

Веб-приложения на РНР

Веб-приложения

Клиент (браузер)



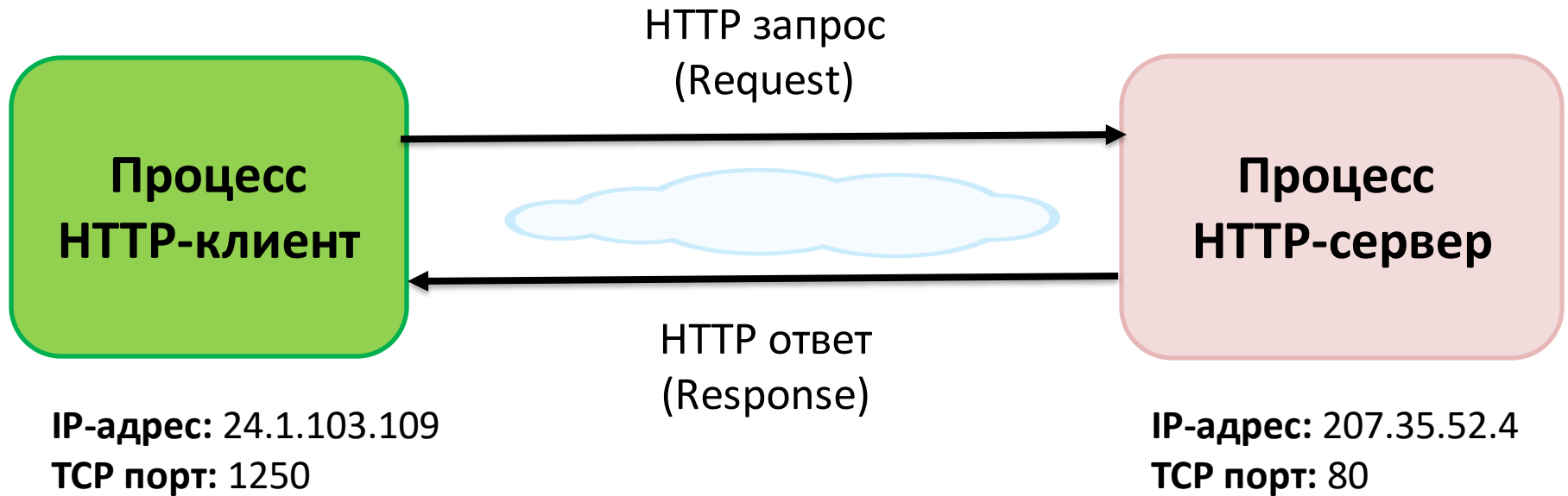
FRONTEND

Сервер



BACKEND

Hypertext Transfer Protocol (HTTP)



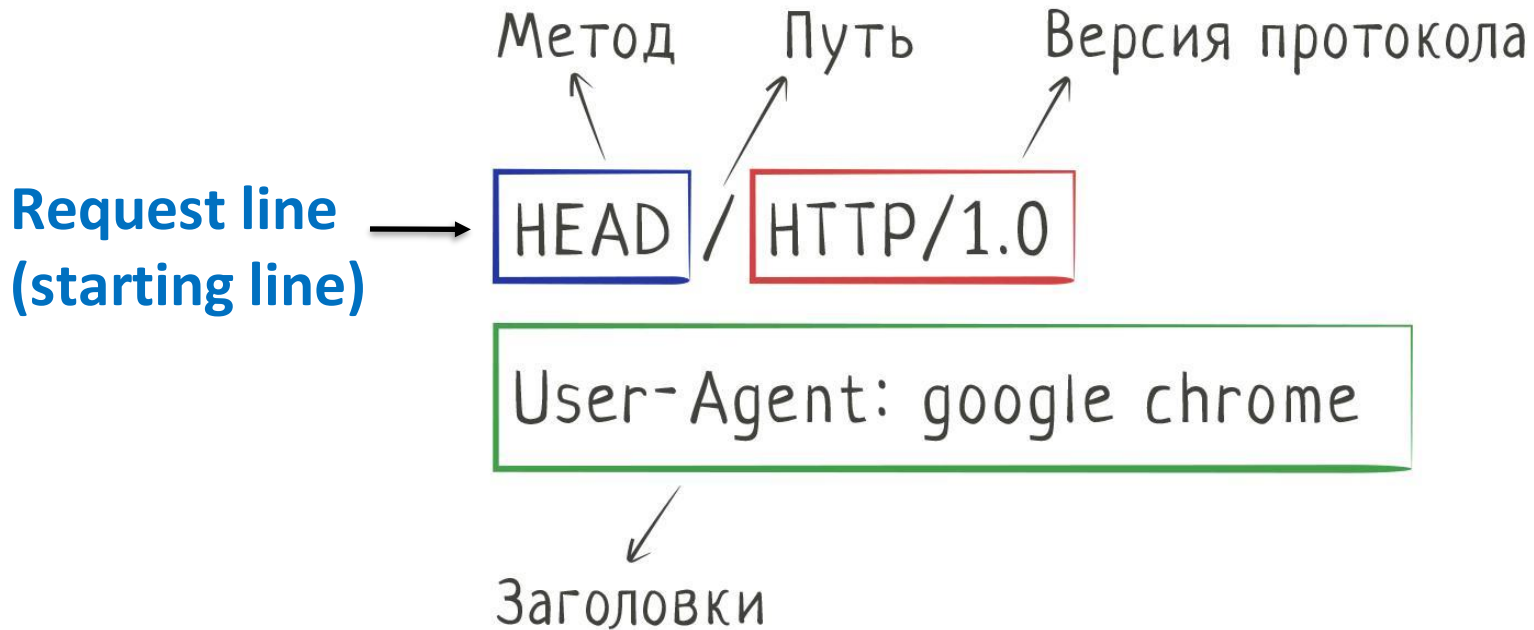
1. Клиент устанавливает TCP соединение.
2. Клиент отправляет HTTP-запрос и ждёт ответа.
3. Сервер обрабатывает запрос и посылает HTTP-ответ.

Протокол HTTP

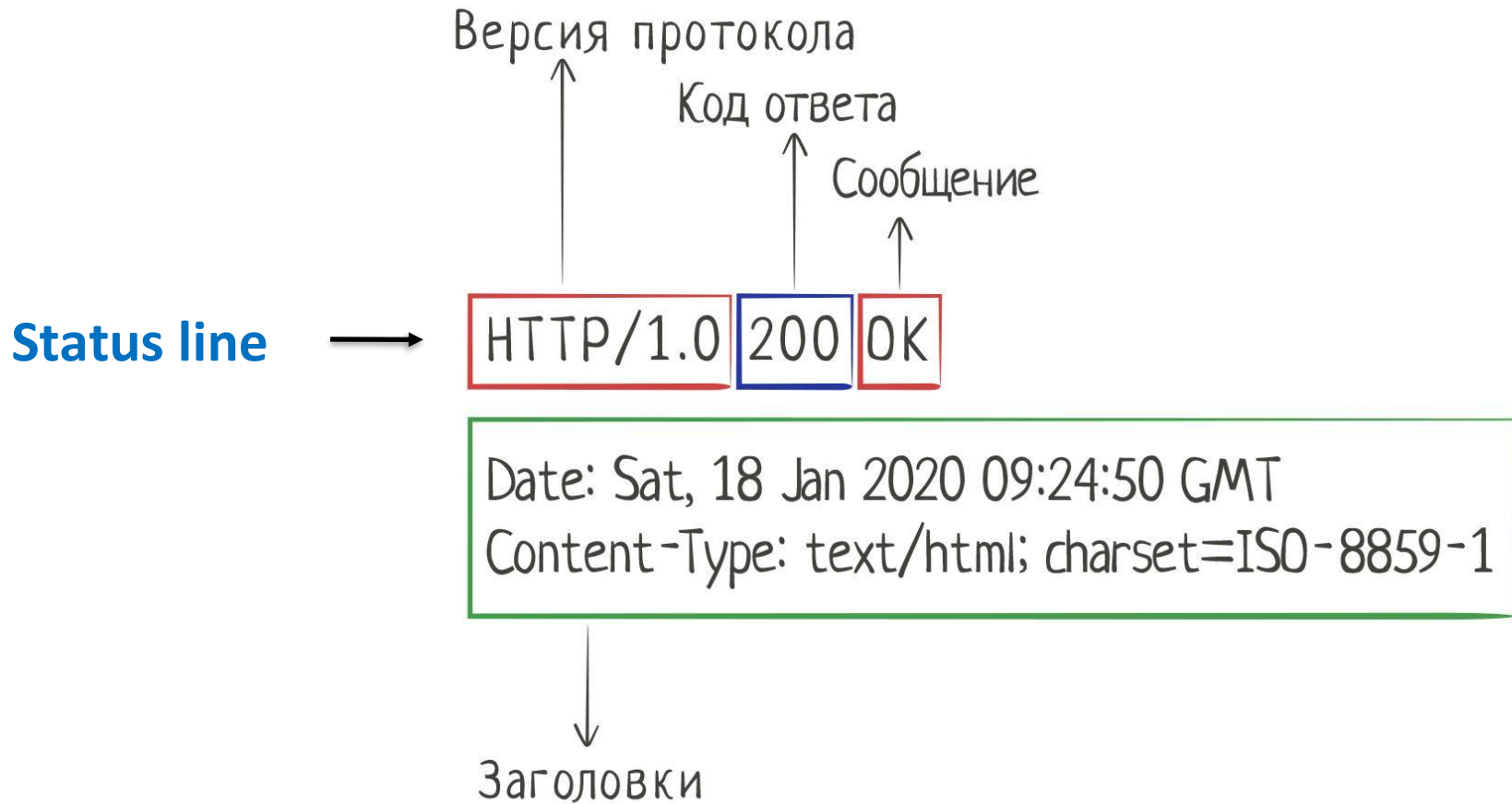
- HTTP - текстовый протокол прикладного уровня (поверх TCP)
- Процесс-клиент соединяется с процессом-сервером по протоколу TCP
- Подключение по IP-адресу и номеру порта (HTTP – 80, HTTPS – 443)

`telnet` : подключение по TCP к серверу, ввод HTTP-запроса, получение HTTP-ответа от сервера

HTTP запрос (request)



HTTP ответ (response)



HTTP ответ (response)

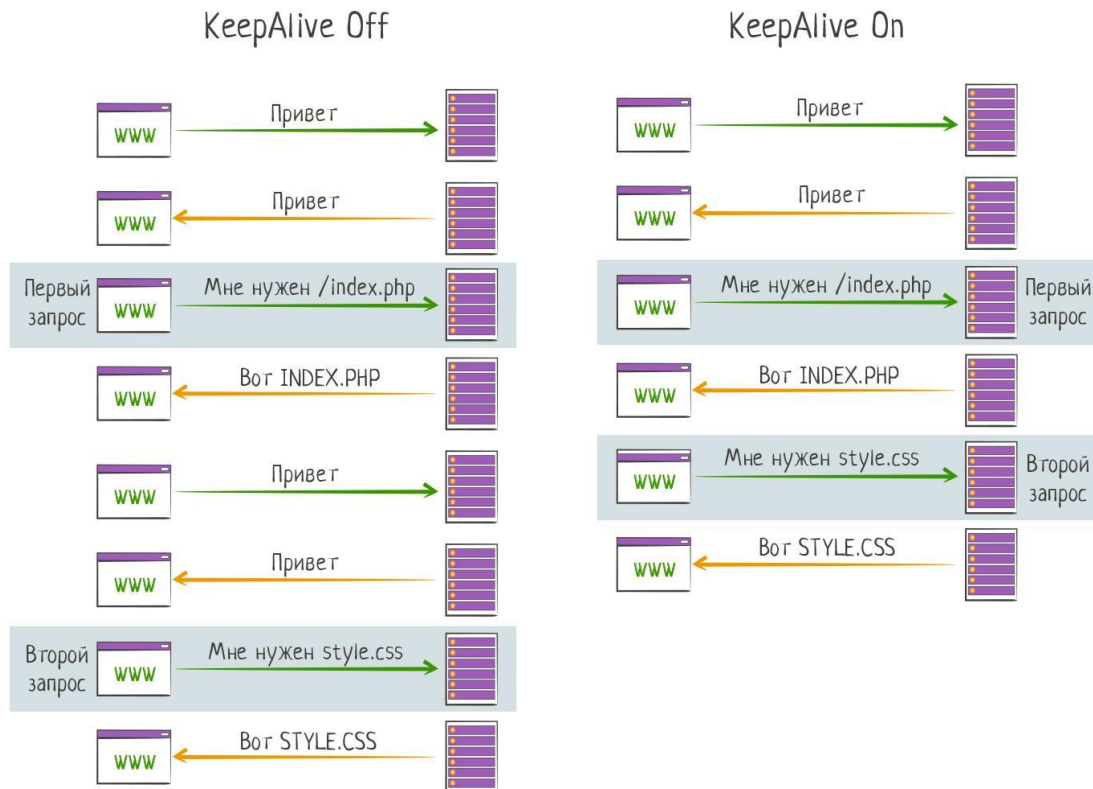
```
andrey@mono:~  
Файл Правка Вид Поиск Терминал Справка  
→ ~ telnet andpop.ru 80  
Trying 81.177.140.92...  
Connected to andpop.ru.  
Escape character is '^]'.  
Request line → GET / HTTP/1.0  
Status line → HTTP/1.1 403 Forbidden  
Заголовки → Date: Thu, 26 Nov 2020 13:37:17 GMT  
Content-Type: text/html  
Content-Length: 601  
Connection: close  
Пустая строка →  
Тело → <!DOCTYPE html><html data-page="noservice" data-version="1.1.0">  
"content-type" content="text/html; charset=utf-8"><meta http-equi  
tent="IE=edge"><title>Сайт не обслуживается</title></head><body>  
бслуживается</h1><p>Запрошенный сайт не обслуживается на хостинг  
f="https://www.jino.ru/">Джино</a></p></noscript><div id="root">  
rking-static.jino.ru/static/main.js" charset="utf-8"></script></  
closed by foreign host.
```

HTTP 1.1 запрос (request)

HEAD / HTTP/1.1

Host: andpop.ru

User-Agent: google chrome



Хост в HTTP-запросе

```
andrey@mono:~  
Файл Правка Вид Поиск Терминал Справка  
→ ~ telnet andpop.ru 80  
INSERT --  
Trying 81.177.140.92...  
Connected to andpop.ru.  
Escape character is '^]'.  
Request line → GET / HTTP/1.1  
Заголовки → Host: andpop.ru  
Status line → HTTP/1.1 200 OK  
Заголовки → Date: Thu, 26 Nov 2020 13:56:56 GMT  
Content-Type: text/html  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: Jino.ru/mod_pizza  
Accept-Ranges: bytes  
Vary: Accept-Encoding  
Пустая строка →  
f60  
<!DOCTYPE html>  
Тело → <html>  
<head>  
    <title>Андрей Попов - будни доцента</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=  
    <meta name="description" content="Андрей Попов, доцент Мордо  
суниверситета. Материалы по учебным курсам, портфолио, резюме." />
```

HTTP-методы

GET, POST, PUT, OPTIONS, HEAD, PATCH, DELETE, TRACE, CONNECT

Create (создание) - POST
Read (чтение) - GET
Update (обновление) - PUT и PATCH
Delete (удаление) - DELETE

HTTP-коды ответов

- **1xx** Информационный
- **2xx** Успешный
- **3xx** Перенаправление
- **4xx** Ошибка клиента
- **5xx** Ошибка сервера

Передача данных в HTTP-запросе

- **Query string.** Для выборки данных на сервере

```
GET /search/?text=Saransk&lr=42 HTTP/1.1  
Host: yandex.ru
```

- **Тело запроса (body).** Для изменения данных на сервере

```
POST / HTTP/1.1  
Host: example.com  
Content-Type: text/plain  
Content-Length: 28
```

```
login=user&password=12345678
```

Инструменты для работы с HTTP-запросами

- Вкладка **Network** в инструментах разработчика (браузер)
- **cURL**
- **Postman**
- `Invoke-WebRequest`, `Invoke-RestMethod` (PowerShell)
- Утилиты, встроенные в IDE (PhpStorm)

Веб-приложения



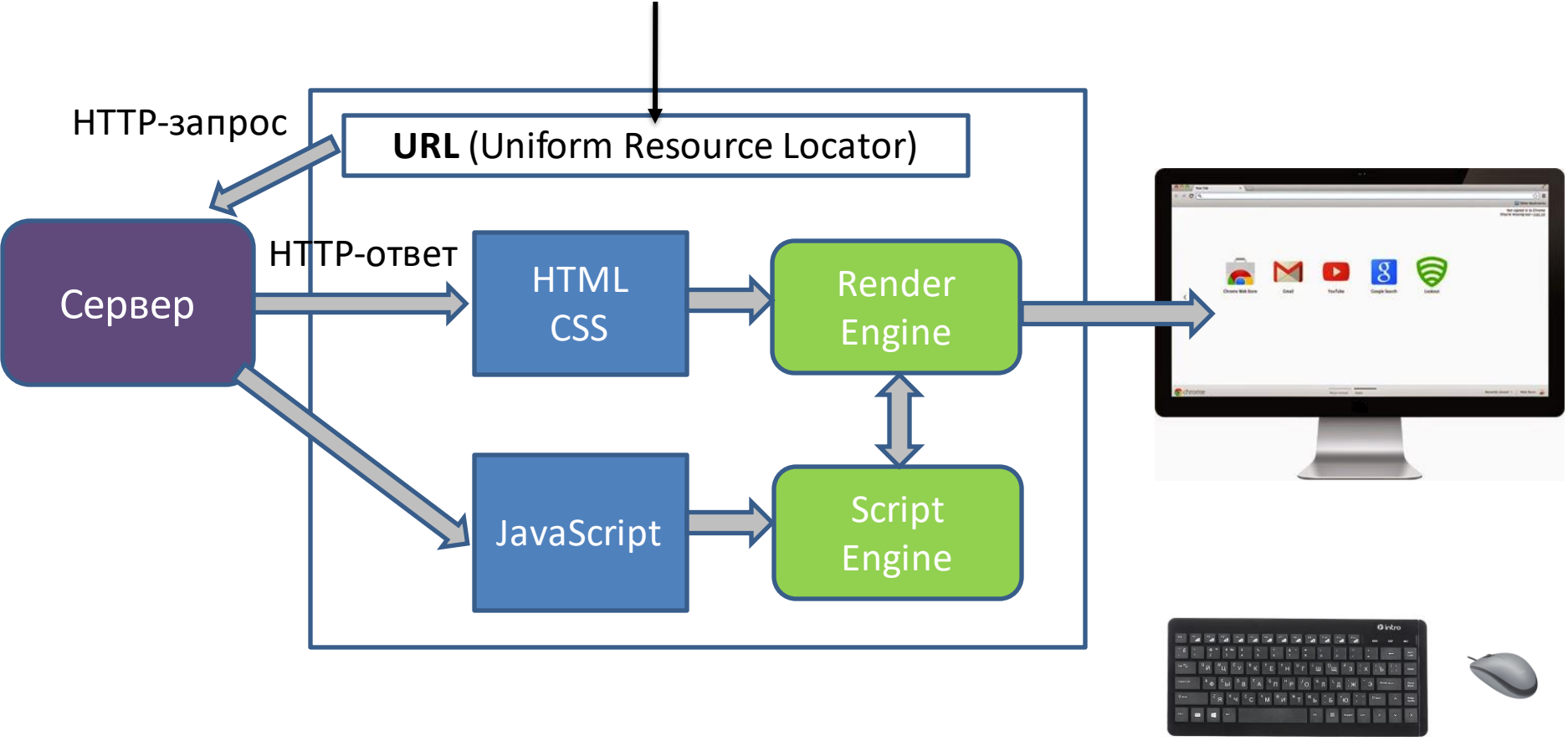
FRONTEND



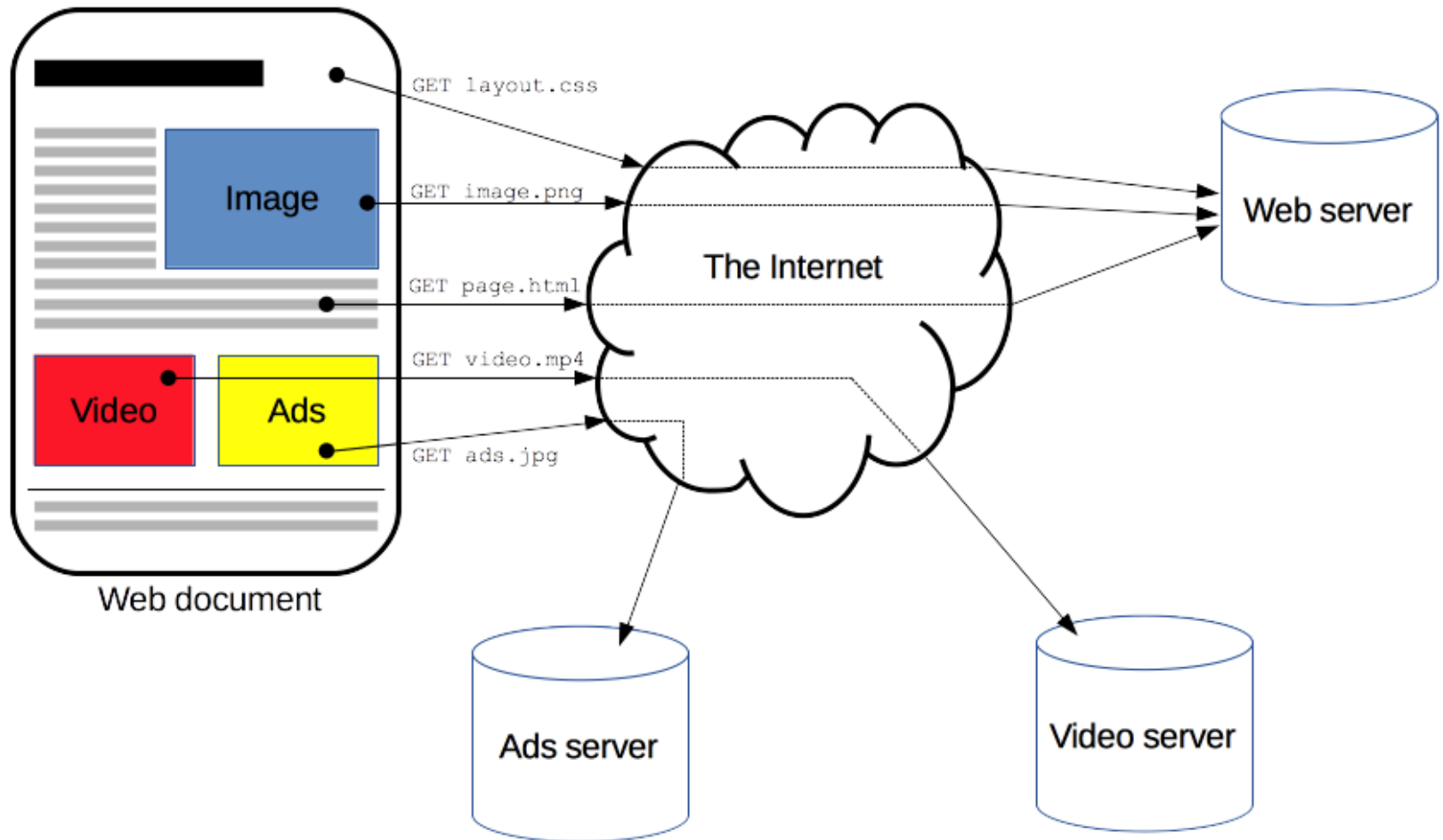
BACKEND

Frontend

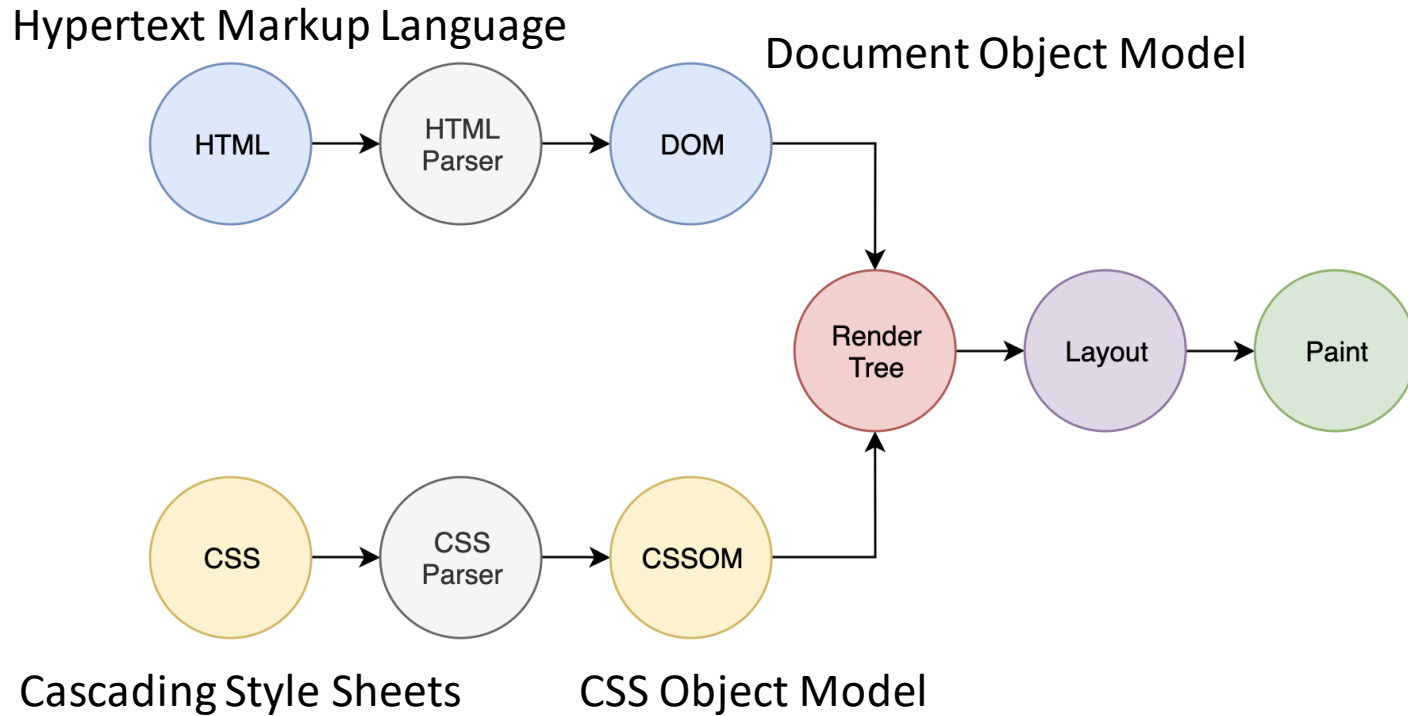
`http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2`



Документ в браузере



Рендеринг веб-страницы



DOM и стили

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:5500/00_index.html`. The page content consists of two orange horizontal bars, each containing a header: "Первый заголовок" (First header) and "Второй заголовок" (Second header). Below the page content, the browser's developer tools are open, showing the "Elements" panel on the left and the "Styles" panel on the right. The "Elements" panel shows the DOM tree with the following structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <h1>Первый заголовок</h1> == $0
    <div class="container">
      <h2>Второй заголовок</h2>
    </div>
    <!-- Code injected by live-server -->
    <script type="text/javascript">...</script>
    <div id="pocket-extension-root">...</div>
    <div id="pocket-extension-styles">...</div>
  </body>
</html>
```

The "Styles" panel shows the default user agent styles for the selected `h1` element:

```
h1 {
  display: block;
  font-size: 2em;
  margin-block-start: 0.67em;
  margin-block-end: 0.67em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
  font-weight: bold;
}
```

At the bottom of the "Styles" panel, a box model diagram is displayed, showing the dimensions of the element and its margins, border, and padding. The diagram indicates a margin of 21.440, a border, a padding, and a content area of 981.278 x 36.364.

Property	Value
margin	21.440
border	-
padding	-
Content Area	981.278 x 36.364
margin	21.440

The breadcrumb at the bottom of the developer tools shows the path: `html > body > h1`.

JavaScript в браузере

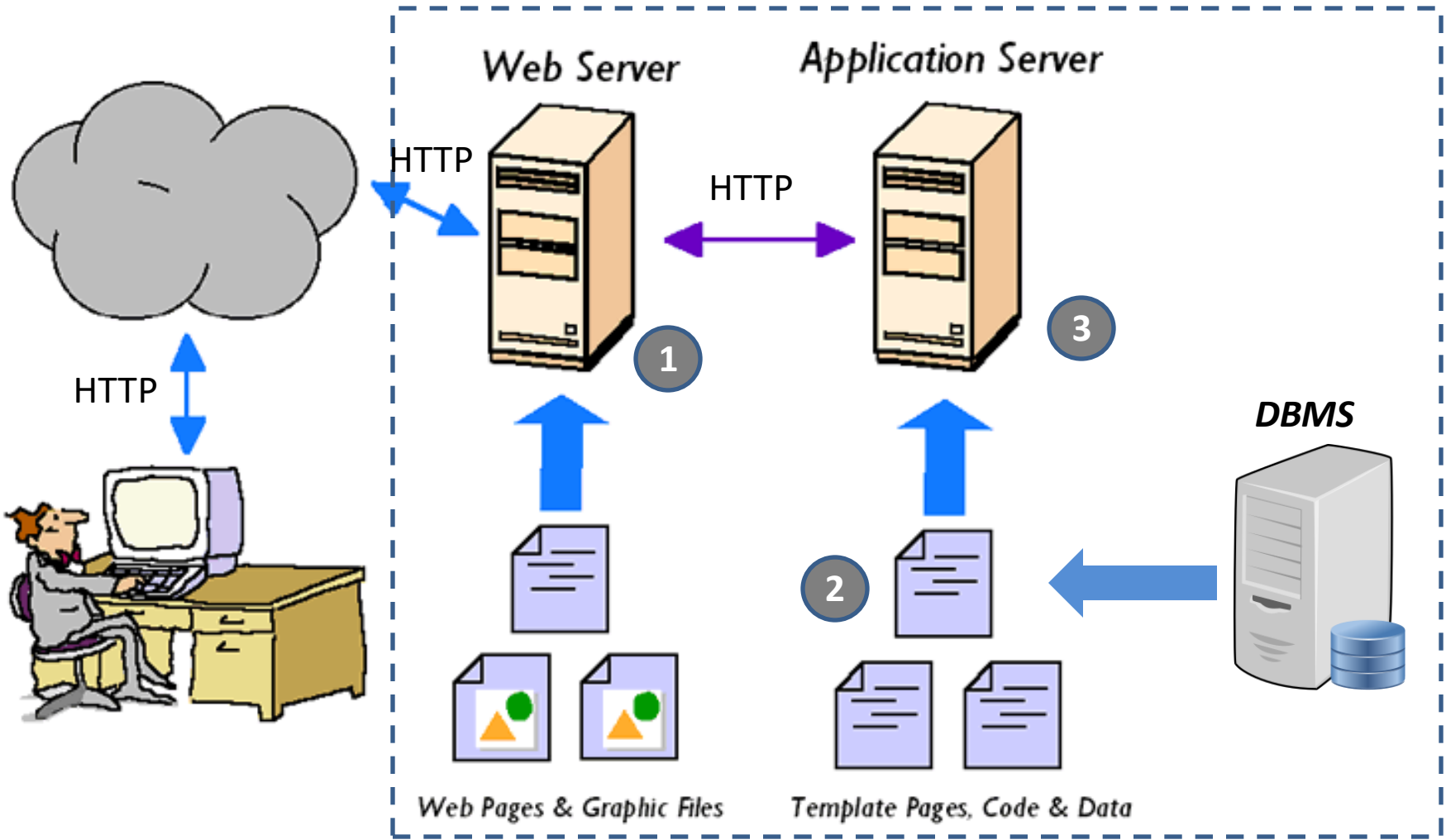
Может

- Изменять DOM, модифицировать стили
- Реагировать на действия пользователя, щелчки мыши, перемещения указателя, нажатия клавиш
- Отправлять сетевые запросы на удалённые сервера, скачивать и загружать файлы
- Запоминать данные на стороне клиента
- Получать и устанавливать куки, показывать сообщения

Не может

- Читать/записывать произвольные файлы на диске
- Напрямую работать с локальными СУБД

Backend



Создание и запуск backend-приложения

1. Установить **веб-сервер** (Nginx, Apache) для выдачи статических файлов.
2. Установить **фреймворк** или дополнительные **библиотеки для работы с HTTP** на выбранном языке (Spring для Java, ASP.NET для C#, Ruby on Rails для Ruby, Django для Python).
3. Установить **сервер приложений** (Apache Tomcat для Spring, Kestrel для ASP.NET, Puma для Ruby on Rails, Gunicorn для Java) для запуска скриптов фреймворка.
4. Настроить веб-сервер для передачи запросов на сервер приложений или запуска скриптов.
5. Установить **шаблонизатор** для генерации HTML-страниц.
6. Написать код

Создание и запуск веб-приложения на PHP

1. Установить PHP
2. Написать код



Веб внутри PHP

Для написания веб-приложения на PHP достаточно установить PHP

- В PHP встроен веб-сервер, запускающий скрипты.
- PHP - это шаблонизатор.
- В PHP встроена поддержка HTTP.

Запускаем dev-сервер

php -S localhost:5000

Файл Правка Вид Поиск Терминал Справка

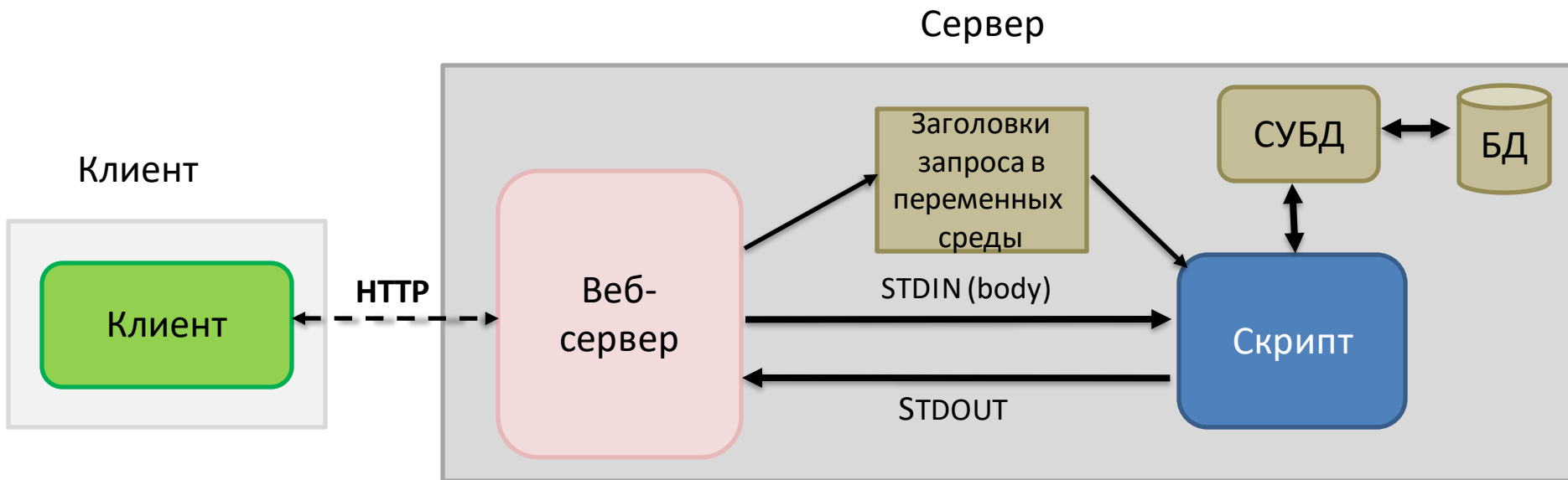
→ test `php -S localhost:5000`

-- INSERT --

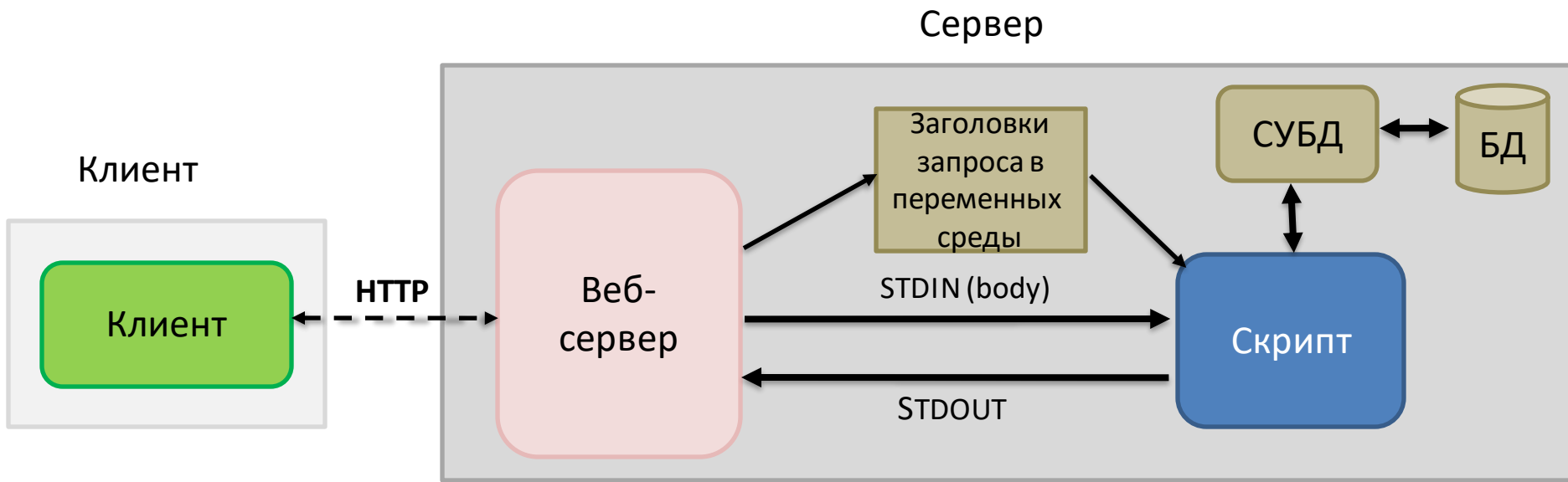
```
[Thu Nov 26 20:15:26 2020] PHP 7.4.9 Development Server (http://localhost:5000) started
[Thu Nov 26 20:15:29 2020] 127.0.0.1:34494 Accepted
[Thu Nov 26 20:15:29 2020] 127.0.0.1:34498 Accepted
[Thu Nov 26 20:15:29 2020] 127.0.0.1:34494 [200]: GET /
[Thu Nov 26 20:15:29 2020] 127.0.0.1:34494 Closing
```

Hello world на PHP

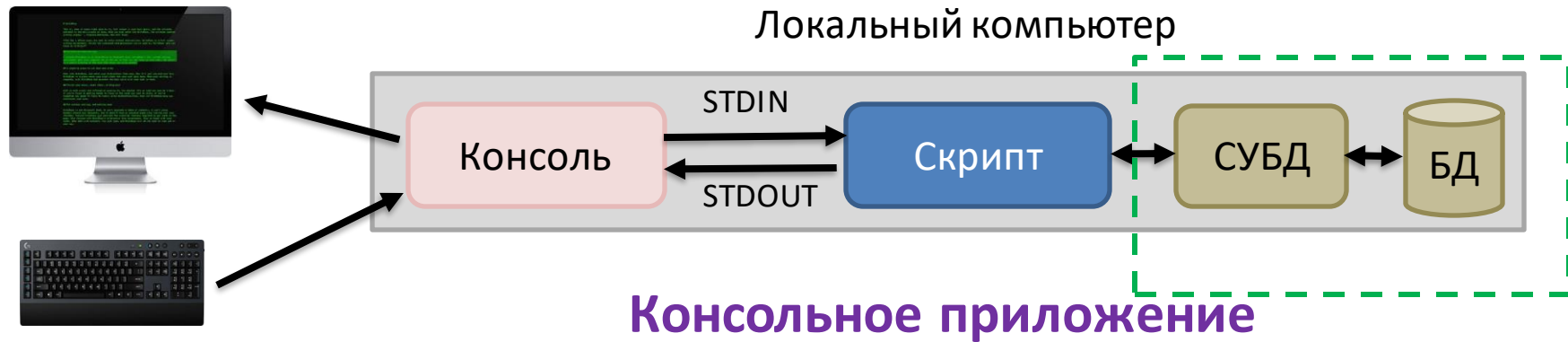




Backend-приложение (CGI-скрипт)



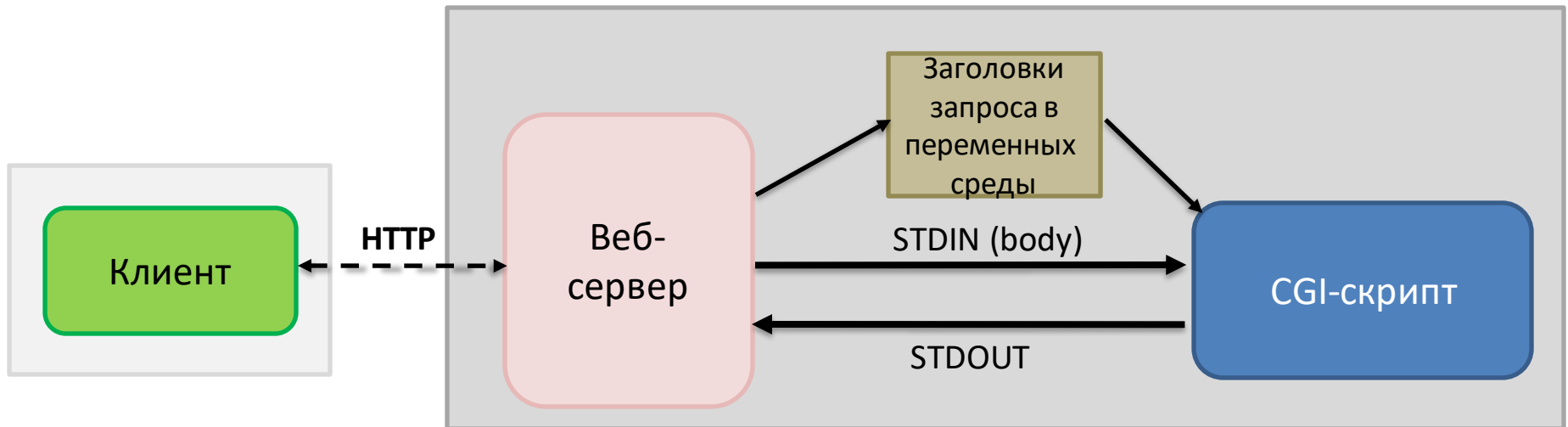
Backend-приложение (CGI-скрипт)



Веб-сервер и скрипты

```
→ ~ telnet localhost 3000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
POST /index.php?a=1&b=2 HTTP/1.1
Host: localhost:3000
Content-type: application/x-www-form-urlencoded
Content-length: 7

c=3&d=4
```



Поддержка HTTP в PHP

```
→ ~ telnet localhost 3000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
POST /index.php?a=1&b=2 HTTP/1.1
Host: localhost:3000
Content-type: application/x-www-form-urlencoded
Content-length: 7

c=3&d=4
```

```
----- $_SERVER -----
Array
(
    [DOCUMENT_ROOT] => /home/andrey/projects/test
    [REMOTE_ADDR] => 127.0.0.1
    [REMOTE_PORT] => 53028
    [SERVER_SOFTWARE] => PHP 7.4.9 Development Server
    [SERVER_PROTOCOL] => HTTP/1.1
    [SERVER_NAME] => localhost
    [SERVER_PORT] => 3000
    [REQUEST_URI] => /?a=1&b=2
    [REQUEST_METHOD] => POST
    [SCRIPT_NAME] => /index.php
    [SCRIPT_FILENAME] => /home/andrey/projects/test/index.php
    [PHP_SELF] => /index.php
    [QUERY_STRING] => a=1&b=2
    [HTTP_HOST] => localhost:3000
    [CONTENT_TYPE] => application/x-www-form-urlencoded
    [HTTP_CONTENT_TYPE] => application/x-www-form-urlencoded
    [CONTENT_LENGTH] => 7
    [HTTP_CONTENT_LENGTH] => 7
    [REQUEST_TIME_FLOAT] => 1606400111.5033
    [REQUEST_TIME] => 1606400111
)
```

```
----- $_GET -----
Array
(
    [a] => 1
    [b] => 2
)
----- $_POST -----
Array
(
    [c] => 3
    [d] => 4
)
```

Альтернативный синтаксис PHP

```
<?php echo $string; ?>
```

```
<?= $string ?>
```

```
<?php foreach ($array as $item): ?>
```

```
<li><?=$item?></li>
```

```
<?php endforeach; ?>
```

```
<?php if ($item > 10): ?>
```

Значение больше десяти.

```
<?php elseif ($item < 10): ?>
```

Значение меньше десяти.

```
<?php else: ?>
```

Значение равно десяти.

```
<?php endif; ?>
```

```
<?php while ($item < 10): ?>
```

```
<li><?=$item ?></li>
```

```
<?php endwhile; ?>
```

```
<?php for ($item = 1; $item <= 10; $item ++): ?>
```

```
<li><?=$item ?></li>
```

```
<?php endfor; ?>
```

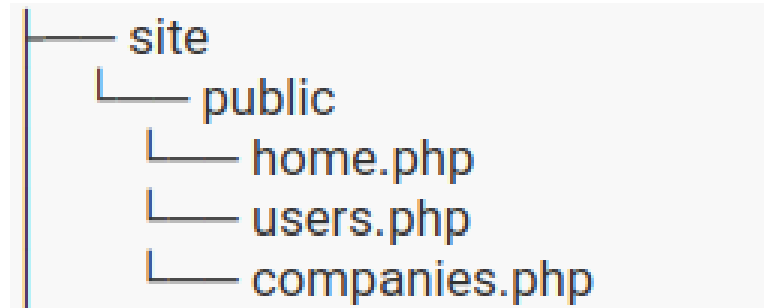
Два подхода

- **Page Controller**

example.com/home.php

example.com/users.php?action=new

example.com/companies.php?id=2



- **Front Controller**

example.com/home

example.com/users/new

example.com/companies/2

