

Технология клиент-сервер

Роли

Сервер – компьютер, управляющий тем или иным ресурсом.

Клиент – компьютер (или программа), запрашивающий и пользующийся каким-либо ресурсом

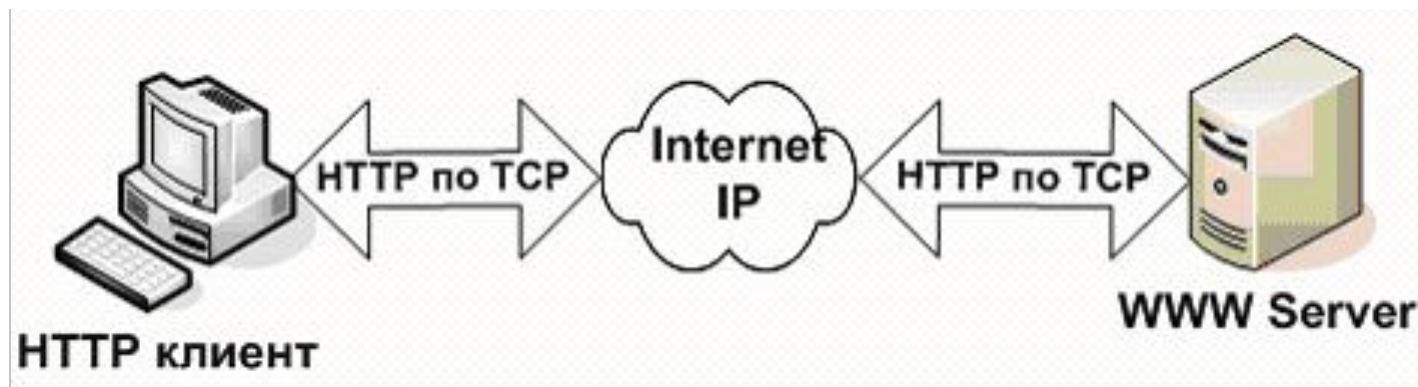
Для одного ресурса можно выполнять роль клиента, для другого – сервера.

Служба WWW

Основная служба в сети Интернет, позволяющая получать доступ к информации на любых серверах, подключенных к сети.

WWW построена по схеме "клиент-сервер".

В качестве клиента выступает браузер, который является также и интерпретатором HTML.



Понятие «Веб-сервера»

Web-сервер

- программа, запущенная на узле сети Интернет и выдающая посетителям этого узла web-страницы по запросам. Также web-сервером часто называется узел, на котором эта программа запущена, или даже компьютер, являющийся таким узлом.

Дополнительные функции

- Автоматизация работы веб страниц;
- ведение журнала обращений пользователей к ресурсам;
- аутентификация и авторизация пользователей;
- поддержка динамически генерируемых страниц;
- поддержка HTTPS для защищённых соединений с клиентами.

Клиенты

В качестве клиентов для обращения к веб-серверам могут использоваться различные программы и устройства:

- веб-браузер, работающий на настольном компьютере или переносном устройстве (например, карманном ПК);
- разнообразные программы, самостоятельно обращающиеся к веб-серверам для получения обновлений или другой информации (например, антивирус может периодически запрашивать у определённого веб-сервера обновления своих баз данных);
- мобильный телефон, получающий доступ к ресурсам веб-сервера при помощи протокола WAP;
- другие цифровые устройства или бытовая техника.

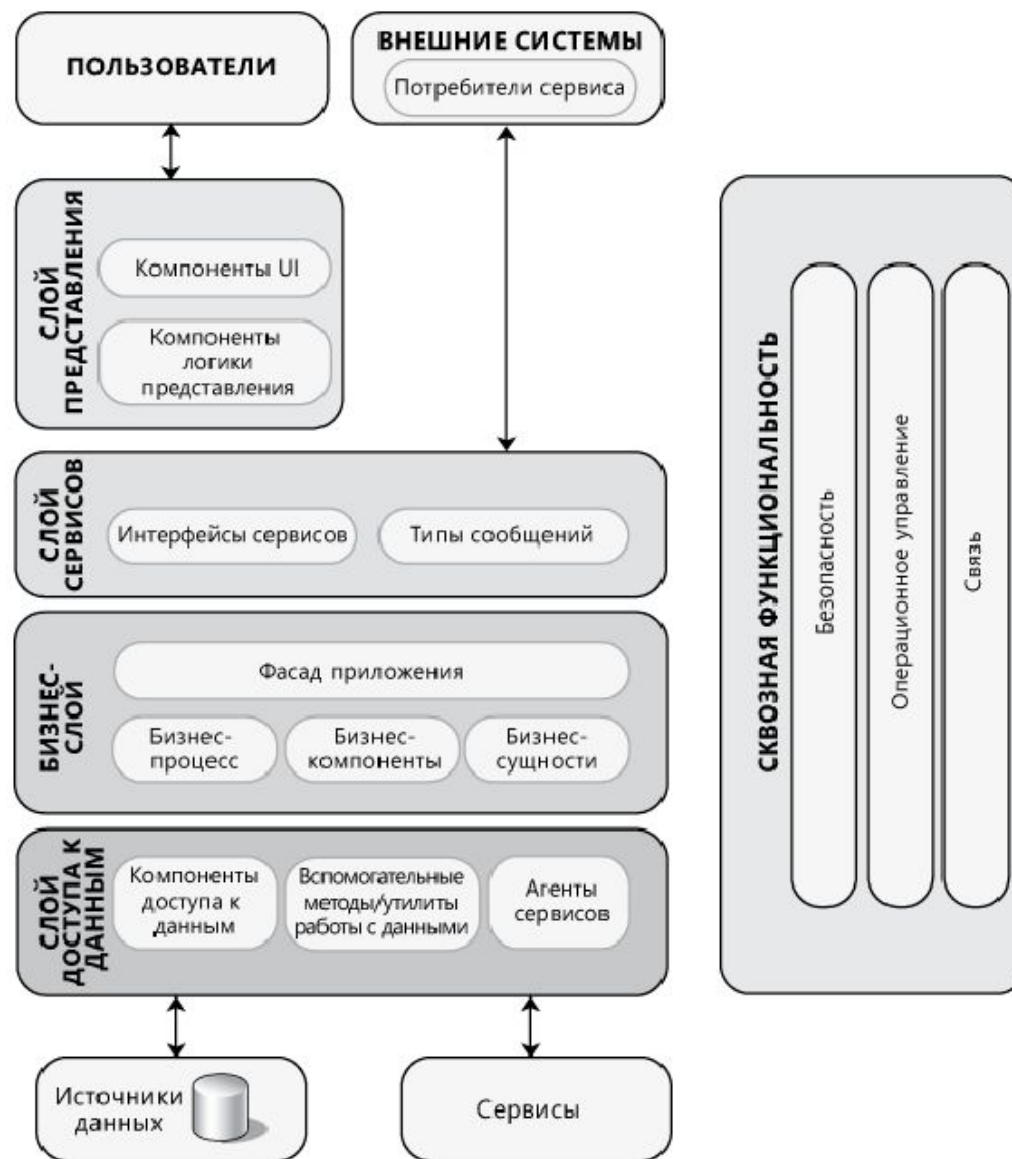


Рис. 2

Типовая архитектура приложения

Принципы работы в двухзвенной архитектуре

Выделяет в каждой программе следующие функциональные составляющие:

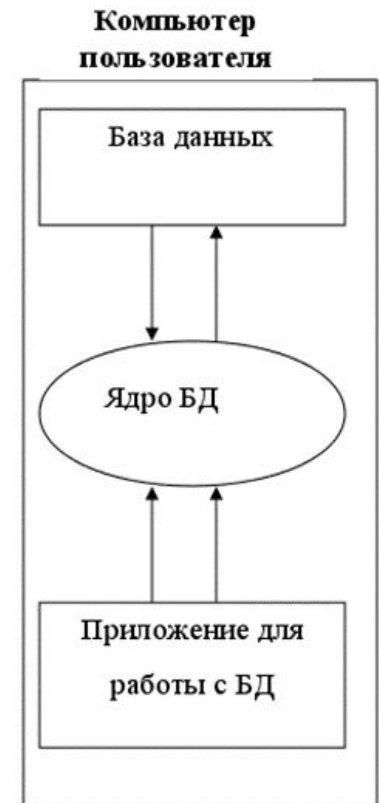
- ⇒ **средства и логика пользовательского интерфейса** — эта часть отвечает за организацию представления данных на экране, она включает в себя средства и ресурсы пользовательского интерфейса, а также правила и возможные сценарии взаимодействия с пользователем; прикладная логика — это часть выполняет основную обработку данных программой, содержит набор правил для принятия решений, а также все необходимые вычислительные процедуры и операции
- ⇒ **логика работы с данными** - эта часть обеспечивает выполнение простейших операций над данными: выбор данных, поиск, сортировка, фильтрация, а также заполнение, модификация и удаление некоторых структур данных;
- ⇒ **файловые операции** — эта часть обеспечивает выполнение стандартных операций над файлами и файловой системой, которые, как правило, реализуются с помощью функций ОС

Архитектуры

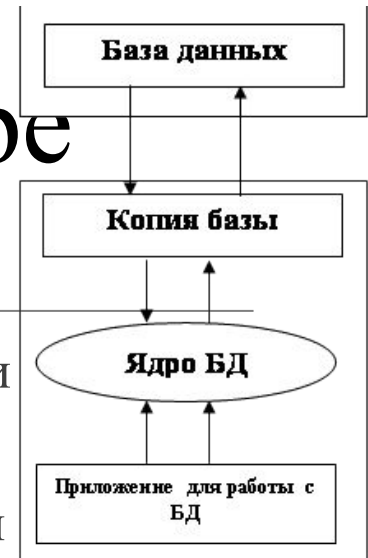
- локальные базы данных и архитектура "файл-сервер";
 - архитектура "клиент-сервер";
- многозвенная (трехзвенная N-tier или multi-tier) архитектура

Локальная архитектура

- При работе с локальными базами данных сами БД расположены на том же компьютере, что приложения, осуществляющие доступ к ним.
- Работа с БД происходит в однопользовательском режиме.
- Ядро БД расположено на компьютере пользователя
- Приложение ответственно за поддержание целостности БД и за выполнение запросов к БД.



Структура приложения, построенного в архитектуре «файл-сервер»



- При работе в архитектуре "файл-сервер" БД и приложение расположены на файловом сервере сети.
- Возможна многопользовательская работа с одной и той же БД, когда каждый пользователь со своего компьютера запускает приложение, расположенное на сетевом сервере.
- Тогда на компьютере пользователя запускается копия приложения. По каждому запросу к БД из приложения, данные из таблиц БД перегоняются на компьютер пользователя, независимо от того, сколько реально нужно данных для выполнения запроса. После этого выполняется запрос.

Архитектура клиент-сервер

- Архитектура "клиент-сервер" разделяет функции приложения пользователя (называемого клиентом) и сервера.
- Взаимодействие сервера БД и приложения клиента происходит сл. Образом: клиент формирует запрос и отправляет серверу, сервер принимает запрос, выполняет его и результат возвращает клиенту. В клиентском приложении — только интерпретация полученных от сервера данных, реализация интерфейса с пользователем и ввод данных.



Классическая двухуровневая модель Клиент-Сервер с источником данных

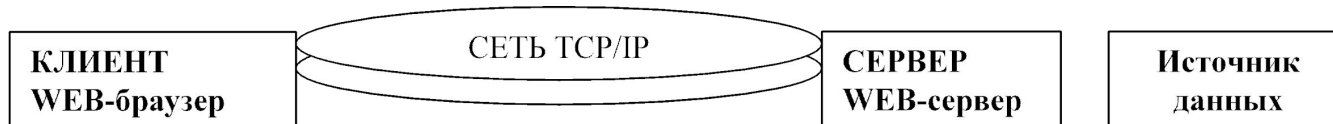
Простейшие клиент-серверные разработки могут реализоваться следующим образом: клиент – на базе Visual Basic, связь с серверной частью через DAO-ODBC (активные объекты и контейнер баз данных) с базой данных сервера – на базе SQL-сервера.

Преимущества:

- уменьшение сетевого трафика
- единые для всех пользователей правила по обеспечению достоверности и секретности данных

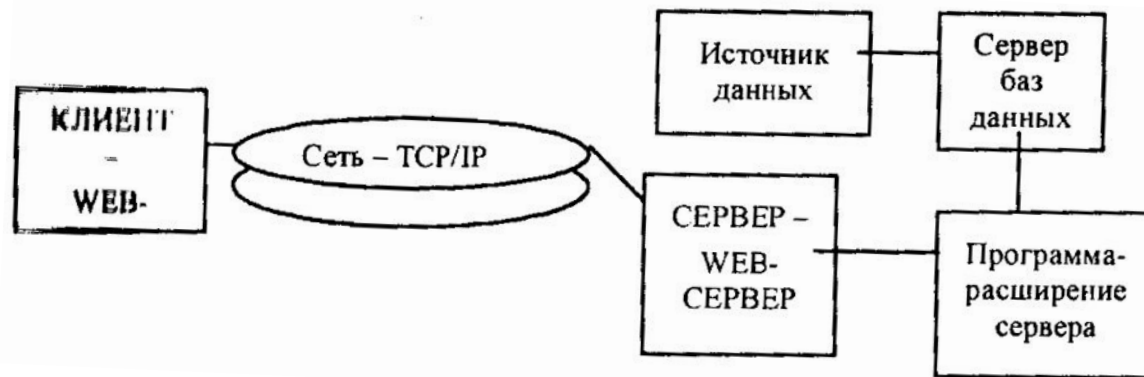
Недостатки такой модели:

- перегрузка клиентской части – длительное время ожидание ответа
- проблемы синхронизации приложений
- затруднительны процессы размножения и модификации разработки.
- Ограниченное число пользователей



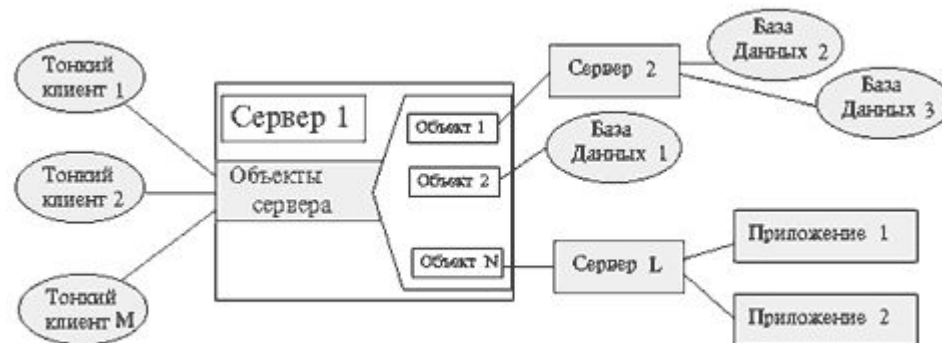
Трёхуровневая модель

Трёхуровневая архитектура, или трёхзвённая архитектура (англ. three-tier или англ. Multitier architecture) — архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиентского приложения (обычно называемого «тонким клиентом» или терминалом), сервера приложений, к которому подключено клиентское приложение, и сервера базы данных, с которым работает сервер приложений.



Распределенная (многоуровневая) модель

Если между клиентом и сервером существует несколько процессов, то говорят о многоуровневой архитектуре. Здесь выделяются WEB-сервер, программа-расширение, сервер баз данных и клиент в виде браузера.



Типы клиентов в системе клиент-сервер

Толстый клиент, rich client архитектуре клиент-сервер — это приложение, обеспечивающее (в противовес тонкому клиенту) полную функциональность и независимость от центрального сервера.

Часто сервер в этом случае является лишь хранилищем данных, а вся работа по обработке и представлению этих данных переносится на машину клиента

Тонкий клиент

Тонкий клиент, thin client в компьютерных технологиях — компьютер или программа-клиент в сетях с клиент-серверной или терминальной архитектурой, где большая часть задач по обработке информации перенесена на сервер и права доступа клиента строго ограничены. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями.

Вопросы

1. Какие функциональные составляющие включает в себя общая модель разбиения каждой программы?
2. Что такое файл-сервер? Какие функции он выполняет?
3. Каким образом поддерживается удаленный доступ к файлам?
4. Какие файловые операции выполняет файл-сервер?
5. Нарисуйте схему доступа к данным для двух приложений типа «файл сервер»
6. Назовите недостатки архитектуры «файл—сервер»
7. Область применения архитектуры «файл-сервер»
8. Архитектура клиент-сервер. Разновидности архитектуры.