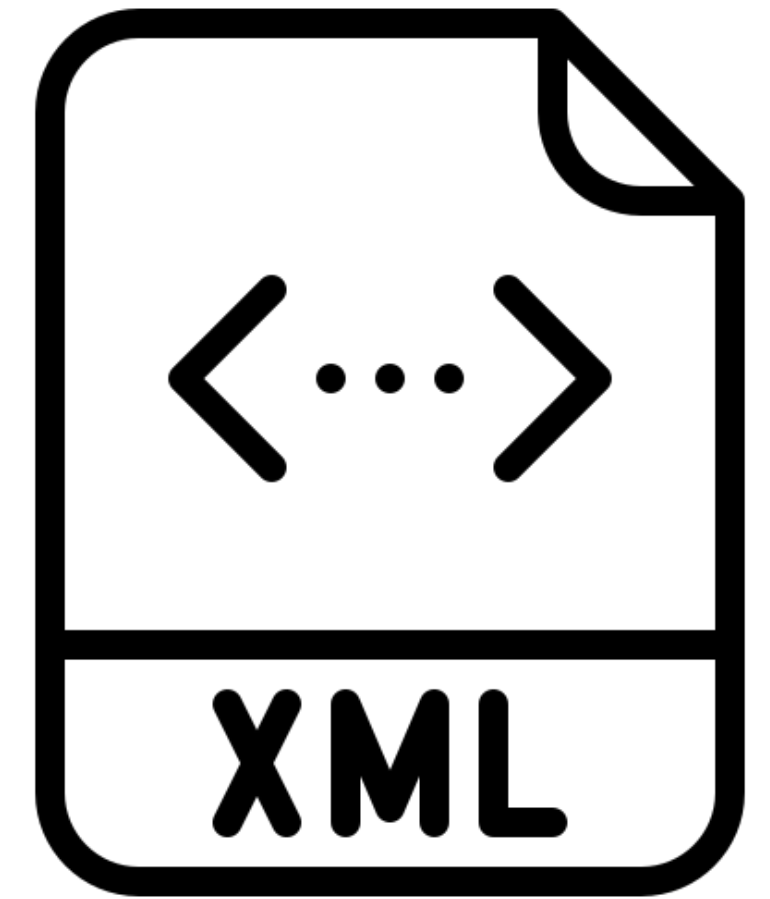
RESOURCE – путь на сервере, по которому выполняется запрос



BODY

Тело запроса или ответа
Содержит данные, предназначенные
для обмена между клиентом и
сервером

{JSON}



TEXT

JSON

Текстовый формат обмена данными,
основанный на JavaScript

```
{
  "message": "Го гулять",
  "files": [
    {
      "type": "image",
      "source": "mem.jpg"
    },
    {
      "type": "document",
      "source": "anegdot.docx"
    }
  ]
}
```


JSON

Текстовый формат обмена данными,
основанный на JavaScript

```
{
  "message": "Го гулять",
  "files": [
    {
      "type": "image",
      "source": "mem.jpg"
    },
    {
      "type": "document",
      "source": "anegdot.docx"
    }
  ]
}
```



```
public class Body{
    private List<FilesItem> files;
    private String message;
}
```

```
public class FilesItem{
    private String source;
    private String type;
}
```

Вопросы



Методы HTTP



Методы

Указывают, какую операцию надо выполнить над ресурсом

GET

POST

HEAD

PUT

PATCH

DELETE

TRACE

GET

- Служит для получения информации
- Не содержит тела
- Может быть вызван из браузера

Method

→ **Request**
GET /users/vasya/messages HTTP/1.1
Host: vk.com

Response
body

→ **Response**
HTTP/1.1 200 OK
Date: Tue, 08 May 2021 12:41:58 GMT
"Го гулять"

POST

- Служит для отправки информации
- Ответ может не содержать тела

Meth
od
Reque
st
body

Request

→ POST /messages/send HTTP/1.1

Host: vk.com

→ “Го гулять”

Response

HTTP/1.1 200 OK

Date: Tue, 08 May 2021 12:41:58 GMT

POST

- Служит для отправки информации на сервер
- Ответ может содержать тело

Meth
od
Reque
st
body

→ **Request**
→ **POST** /messages/send HTTP/1.1
Host: vk.com
→ "Го гулять"

Respon
se
body

→ **Response**
HTTP/1.1 200 OK
Date: Tue, 08 May 2021 12:41:58 GMT
→ "Отправлено успешно"

PUT/PATCH

- Служит для отправки обновленной информации на сервер
- Ответ может содержать тело

Method
Request
body

→ **Request**
→ PUT /messages/update/<messageId> HTTP/1.1
Host: vk.com
→ "Го гулять, а то побью"

Response
body

→ **Response**
HTTP/1.1 200 OK
Date: Tue, 08 May 2021 12:41:58 GMT
→ "Обновлено успешно"

DELETE

- Служит для удаления информации
- Ответ может содержать тело

Method

→ **Request**
Delete /messages/delete/<messageId> HTTP/1.1
Host: vk.com

Response
body

→ **Response**
HTTP/1.1 200 OK
Date: Tue, 08 May 2021 12:41:58 GMT
"Уделено успешно"

HTTP методы

Название	Описание
GET	Получение информации с сервера
POST	Отправка информации на сервер
PATCH/PUT	Отправка информации на сервер для обновления
DELETE	Удаление информации на сервере

Вопросы



Параметры запроса HTTP

Path parameters

Обязательный параметр - часть resource.
Расположен в строго определенном месте

`http://vk.com/messages/{messageID}`

Path parameters

Обязательный параметр - часть resource.
Расположен в строго определенном месте

`http://vk.com/messages/{messageID}`

`http://vk.com/person/{personID}/profile`

Path parameters

Обязательный параметр - часть resource.
Расположен в строго определенном месте

`http://vk.com/messages/{messageID}`

`http://vk.com/person/{personID}/profile`

`http://vk.com/person/{personID}/photo/{photoID}`

Path parameters

Обязательный параметр - часть resource.

Расположен в строго определенном месте

<https://www.mvideo.ru/products/pioneer-mvh-09ub-10012692/specification>



Path parameters

Обязательный параметр - часть resource.
Расположен в строго определенном месте

`https://www.mvideo.ru/products/specification/pioneer-mvh-09ub-10012692`



Страница не найдена.

Query parameters

Не обязательный параметр.

Расположен после resource.
Отделен знаком «?»

`http://vk.com/search?`

Query parameters

Не обязательный параметр.

Расположен после resource.
Отделен знаком «?»

Представляет собой
<Ключ>=<Значение>

`http://vk.com/search?city=Ryazan`

Query parameters

Количество не
ограничено

Соединяются через "&"

`http://vk.com/search?city=Ryazan`

`http://vk.com/search?city=Ryazan&school=1`

`http://vk.com/search?city=Ryazan&school=1&...`

Query parameters

Порядок не
важен

`http://vk.com/search?city=Ryazan&school=1`



`http://vk.com/search?school=1&city=Ryazan`

Параметры запроса и ответа HTTP



Headers

Предназначены для передачи сервисной информации

Представляют собой пары <Ключ>:<Значение>

:method: POST

:scheme: https

accept-language:

ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7

content-length: 470

content-type: application/x-www-form-urlencoded

origin: https://www.mvideo.ru

Вопросы



Авторизация

Используется для ограничения
доступа к ресурсам



BASIC

Передается в хэдере Authorization

Header



Request

GET /users/vasya/messages HTTP/1.1

Host: vk.com

Headers:

Authorization: Basic

YWxhZGRpbjpvYVuc2VzYW1l

Response

HTTP/1.1 200 OK

Date: Tue, 08 May 2021 12:41:58 GMT

“Го гулять”

Base64

Стандарт кодирования двоичных данных

DECODE
ENCODE

→ aladdin:opensesame

→ YWxhZGRpbjpvYVuc2VzYW1l

Base64

Стандарт кодирования двоичных данных



Прежде всего - безопасность

HTTPS

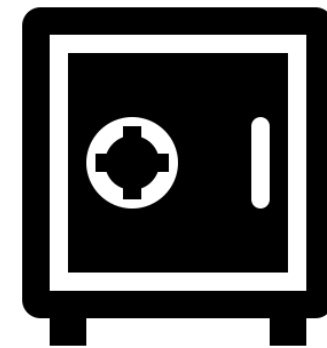
Протокол безопасной передачи данных, поддерживает технологию шифрования TLS/SSL

HTTPS

Протокол безопасной передачи данных, поддерживает технологию шифрования TLS/SSL



HTT
P



SS
L



HTTP
S

Лайвбраузинг



Вопросы



Операции над данными

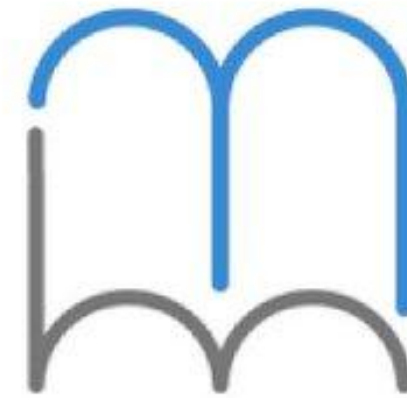


CRUD

Акроним, обозначающий четыре базовые функции, используемые при работе с данными: создание (create), чтение (read), модификация (update), удаление (delete)



CREATE



READ



UPDATE



DELETE

C

R

U

D

CRUD

Какие методы можно отнести к CRUD?

Название	Описание
GET	Получение информации с сервера
POST	Отправка информации на сервер
PATCH/PUT	Отправка информации на сервер для обновления
DELETE	Удаление информации на сервере

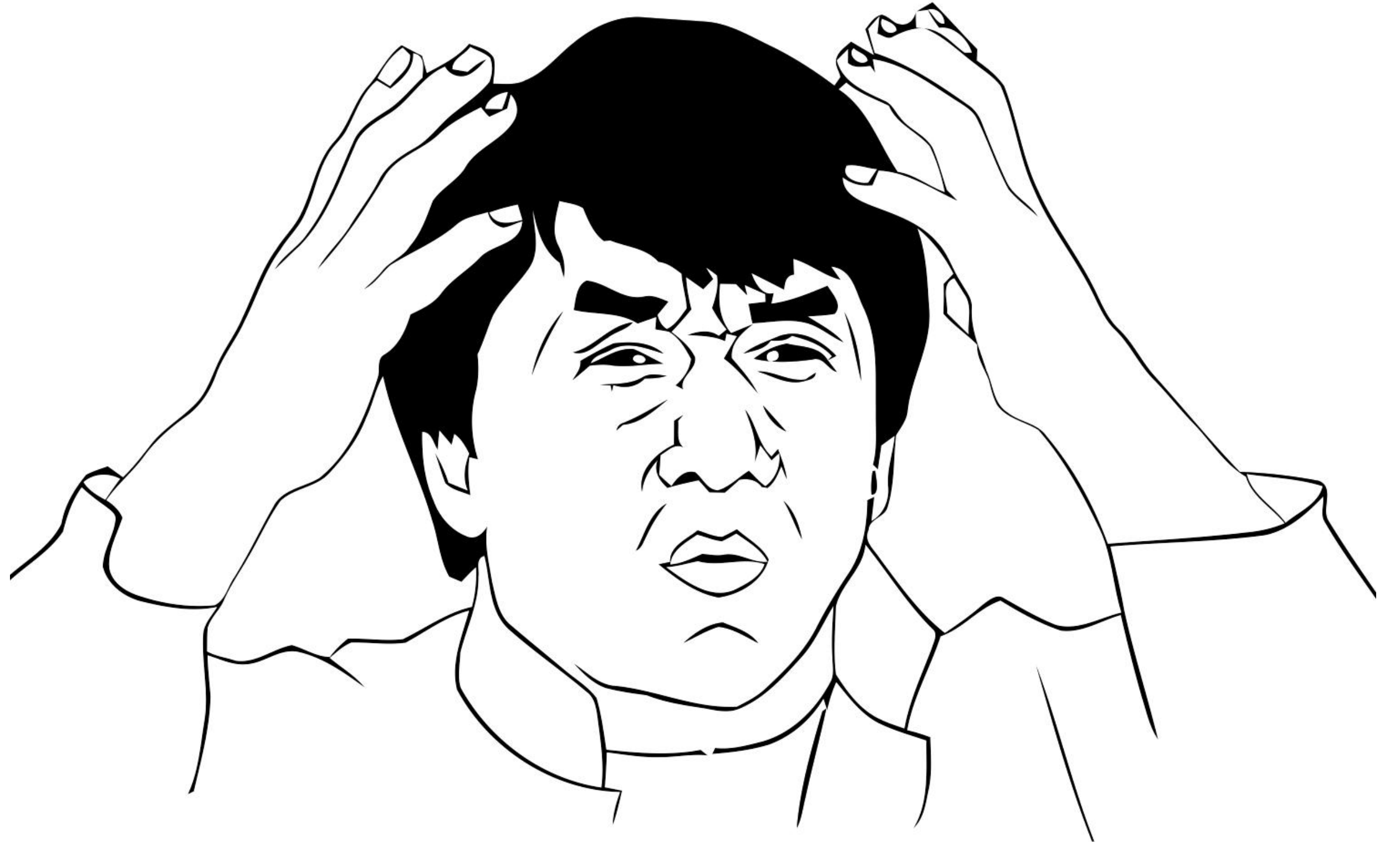
CRUD

Какие методы можно отнести к CRUD?

CRUD	Название	Описание
READ	GET	Получение информации с сервера
CREATE	POST	Отправка информации на сервер
UPDATE	PATCH/PUT	Отправка информации на сервер для обновления
DELETE	DELETE	Удаление информации на сервере

Зачем?

- Договоренности по типам методов
- Удобно проектировать API
- Шаблонный код
- Шаблонные операции



Вопросы



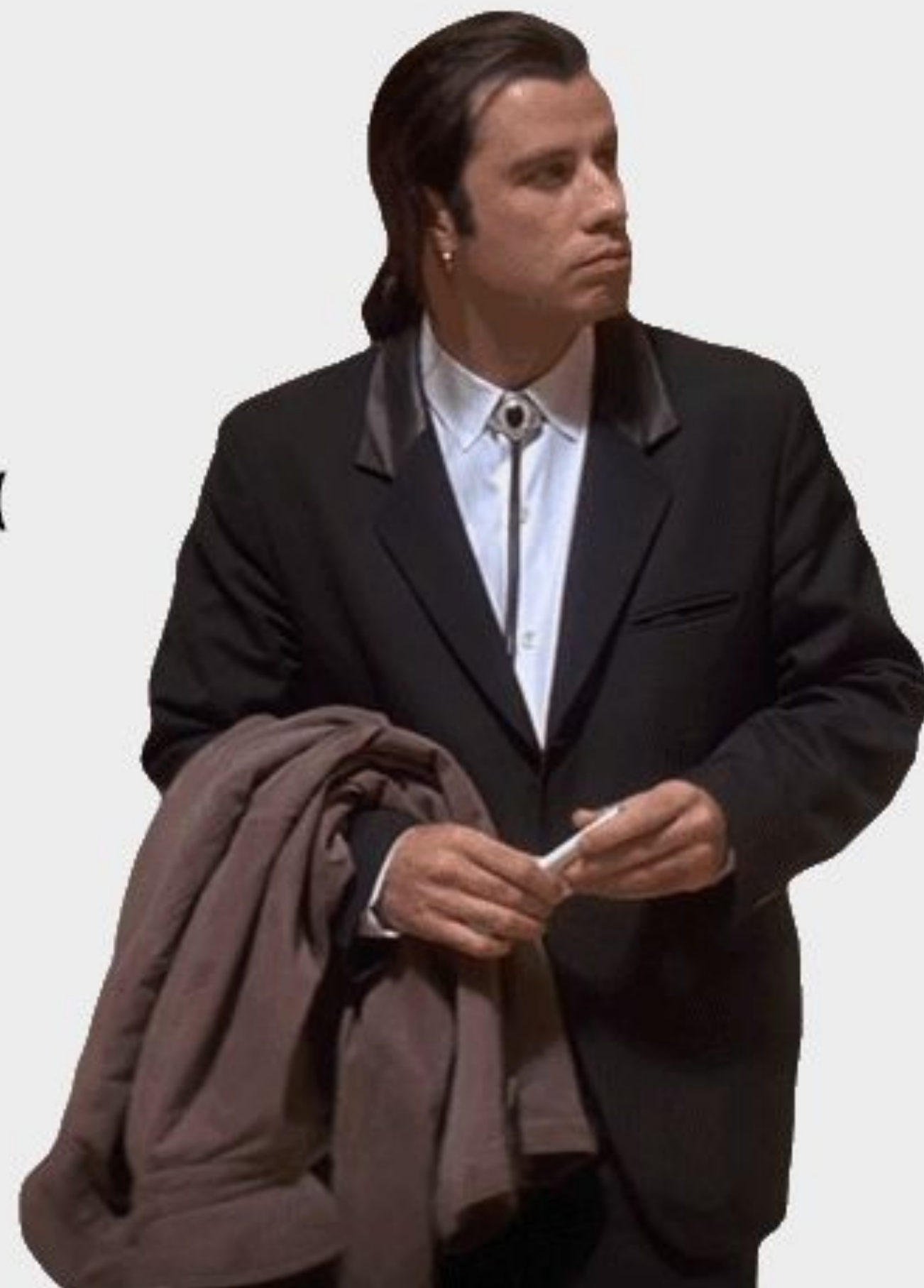
Коды ответов

Код ответа (состояния) HTTP показывает, был ли успешно выполнен определённый HTTP запрос.

404

John Travolta is lost your page:(

[Go to homepage](#)



Список кодов

Успешные 2XX

Код	Описание
200	Успешно
202	Принято в обработку



Список кодов

Клиентские ошибки 4XX

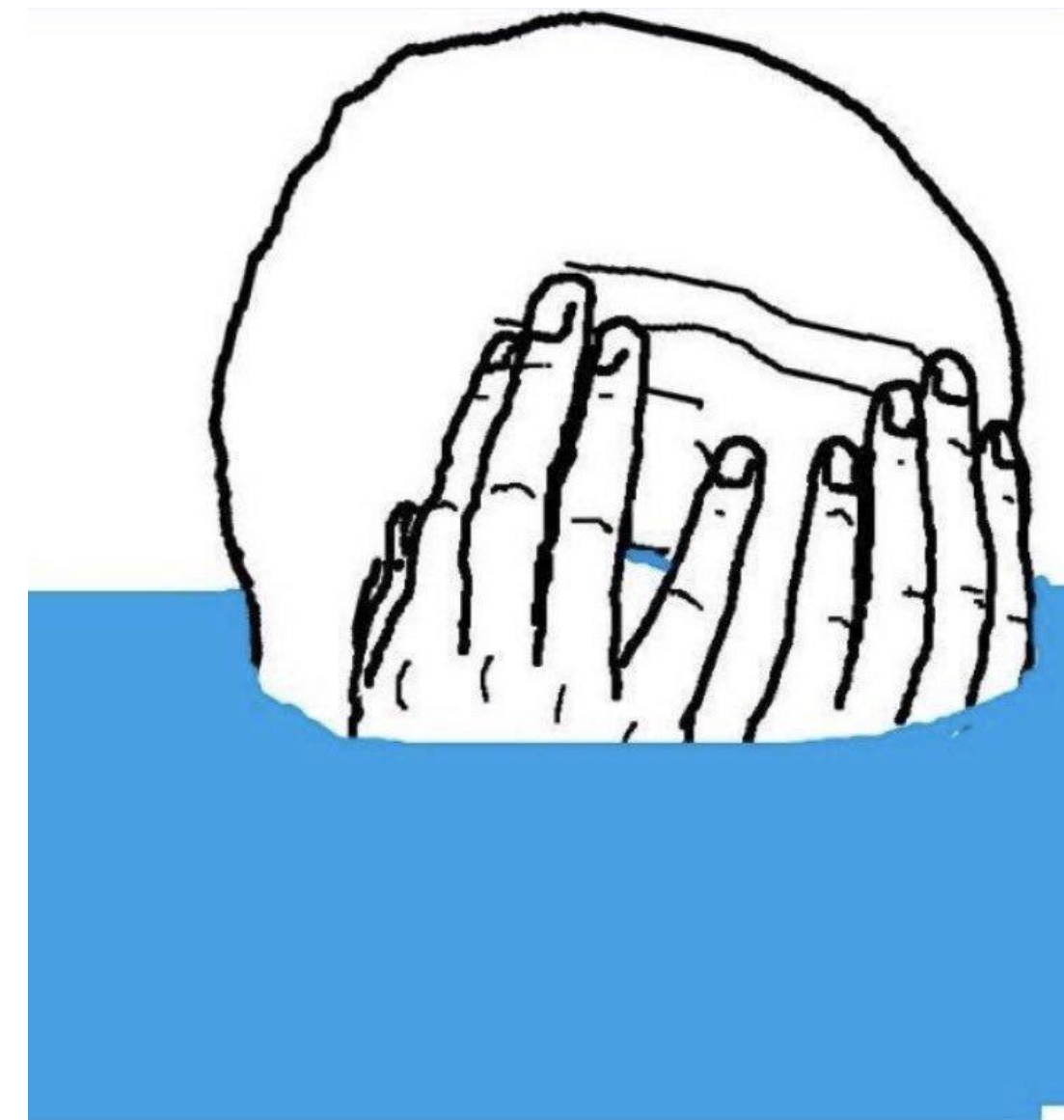
Код	Описание
400	Bad Request
401	Unauthorized
404	Не найден ресурс



Список кодов

Серверные ошибки 5XX

Код	Описание
503	Service Unavailable
504	Gateway Timeout



Вопросы



Rest



ТИНЬКОФФ

tinkoff.ru

Rest

REST (Representational State Transfer)
— архитектурный стиль
взаимодействия компонентов
распределённого приложения в сети.



REST
API

Важно помнить!

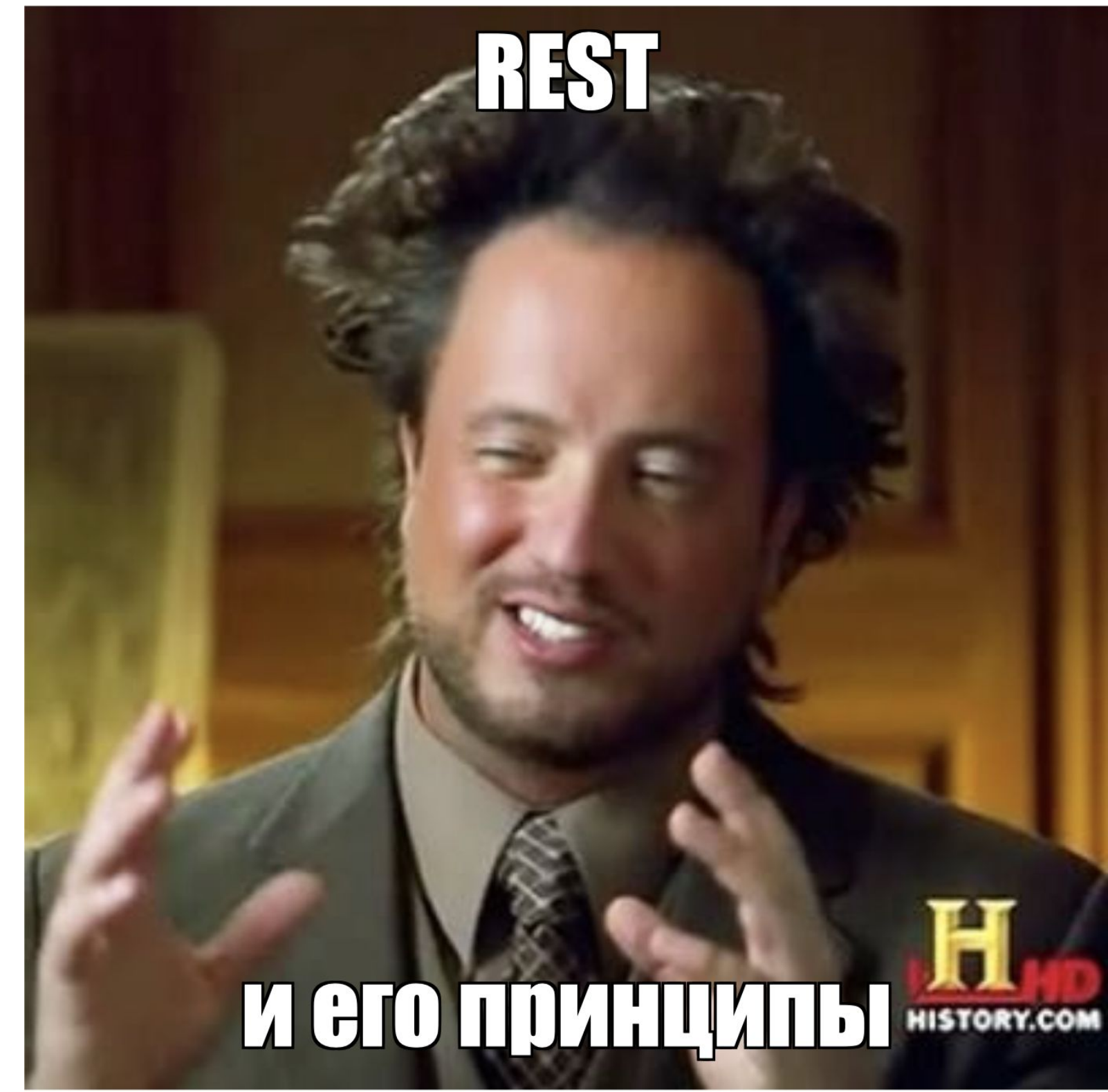
REST — это HE протокол и не стандарт

Proteæ

et

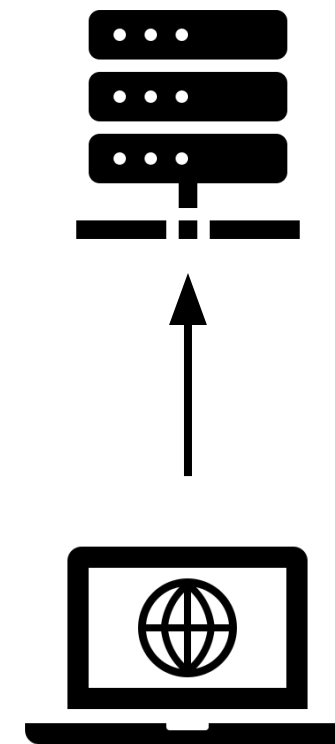
Принципы Rest

- Client-Server
- Stateless
- Cache
- Uniform Interface
- Layered System
- Code-On-Demand



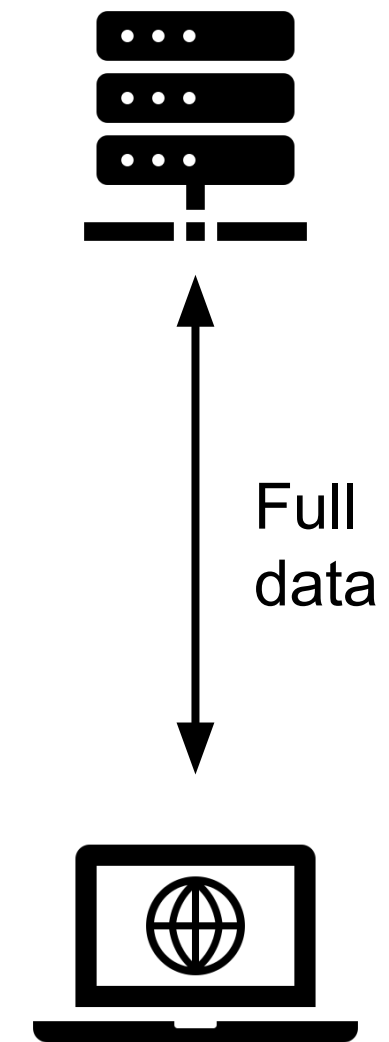
Client-Server

Все REST приложения следуют клиент серверной архитектуре



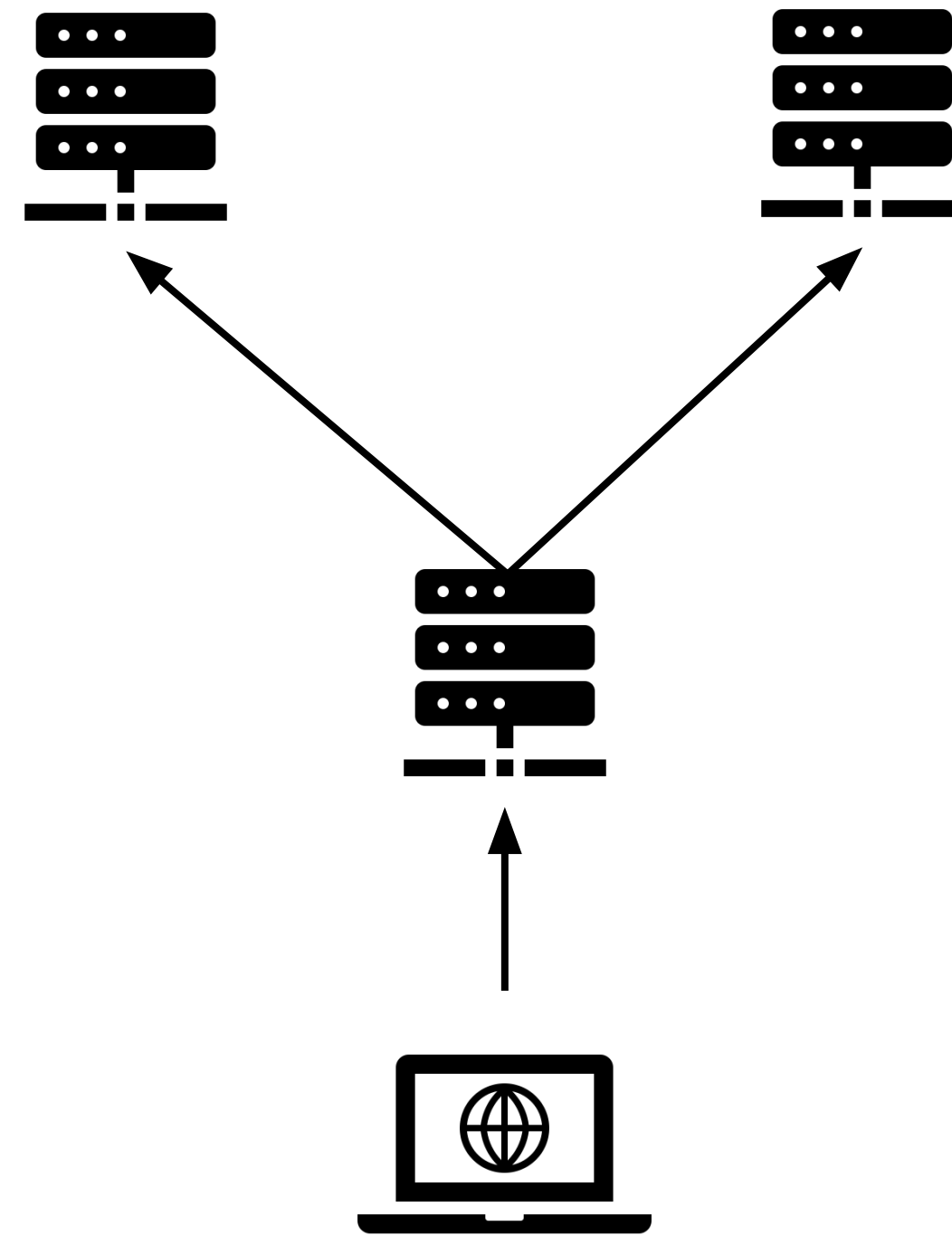
Stateless

В период между запросами клиента никакая информация о состоянии клиента на сервере не хранится



Layered System

Клиент не знает, общается он с сервером или с промежуточным сервисом



Лайвбраузинг



Домашнее задание

Нужно проверить работу методов в petstore.

Сделать коллекцию с папками по методам.

В папке с методом должны быть подпапки с проверками, каждая подпапка - это отдельная проверка.

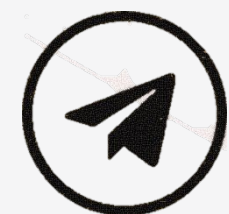
Методы, которым необходима проверка

- POST /pet
- GET /pet/{petId}
- PUT /pet
- DELETE /pet/{petId}

ЛайвРоуминг



Спасибо



Telegram: @Ivanivskiy

Ивановский Николай



ТИНЬКОФФ

tinkoff.ru