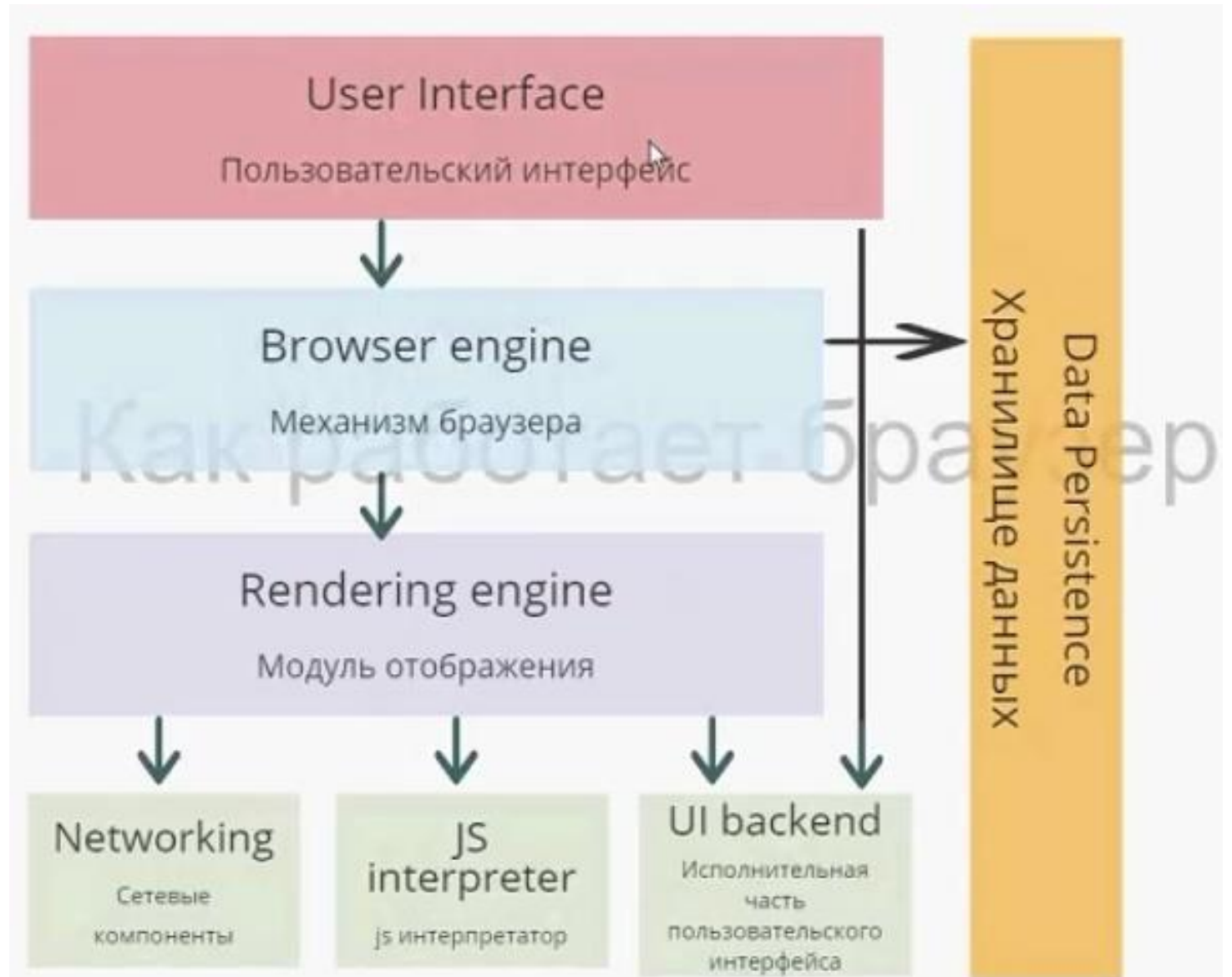


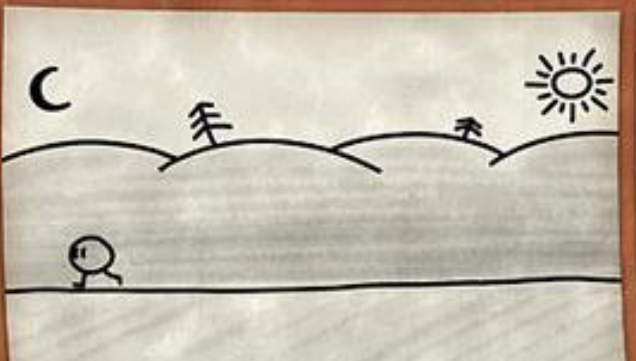
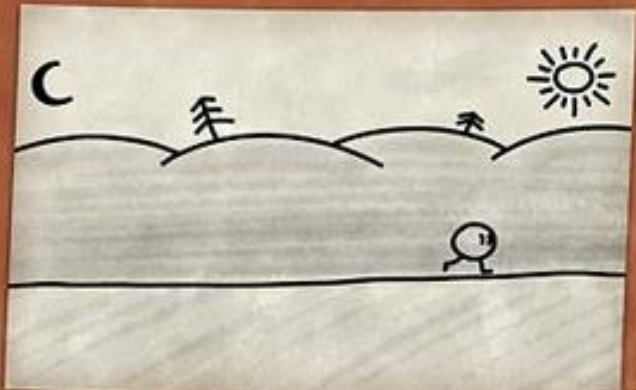
От запроса до  
отрисовки

# О браузерах

# Из чего СОСТОИТ

- **Пользовательский интерфейс** — то, что видит перед собой пользователь
- **Механизм браузера** - взаимодействие пользовательского интерфейса и модуля отображения, а также за сохранение данных в памяти.
- **Модуль отображения** - отображение информации на экране.



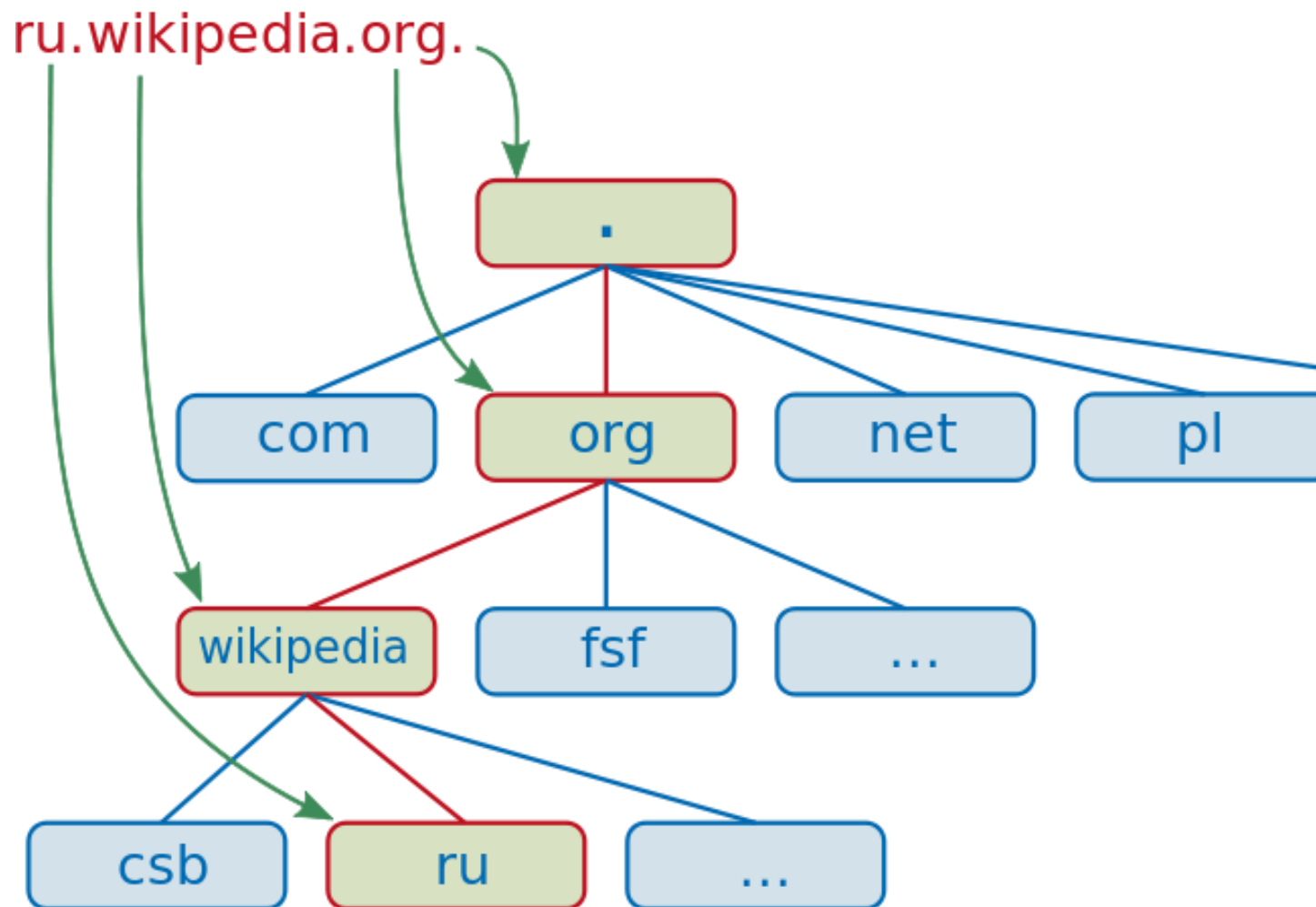


# От запроса до отрисовки

- DNS
- HTTP / HTTPS
- Sockets
- Parsing
- Rendering

# DNS

**Domain Name System** — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства)



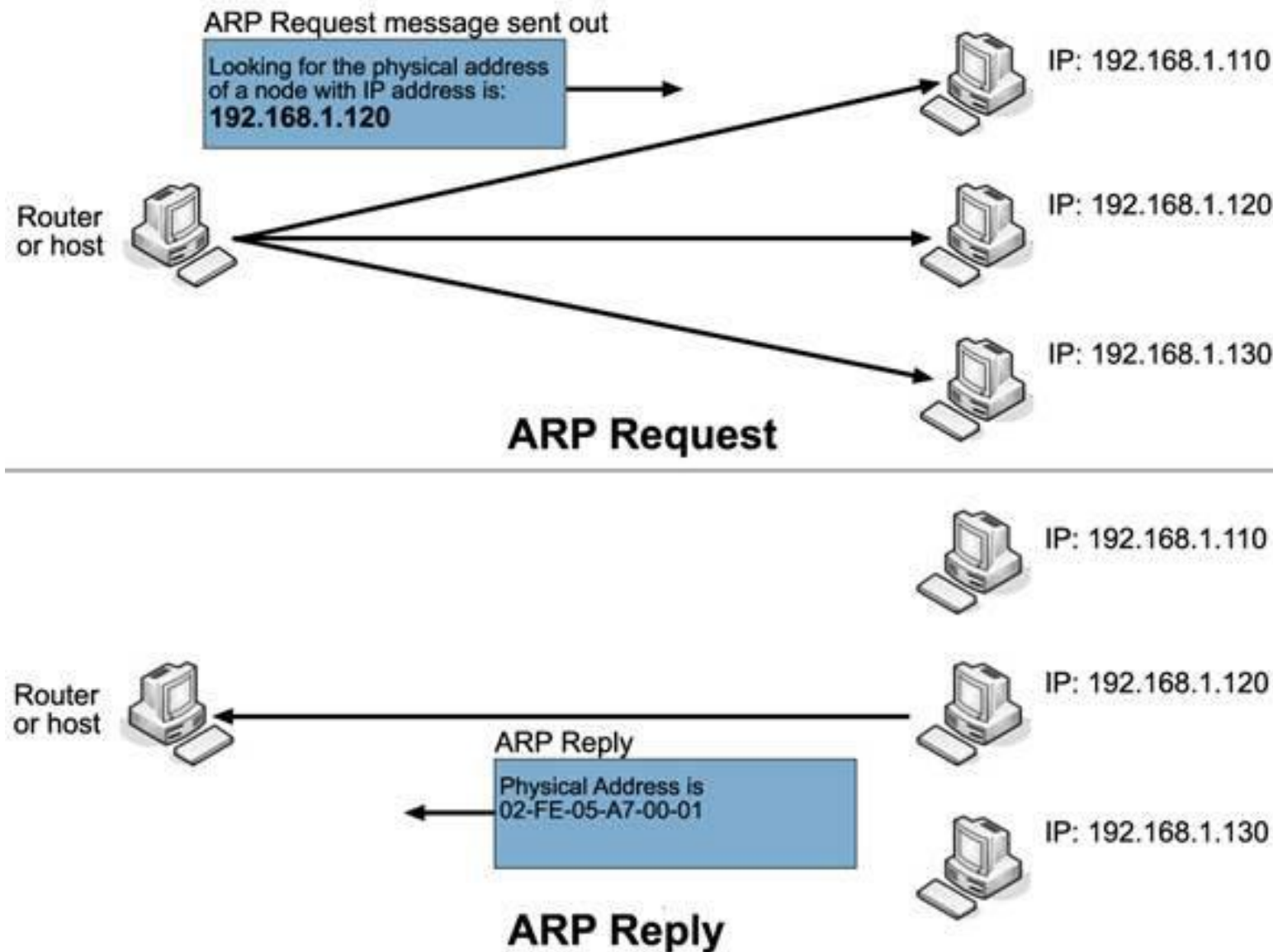
# DNS

- Браузер проверяет наличие домена в своём кэше
- Если домена там нет, то браузер вызывает библиотечную функцию *gethostbyname* (отличается в разных ОС) для поиска нужного адреса
- Прежде, чем искать домен по DNS *gethostbyname* пытается найти нужный адрес в файле *hosts* (его расположение отличается в разных ОС)
- Если домен нигде не закеширован и отсутствует в файле *hosts*, *gethostbyname* отправляет запрос к сетевому DNS-серверу. Как правило, это локальный роутер или DNS-сервер интернет-провайдера
- Если DNS-сервер находится в той же подсети, то **ARP-запрос** отправляется этому серверу
- Если DNS-сервер находится в другой подсети, то **ARP-запрос** отправляется на IP-адрес шлюза по умолчанию (default gateway)



# ARP

**Address Resolution Protocol** — протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения MAC-адреса по известному IP-адресу.



# DNS

- ARP-ответ:  
Sender MAC: target:mac:address:here  
Sender IP: target.ip.goes.here  
Target MAC: interface:mac:address:here  
Target IP: interface.ip.goes.here

Теперь у сетевой библиотеки есть IP-адрес либо DNS-сервера либо шлюза по умолчанию, который можно использовать для разрешения доменного имени:

- Порт 53 открывается для отправки UDP-запроса к DNS-серверу (если размер ответа слишком велик, будет использован TCP).
- Если локальный или на стороне провайдера DNS-сервер «не знает» нужный адрес, то запрашивается рекурсивный поиск, который проходит по списку вышестоящих DNS-серверов, пока не будет найдена SOA-запись, а затем возвращается результат.



# 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).

## Request

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request Message Header

A blank line separates header & body

Request Message Body

## Response

```
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

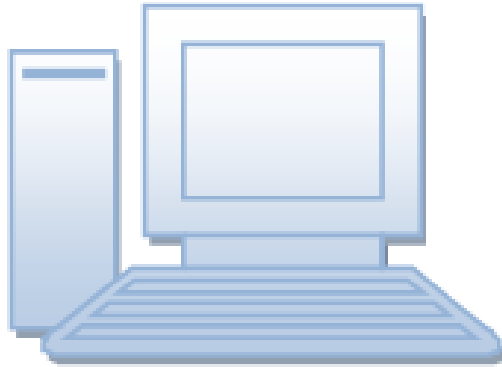
Response Message Header

A blank line separates header & body

Response Message Body

# HTTP

- (1) User issues URL from a browser  
http://host:port/path/file



- (5) Browser formats the response  
and displays

**Client** (Browser)

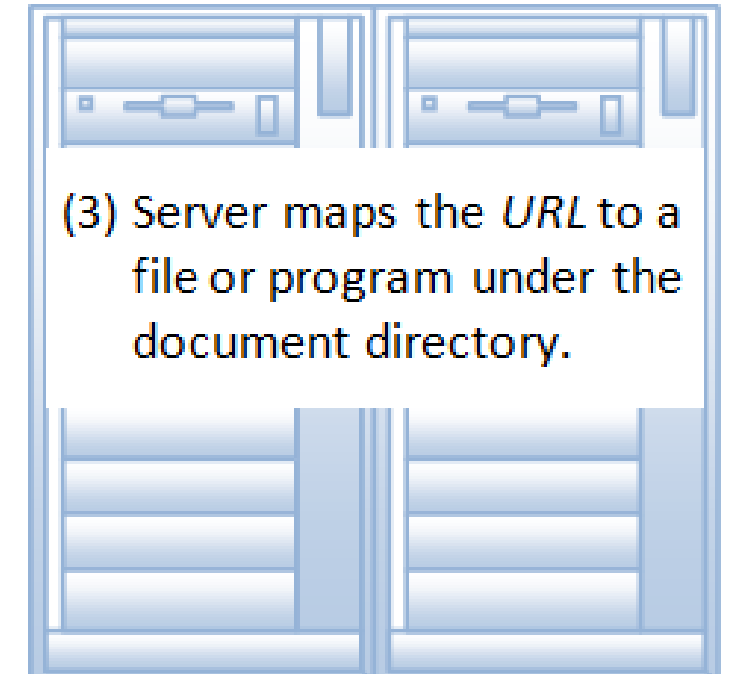
- (2) Browser sends a request message

GET *URL* HTTP/1.1  
Host: *host:port*  
.....  
.....

- (4) Server returns a response message

HTTP/1.1 200 OK  
.....  
.....  
.....

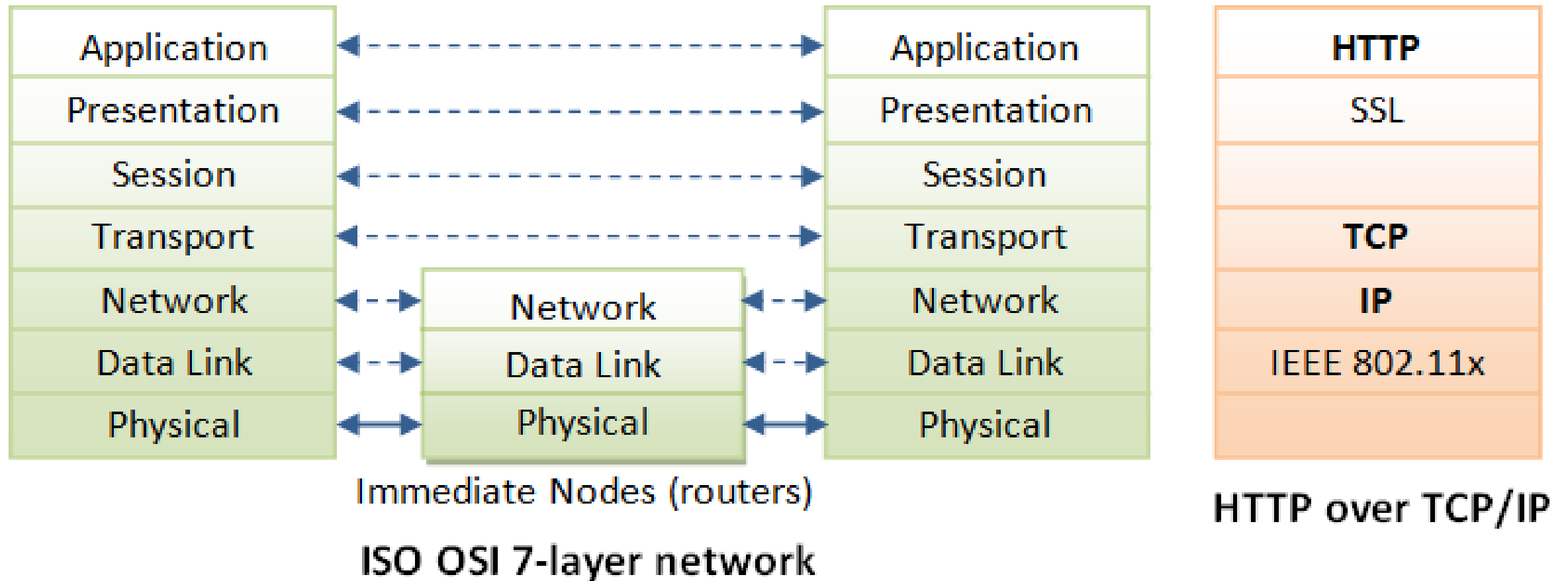
**HTTP** (Over TCP/IP)



- (3) Server maps the *URL* to a  
file or program under the  
document directory.

**Server** (@ *host:port*)

# Модель OSI



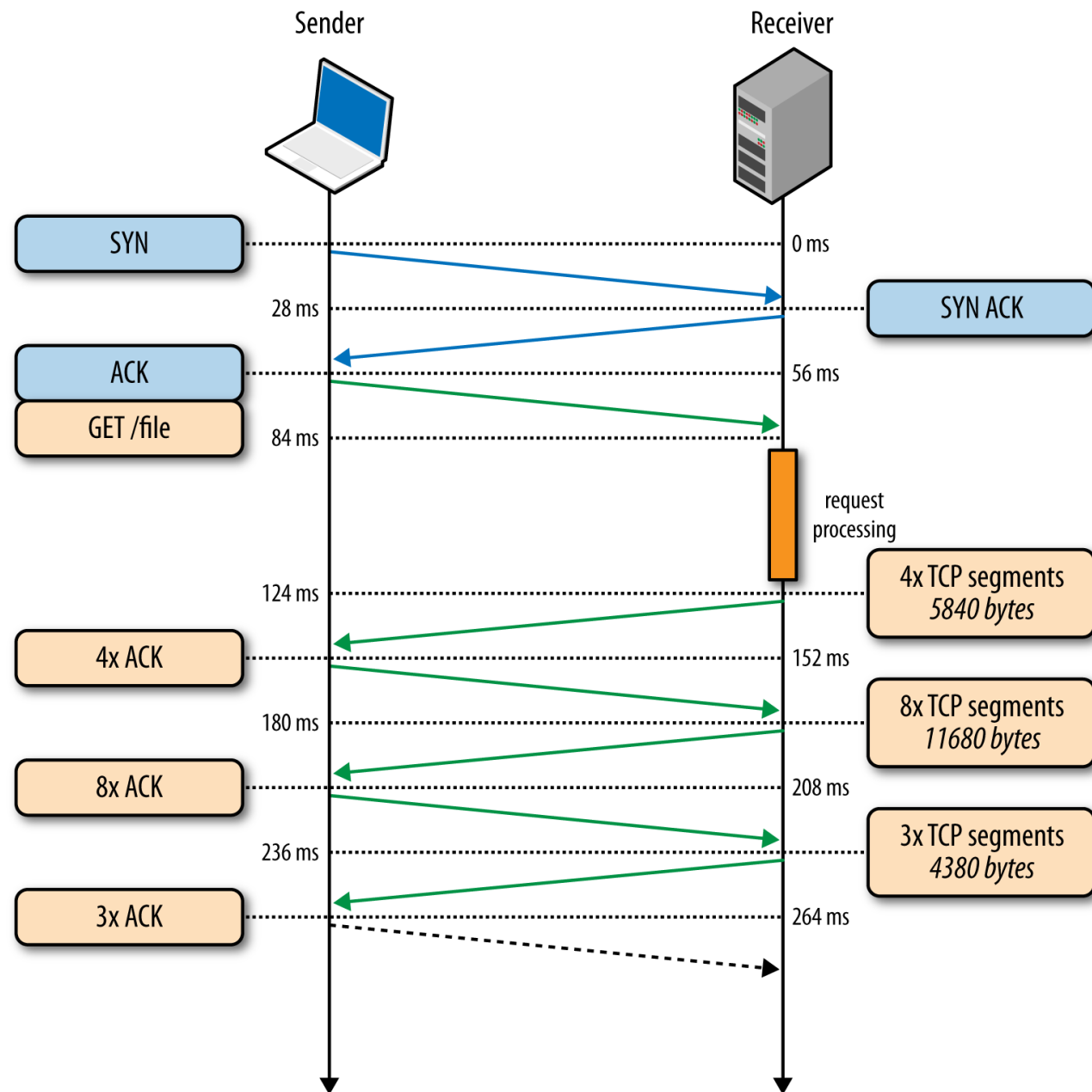
# TCP

**Transmission control protocol** — протокол управления передачей.

TCP протокол базируется на IP для доставки пакетов, но добавляет две важные вещи:

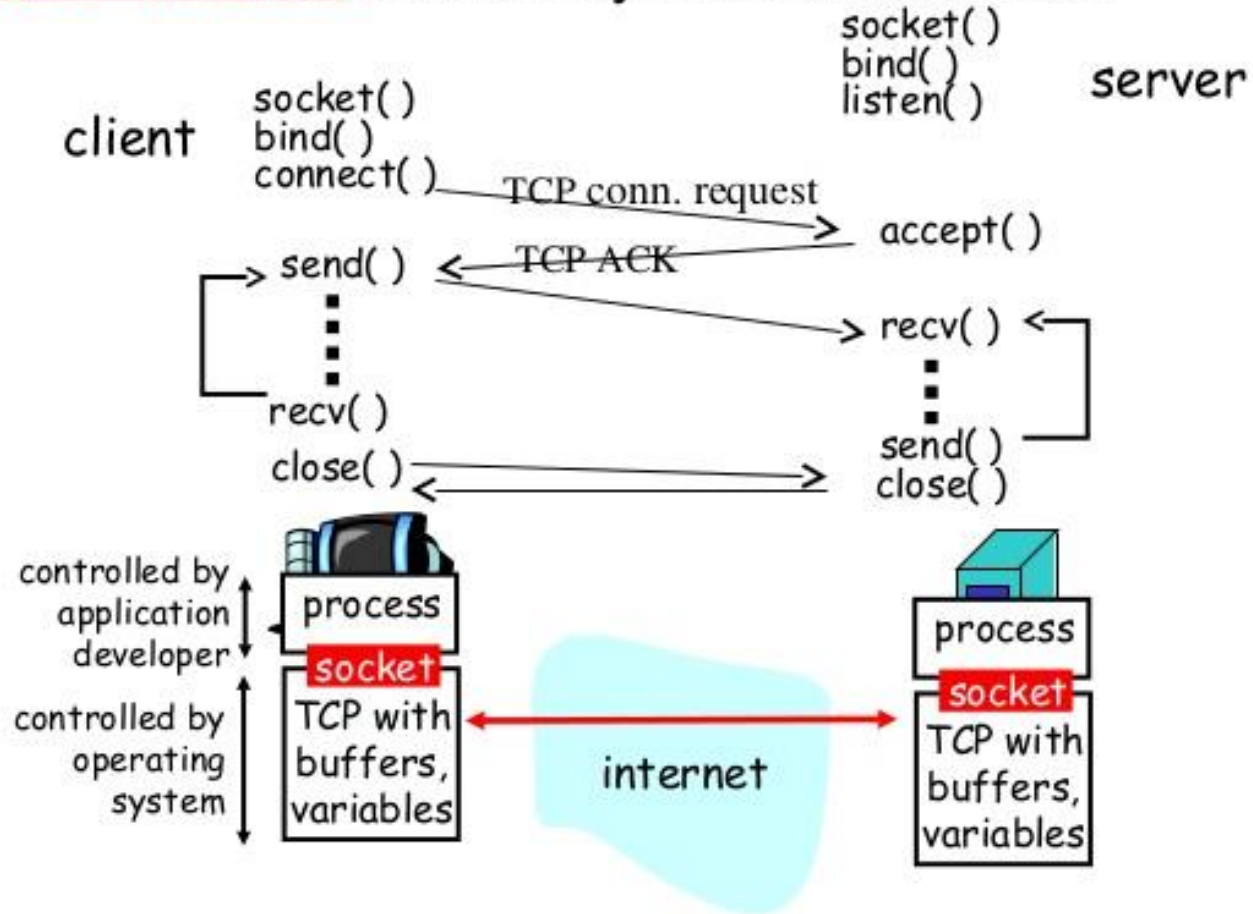
- установление соединения — это позволяет ему, в отличие от IP, гарантировать доставку пакетов
- порты — для обмена пакетами между приложениями, а не просто узлами

Протокол TCP предназначен для обмена данными — это «надежный» протокол.



# Socket-programming using TCP

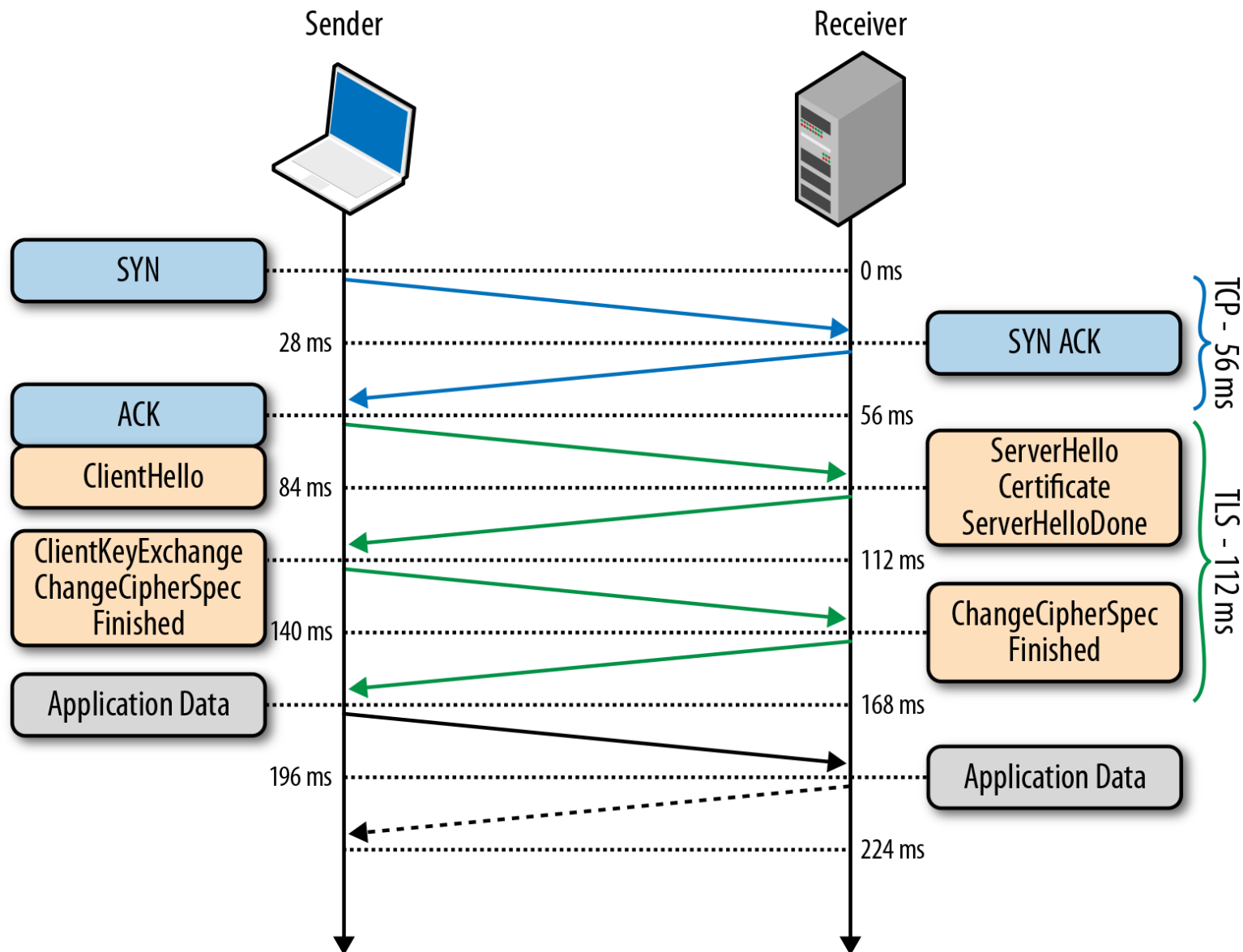
TCP service: reliable byte stream transfer



Sockets


# TLS

- После этого TLS-сессия передаёт данные приложения (HTTP), зашифрованные с помощью подтверждённого симметричного ключа.

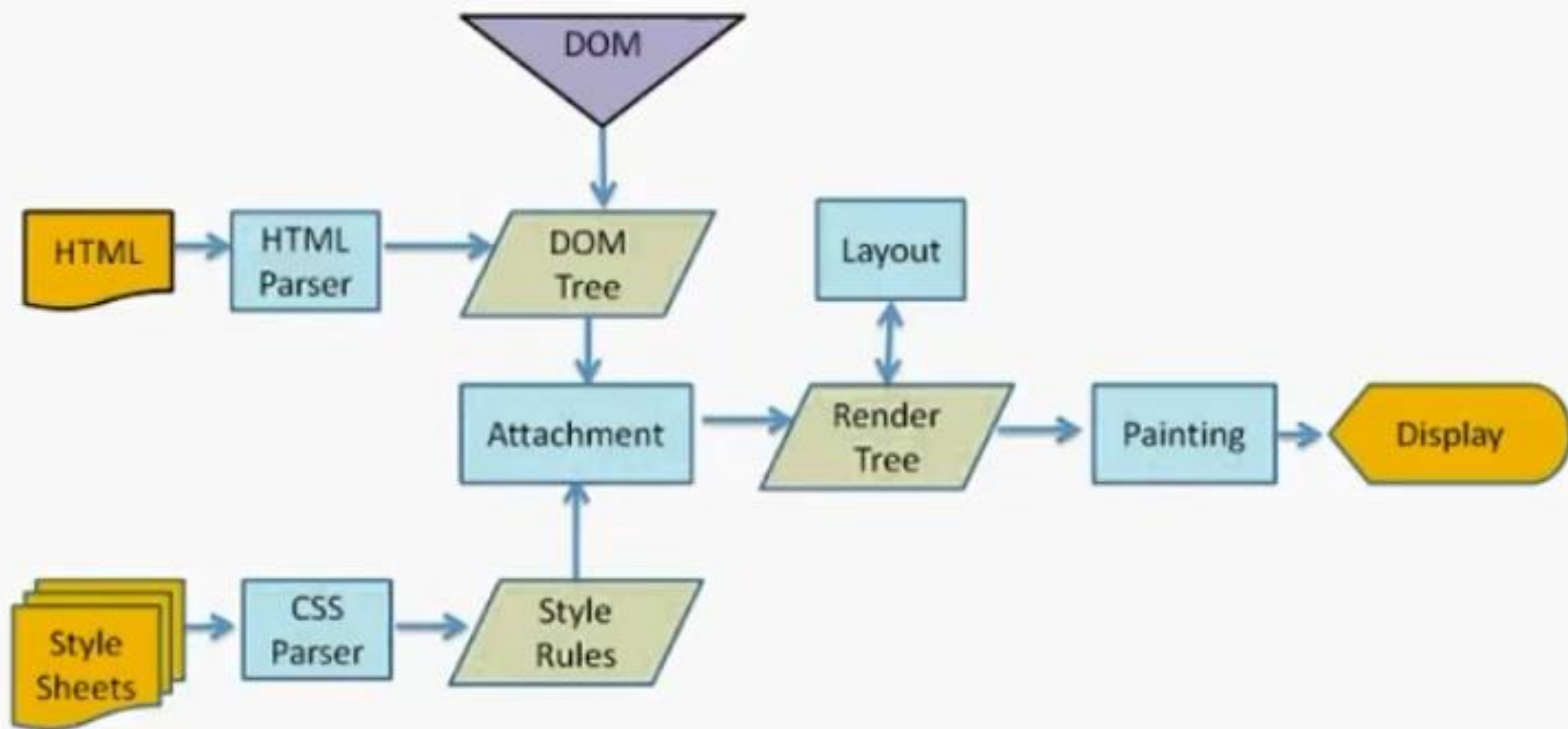




Парсинг  
Рендеринг  
Отображение



# Webkit (Chrome, Safari)



# HTML

## HyperText Markup Language —

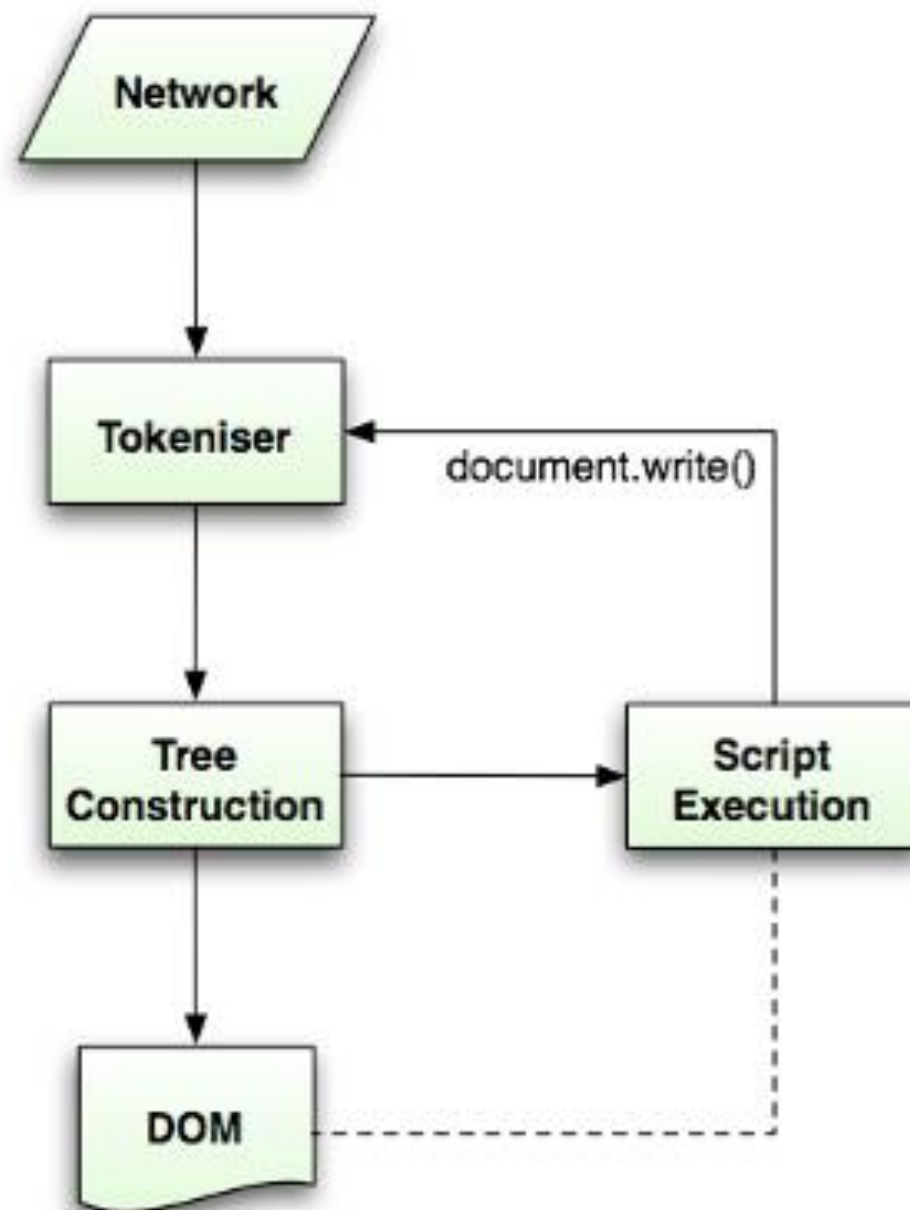
«язык гипертекстовой  
разметки») —  
стандартизированный  
язык разметки  
документов  
во Всемирной  
паутине.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Page Title</title>
5   </head>
6   <body>
7
8     <h1>This is a Heading</h1>
9     <p>This is a paragraph.</p>
10
11   </body>
12 </html>
```

# Парсинг HTML

HTML-нельзя «распарсить» с помощью обычных анализаторов (нисходящих или восходящих). Тому есть несколько причин:

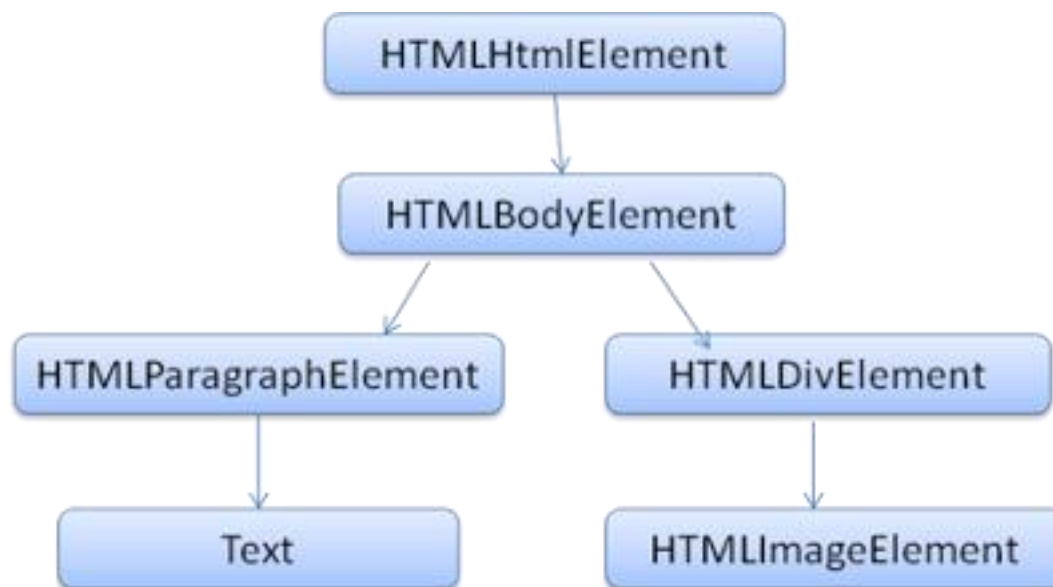
Невозможность использования привычных технологий парсинга приводит к тому, что разработчики браузеров реализуют собственные механизмы разбора HTML.



# DOM

Полученное синтаксическое дерево состоит из элементов DOM и узлов атрибутов. DOM – объектная модель документа (Document Object Model) – служит для представления HTML-документа и интерфейса элементов HTML таким внешним объектам, как код JavaScript.

```
<html>
  <body>
    <p>
      Hello World
    </p>
    <div> </div>
  </body>
</html>
```



# CSS

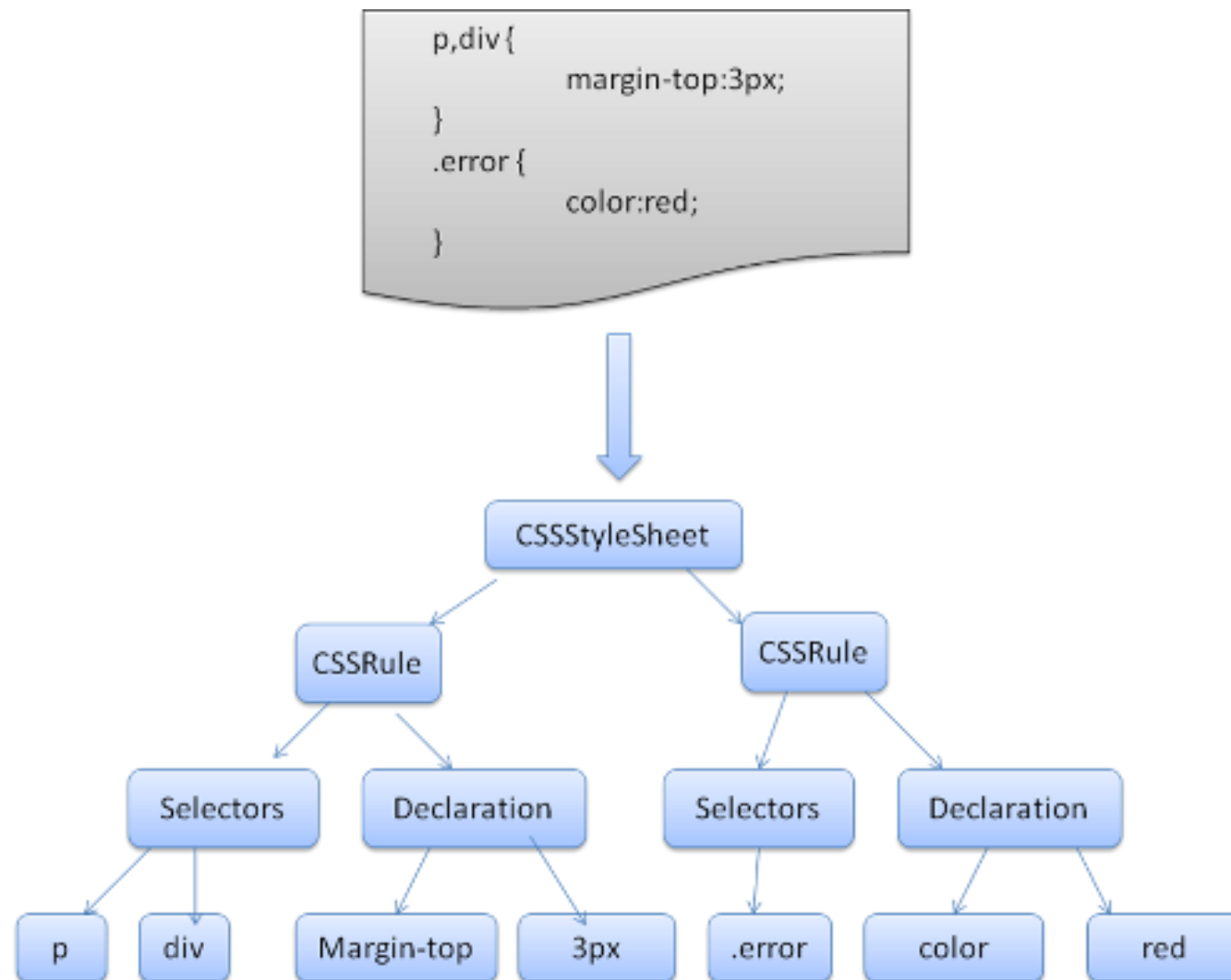
**Cascading Style Sheets** — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки

```
1 body {  
2     background-color: lightblue;  
3 }  
4  
5 h1 {  
6     color: white;  
7     text-align: center;  
8 }  
9  
10 p {  
11     font-family: verdana;  
12     font-size: 20px;  
13 }
```



# Интерпретация CSS

Файл CSS разбирается на объекты StyleSheet, содержащие правила CSS. Объект правил CSS содержит селектор и объявление, а также другие объекты, характерные для грамматики CSS



# Render Tree

## HTML

```
<html>
<head>
  <title>Beautiful page</title>
</head>
<body>

  <p>
    Once upon a time there was
    a looong paragraph...
  </p>

  <div style="display: none">
    Secret message
  </div>

  <div></div>
  ...

</body>
</html>
```

## DOM tree

```
documentElement (html)
  head
    title
  body
    p [text node]
    div [text node]
    div
      img
    ...
```

## Render tree

```
root (RenderView)
  body
    p
      line 1
      line 2
      ...
    div
      img
    ...
```

# Порядок обработки скриптов и таблиц стилей

- **Скрипты**

Веб-документы придерживаются синхронной модели. Предполагается, что скрипты будут анализироваться и исполняться сразу же, как только анализатор обнаружит тег `<script>`. Синтаксический анализ документа откладывается до завершения выполнения скрипта. Если речь идет о внешнем скрипте, сначала необходимо запросить сетевые ресурсы. Это также делается синхронно, а анализ откладывается до получения ресурсов.

- **Ориентировочный синтаксический анализ**

Этот механизм оптимизации используется и в WebKit, и в Firefox. При выполнении скриптов остальные части документа анализируются в другом потоке, чтобы оценить необходимые ресурсы и загрузить их из сети. Таким образом, ресурсы загружаются в параллельных потоках, что повышает общую скорость обработки.

# Вызванное пользователем или пост-рендеринговое исполнение

После завершения рендеринга, браузер исполняет JavaScript-код в результате срабатывания некоего часового механизма (так работают дудлы на странице Google) или в результате действий пользователя (ввод поискового запроса в строку и получение рекомендаций в ответ). Также могут срабатывать плагины вроде Flash или Java (но не в рассматриваемом примере с домашней страницей Google). Скрипты могут потребовать обработки дополнительных сетевых запросов, изменять страницу или её шаблон, что приведёт к следующему этапу рендеринга и отрисовки.