

Networking

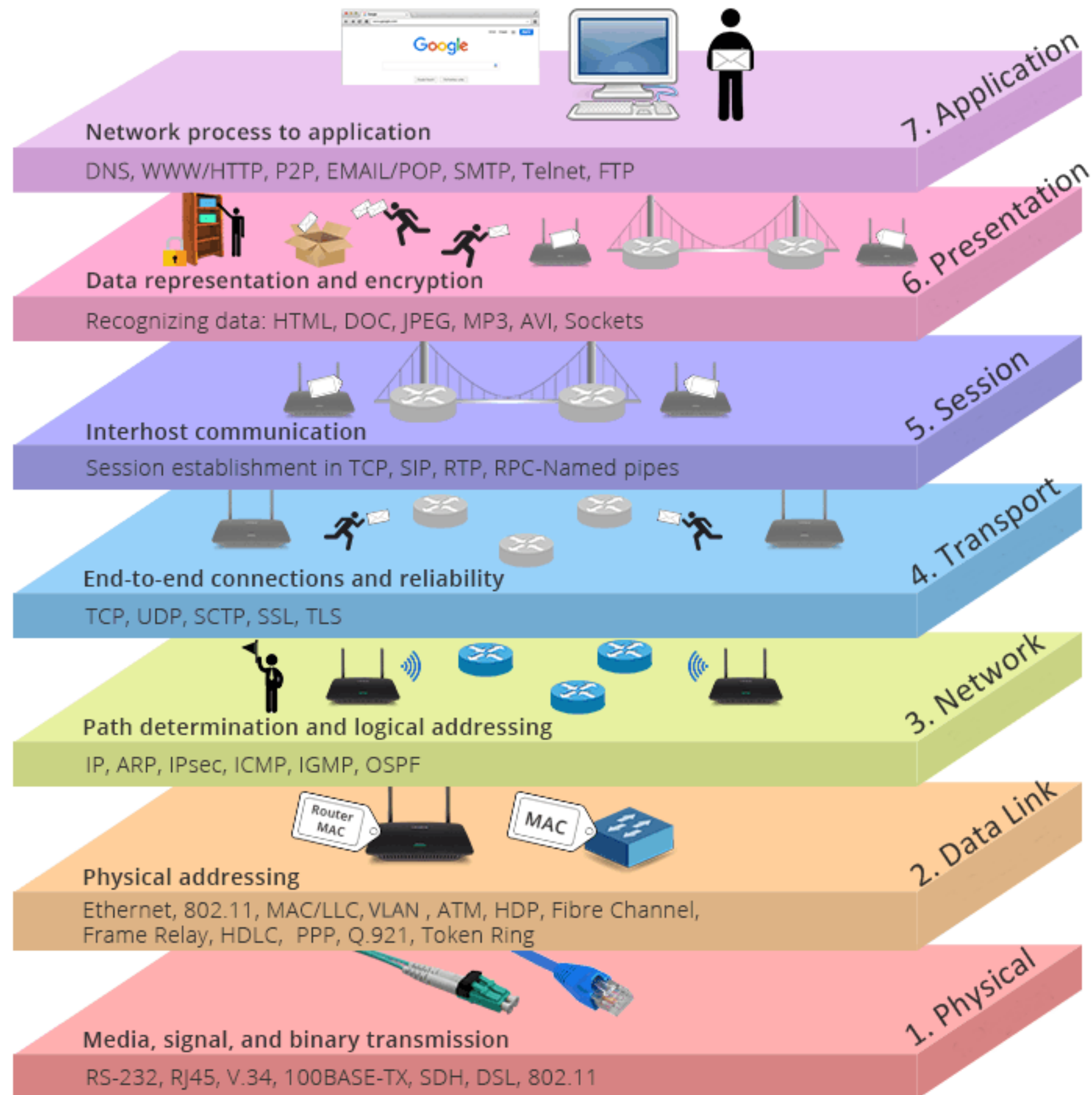
OSI, TCP, HTTP

Неделя 9



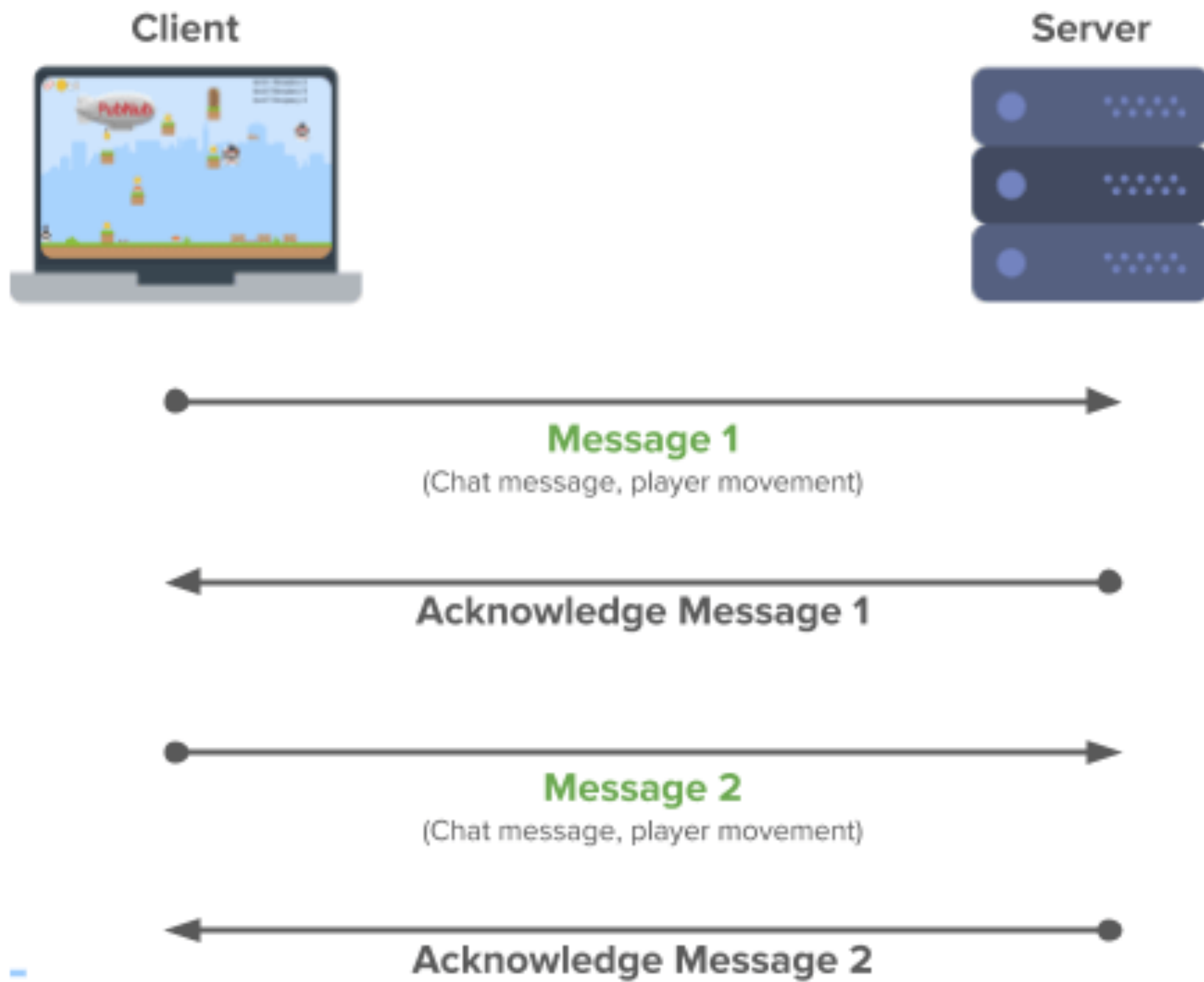
Модель OSI

1. **Физический уровень** – передача битов по каналу связи.
2. **Канальный уровень** – обеспечение надежной передачи данных.
3. **Сетевой уровень** – маршрутизация пакетов (IP, ICMP).
4. **Транспортный уровень** – управление передачей данных (TCP, UDP).
5. **Сеансовый уровень** – управление сеансами связи.
6. **Представительский уровень** – преобразование форматов данных.
7. **Прикладной уровень** – взаимодействие с приложениями (HTTP, FTP).

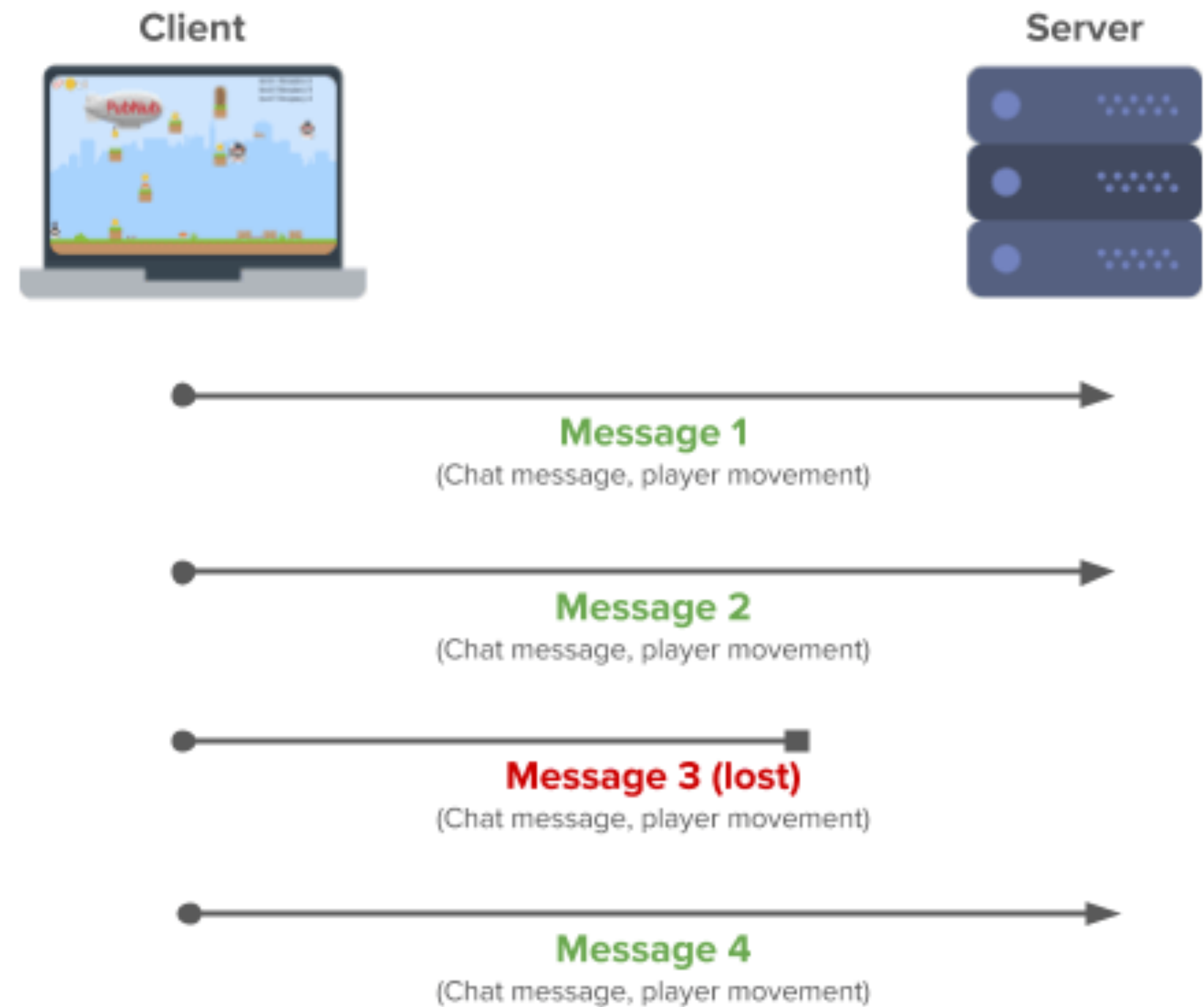


TCP vs UDP

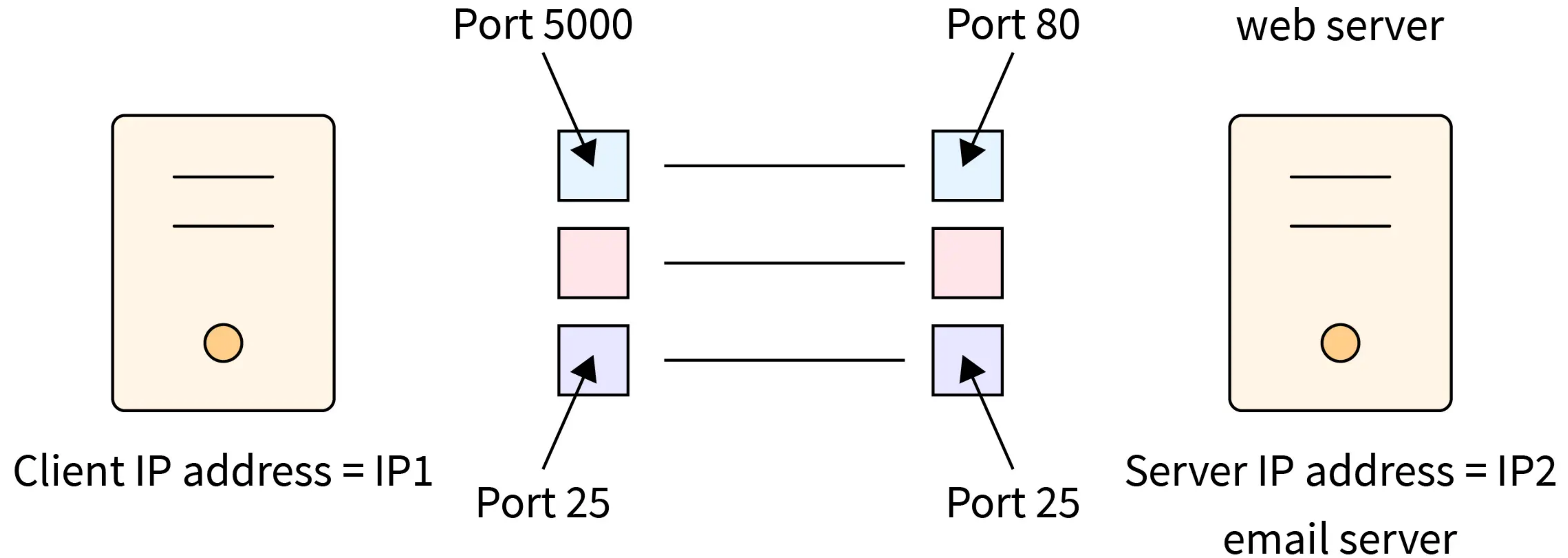
TCP



UDP



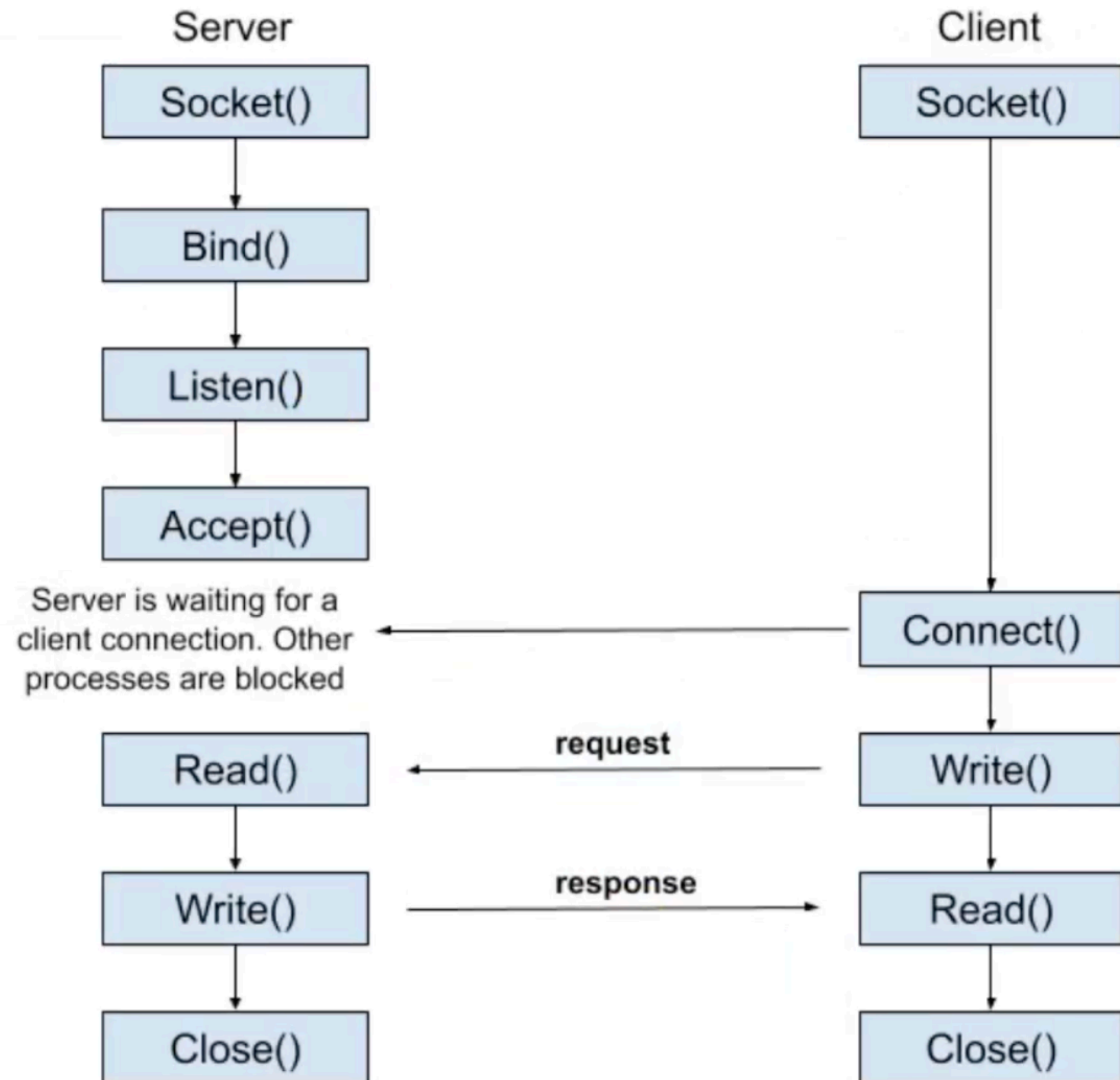
TCP/IP Port And Sockets



IP Address + Port number = Socket

System.NET.Sockets

Позволит реализовать чтение/
запись в сокет



Пример использования Socket на клиенте

```
using System.Net.Sockets;
using System.Text;

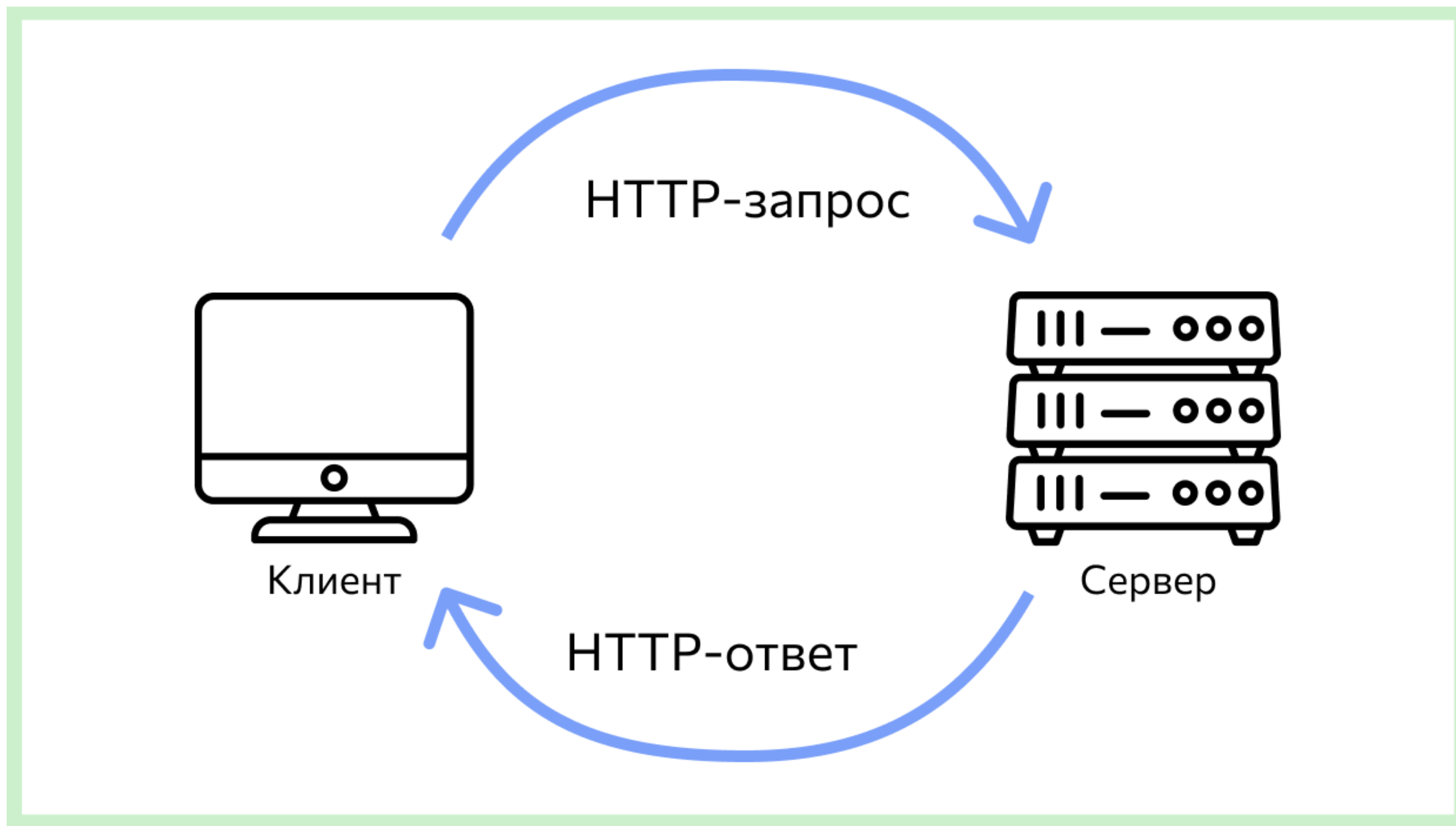
var socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
socket.Connect("hse.ru", 80);

string message = "Hello, HSE!";
byte[] dataToSend = Encoding.UTF8.GetBytes(message);
socket.Send(dataToSend);

byte[] buffer = new byte[1024];
int bytesRead = socket.Receive(buffer);
string response = Encoding.UTF8.GetString(buffer, 0, bytesRead);
Console.WriteLine(response);
```

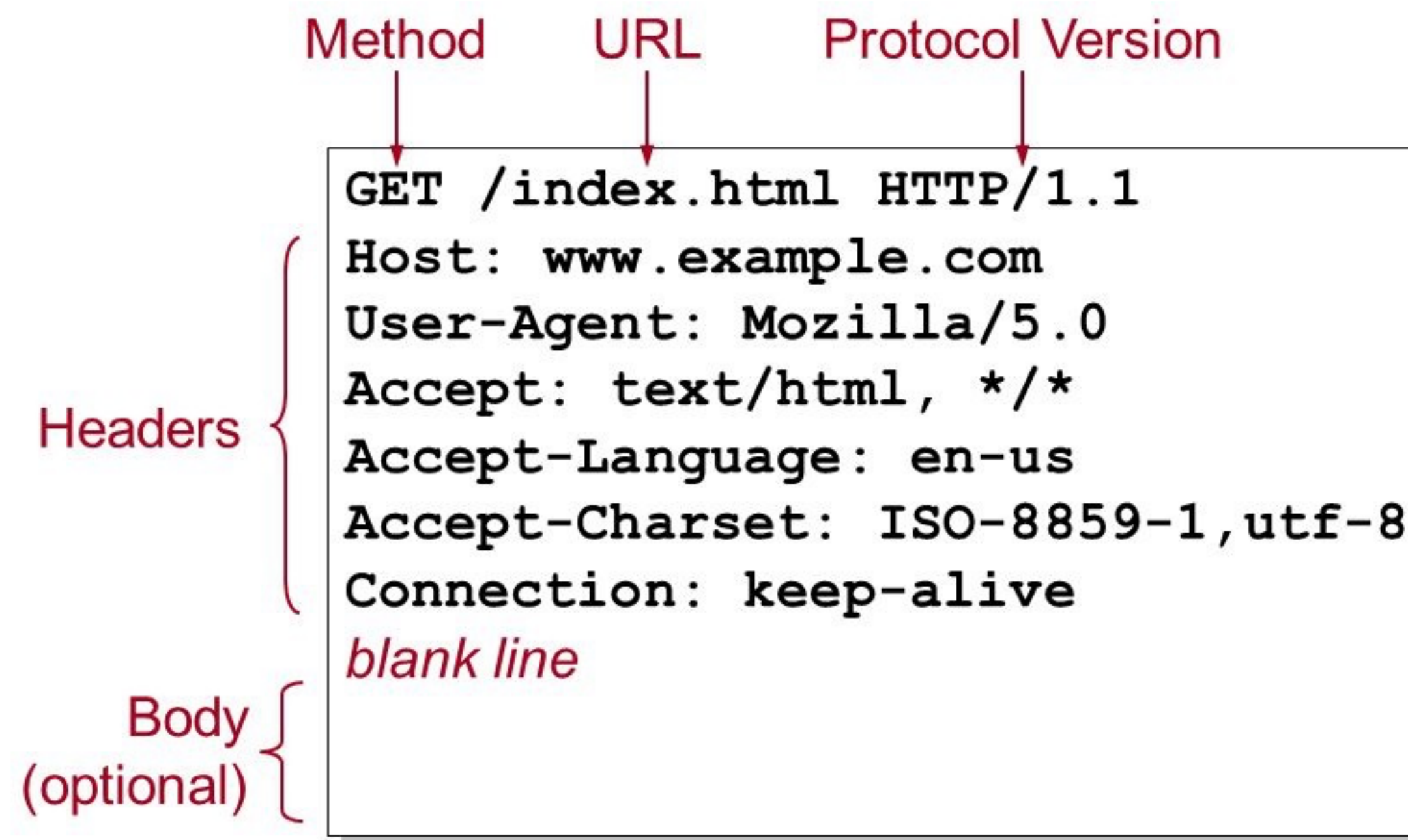
Здесь происходит создание сокета для установления TCP подключения, осуществляется передача данных, а также чтение ответа сервера

HTTP



HTTP Запросы

GET/POST/PATCH/PUT/DELETE



POST / HTTP/1.1

Host: localhost:8000

User-Agent: Mozilla/5.0 (Macintosh;...)... Firefox/51.0

Accept: text/html,application/xhtml+xml,..., */*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Content-Type: multipart/form-data; boundary=-12656974

Content-Length: 345

-12656974

(more data)

Request headers

General headers

Representation
headers

Response code

100 – 199	Информационные ответы
200 – 299	Успешное выполнение запроса
300 – 399	Редиректы
400 – 499	Клиентские ошибки
500 – 599	Ошибки на стороне сервера

Пример с HttpClient

```
using HttpClient httpClient = new HttpClient();

try
{
    string uri = "https://cs.hse.ru";

    HttpResponseMessage response = await httpClient.GetAsync(uri);

    response.EnsureSuccessStatusCode();

    string responseBody = await response.Content.ReadAsStringAsync();

    Console.WriteLine(responseBody);

    . . . .
```

Controller-Service-Repository

