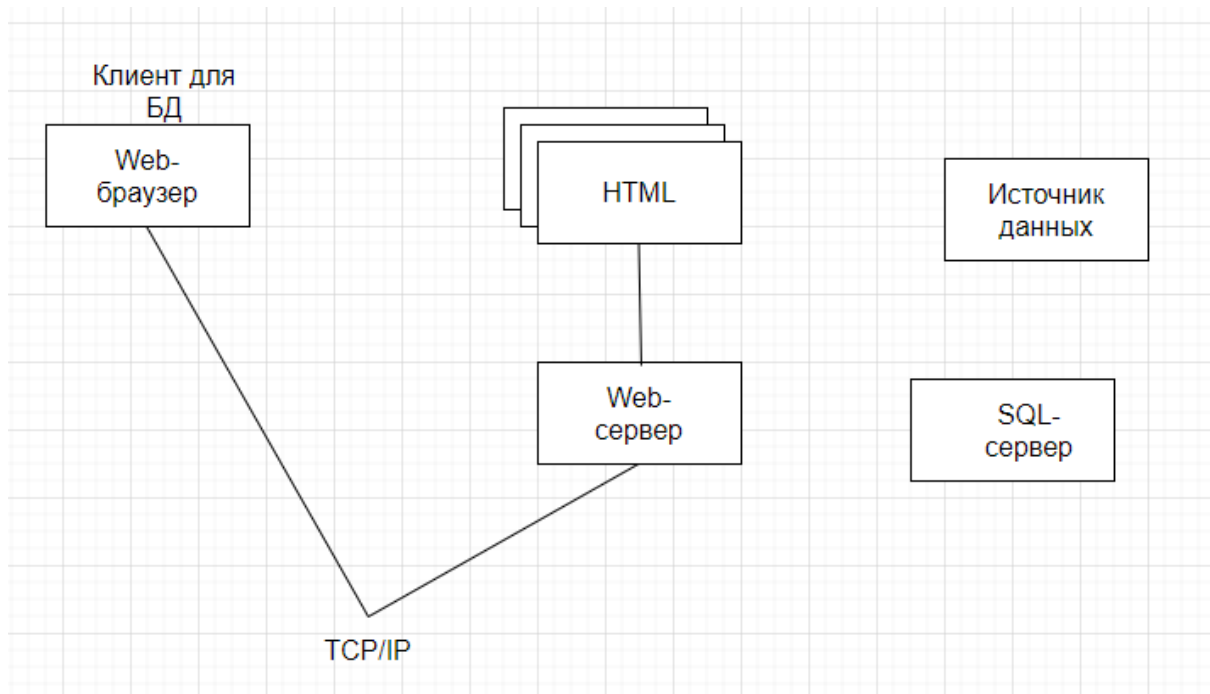


Публикация базы данных в Internet

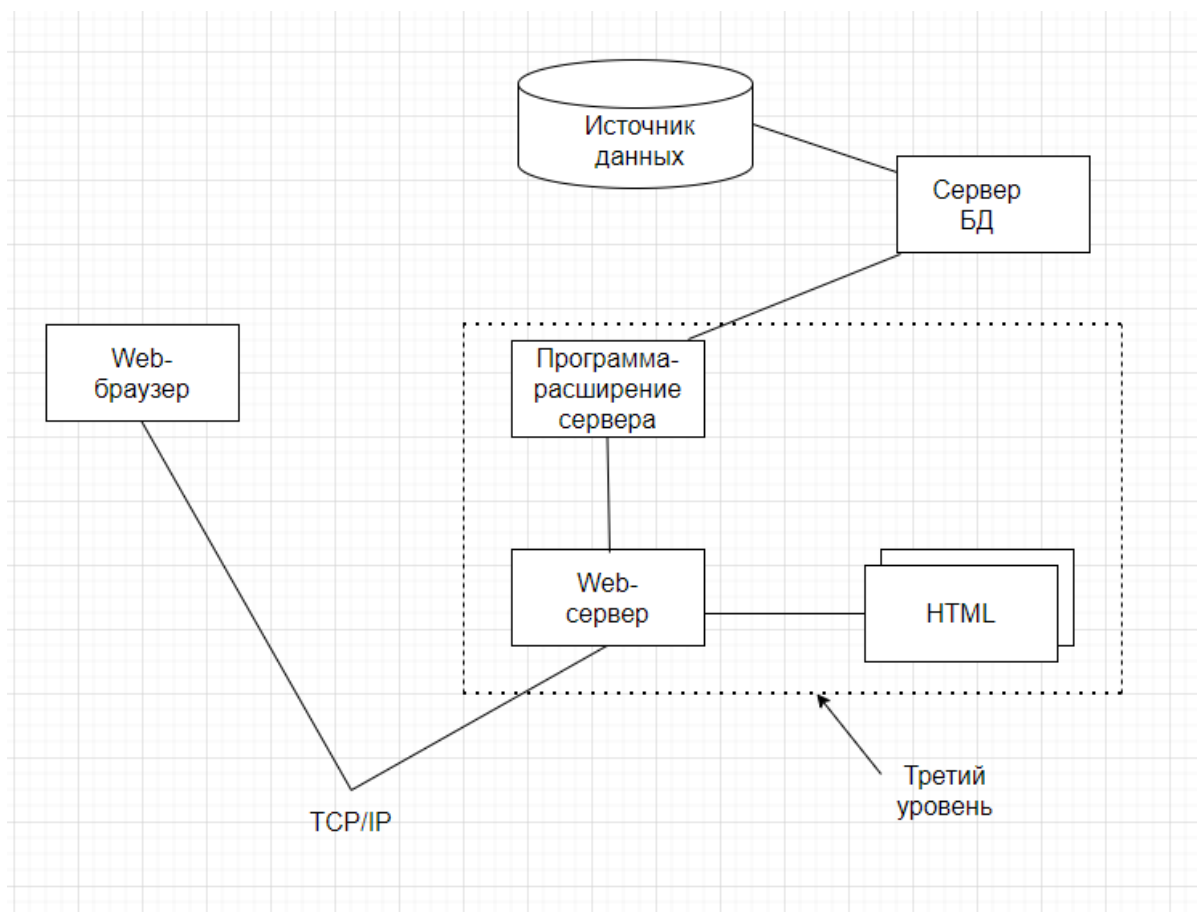
Инструментарий www и баз данных - это две отдельные технологии, разрабатываемые независимо друг от друга. Но и та и другая основаны на двухуровневой архитектуре клиент/сервер.



web-сервер предоставляет html-страницы, а браузер отображает эти страницы, путем интерпретации тегов HTML. Аналогично SQL сервер позволяет клиенту отображать и модифицировать данные в базе.

Третий уровень

Приложения, работающие с базами данных для www, объединяют свои двухуровневые технологии в новый тип систем, основанный на трехуровневой архитектуре клиент/сервер.



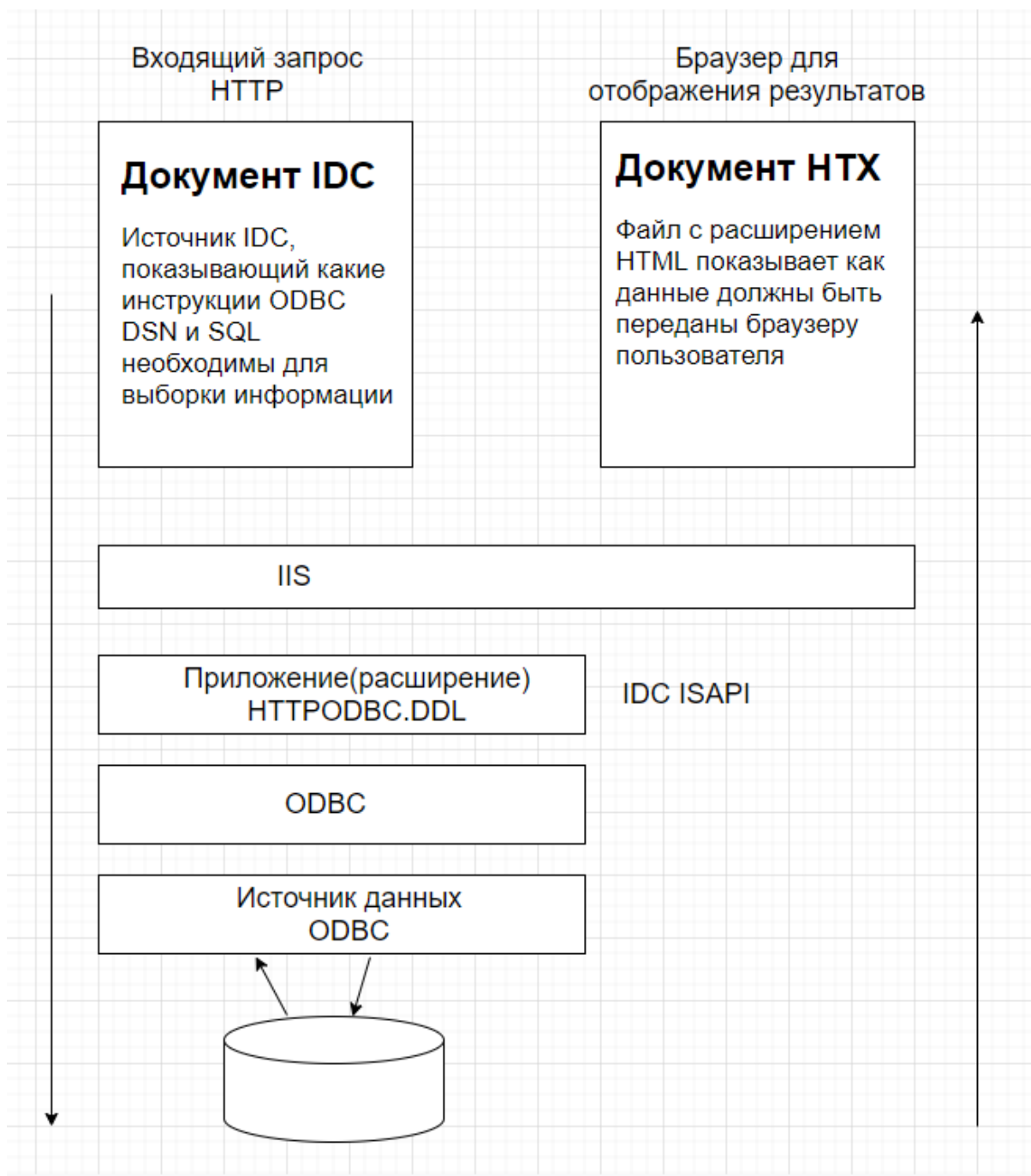
Клиентский уровень занимает браузер, на уровне сервера находится сервер базы данных, а на промежуточном уровне располагается web-сервер и программа расширения сервера.

Браузер посылает Web-серверу запросы на доставку web-страниц или данных. Web-сервер обслуживает заявки на web-страницы, а запросы данных отправляет программе-расширению серверной части. Последняя принимает переданные ей запросы и преобразует их в форму, понятную серверу базы данных (например, в запросы ODBC SQL) и передает их серверу базы данных. Затем сервер базы данных выполняет работу по обслуживанию запроса, например добавление записи, и возвращает результат программе-расширению серверной части. Наконец та преобразует результат в формат, приемлемый для браузера (HTML), и передает их Web-серверу, а тот в свою очередь - браузеру.

Существуют различные виды программ-расширений серверной части. Это программы с обычными CGI, с гибридными CGI и с API (ISAPI, NSAPI).

Пример: одно из решений Microsoft для IIS на базе источников данных ODBC.

Web-сервер IIS предоставляет доступ к уровню ODBC с использованием средства доступа к базам данных по Internet IDC (Internet Database Connection). IDC работает как промежуточное звено системы, обеспечивая взаимодействие между тем, что обрабатывается при просмотре данных HTML и тем, как информация запрашивается на уровне базы данных.



Доступ ODBC с приложениями HTML и расширение HTML с IDC

IDC предоставляет доступ к любому источнику данных ODBC

Когда пользователь в браузере в URL указывает файл IDC, они просят систему IIS использовать этот файл IDC и его инструкции для запроса к базе данных и вернуть результат. В URL указывается файл IDC, но на самом деле возвращается файл HTX или файл с расширением HTML. Файл с расширением HTX - это шаблон, показывающий как должны отображаться результирующие данные. Из нашего рисунка видно, что инструментом, выполняющем работу с базой данных с помощью ODBC, является HTTPODBC.DDL. Этот файл создается при установке IIS. Он является

приложением, разработанным с помощью Internet Server API(ISAPI) и работает в виде программного обеспечения сервера.

Подобное решение специально предназначено для работы с базой данных и имеет возможность применения двух исходных файлов IDC и HTX, которые требуются для передачи информации обратно пользователю.

Пример файла IDC sample.idc

Data source: web sql

Username: sa

Template: sample.htx

SQLStatement: +SELECT au_lname, ytd_sales FROM title WHERE ytd_sales > 5000

Применение HTML при размещении без данных в www

Мы познакомимся с теми элементами языка HTML, которые необходимо знать для разработки Web-страниц, соединенных с базами данных.

HTML состоит из тегов, которыми обрамляется любой элемент документа(то есть фрагмент текста или изображение). Теги идентифицируют охваченные ими объекты как принадлежащие одной из статических страниц HTML.

Каждый HTML-документ начинается и заканчивается парой тегом <html> </html>, а так же разделяется на заголовок и тело тегами <head> </head> и <body> </body>.

Обличение тегом <title> браузер обычно отображает в своей области заголовка.

Пример:

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>

Пример веб-страницы

</title>

</head>

<body>

</body>

</html>

Статический HTML и динамический HTML

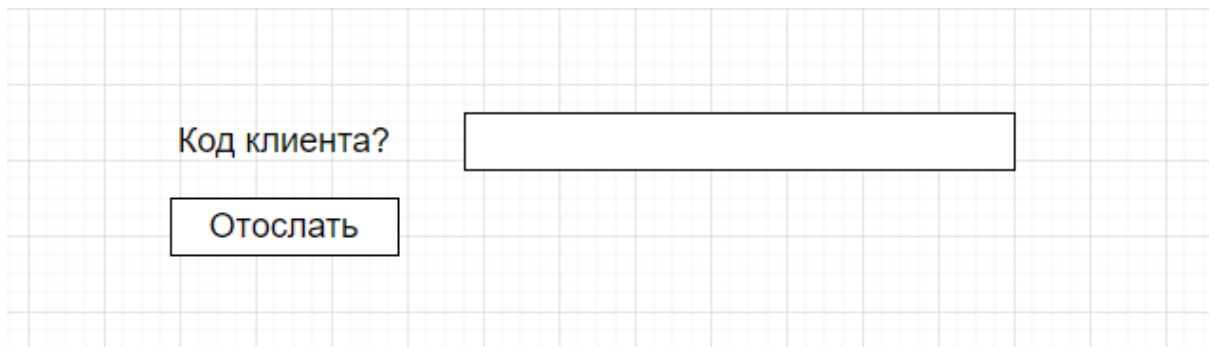
Статическими называются HTML документы, созданные и помещенные на Web-сервер. Когда браузер заказывает статическую HTML страницу, Web сервер всегда присылает ее в той форме, в какой она была создана.

Динамические HTML документы создаются на ходу в ответ на запрос от браузера. Сам Web-сервер не принимает участия в процессе динамической генерации, а просто передает запрос на доставку страницы программе-расширению сервера и отдает возвращаемый результат браузеру.

Создание форм ввода

Поля ввода задаются формами HTML <form> </form>

Пример формы:



Код клиента?

Отослать

```
<form method = POST action = /cgi-bin/query.exe>
```

```
<b>
```

```
Введите код клиента
```

```
</b>
```

```
<input type="text" name="WP" value="" size=5/>
```

```
<p>
```

```
<input type="submit" value="Отправить"/>
```

```
</p>
```

```
</form>
```

Использование JavaScript для контроля ввода

JavaScript - это язык сценариев, имеющий встроенные объекты и функции. JavaScript в отличие от языка Java обычно интегрируется с HTML.

Как функции JavaScript интегрируются с HTML-формами.

Один способ вызвать функцию JavaScript состоит в том, чтобы поместить вызов между формами `<script> </script>` в части HTML документа.

Например:

```
<body>
```

```
    <script>
```

```
        alert("Hello");
```

```
    </script>
```

```
</body>
```

Другой способ вызвать функцию JavaScript состоит в том, чтобы определить ее как обработчик событий(event handler). Обработчиком событий называют функцию, вызываемую, когда происходит событие. Браузер передает событие обработчику, который и совершает требуемое действие. Задав определенную функцию для определенного события, браузер вызывает эту функцию как только произошло это событие.

Пример событий:

onSubmit - пользователь нажимает кнопку

onChange - текст, область текста или поле выбора были изменены.

Пример:

```
<input type="text" name="test" value="" onChange = "isValidInt('900', 101, 808)">
```

```
<form name = MainForm method = POST action =input.js  
onSubmit="returnCheckAll">
```

Как включить текст функций в документ HTML:

```
<script language = "JavaScript" >
```

```
    function HelloDB{
```

```
        alert("Hello user");
```

```
    }
```

```
</script>
```

Пример:

Таблица

Pass Client

Client Code	Client Pass
-------------	-------------

Таблица

Clients

Client Code	Client Name	Address	Telephone
-------------	-------------	---------	-----------

Клиент вызывает браузер и вводит код клиента и пароль. Осуществляется контроль ввода данных. Если данные клиента правильные, то он получает информацию на HTML странице из базы данных