

# АРХИТЕКТУРЫ УДАЛЕННЫХ БАЗ ДАННЫХ

# Режимы работы с базой данных

Однопользовательский

Последовательный

С централизованной БД

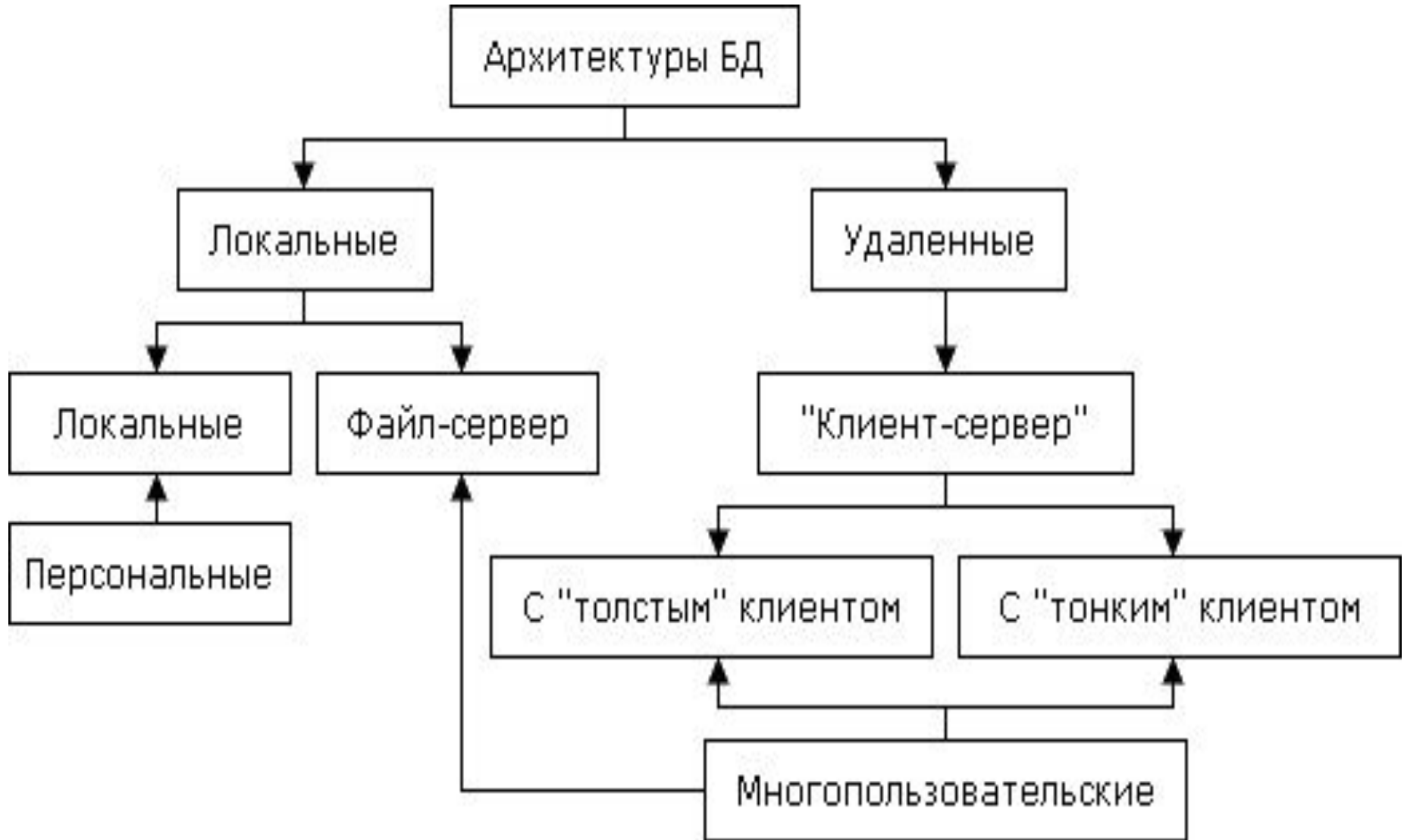


Многопользовательский

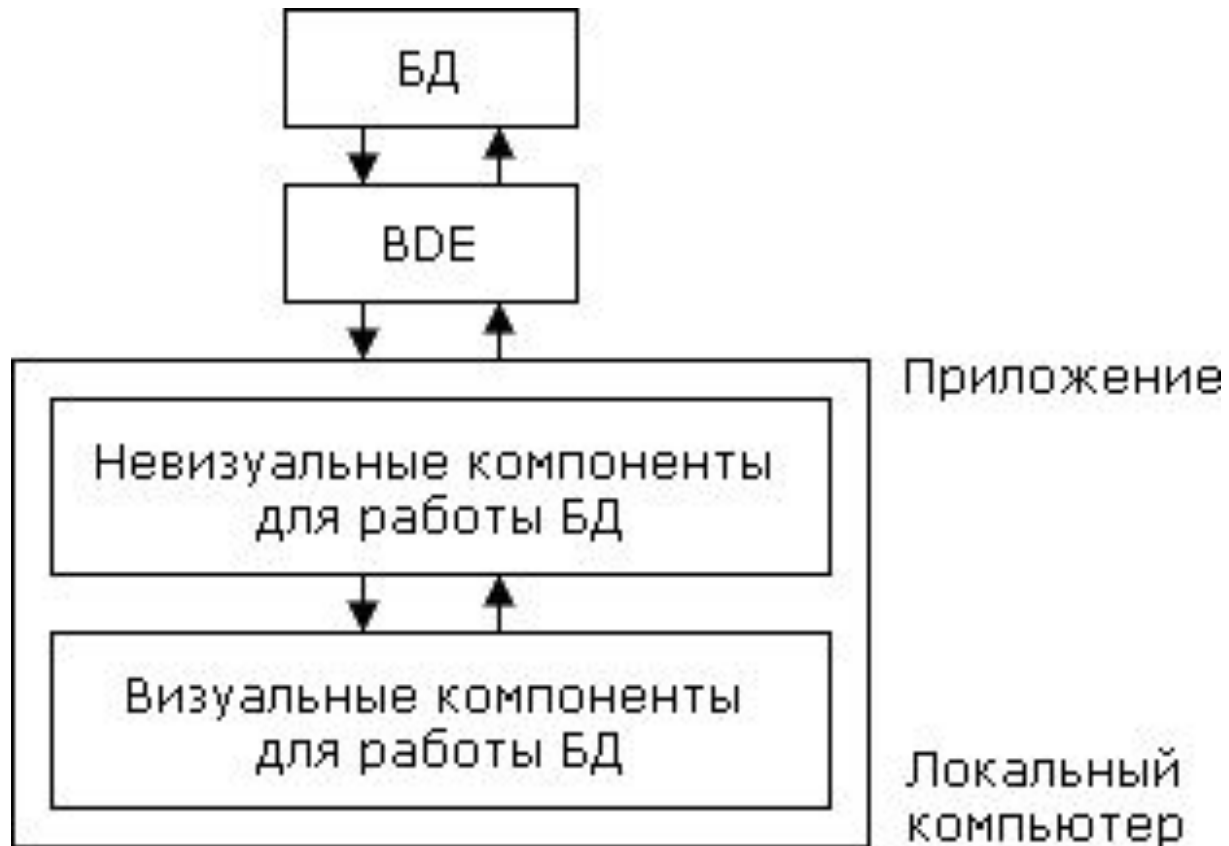
Параллельный

С распределенной БД

# Структурная схема терминов



# Локальная БД



КЛИЕНТ



СУБД



СЕРВЕР БАЗ ДАННЫХ



БАЗА ДАННЫХ

# Архитектура многопользовательских СУБД

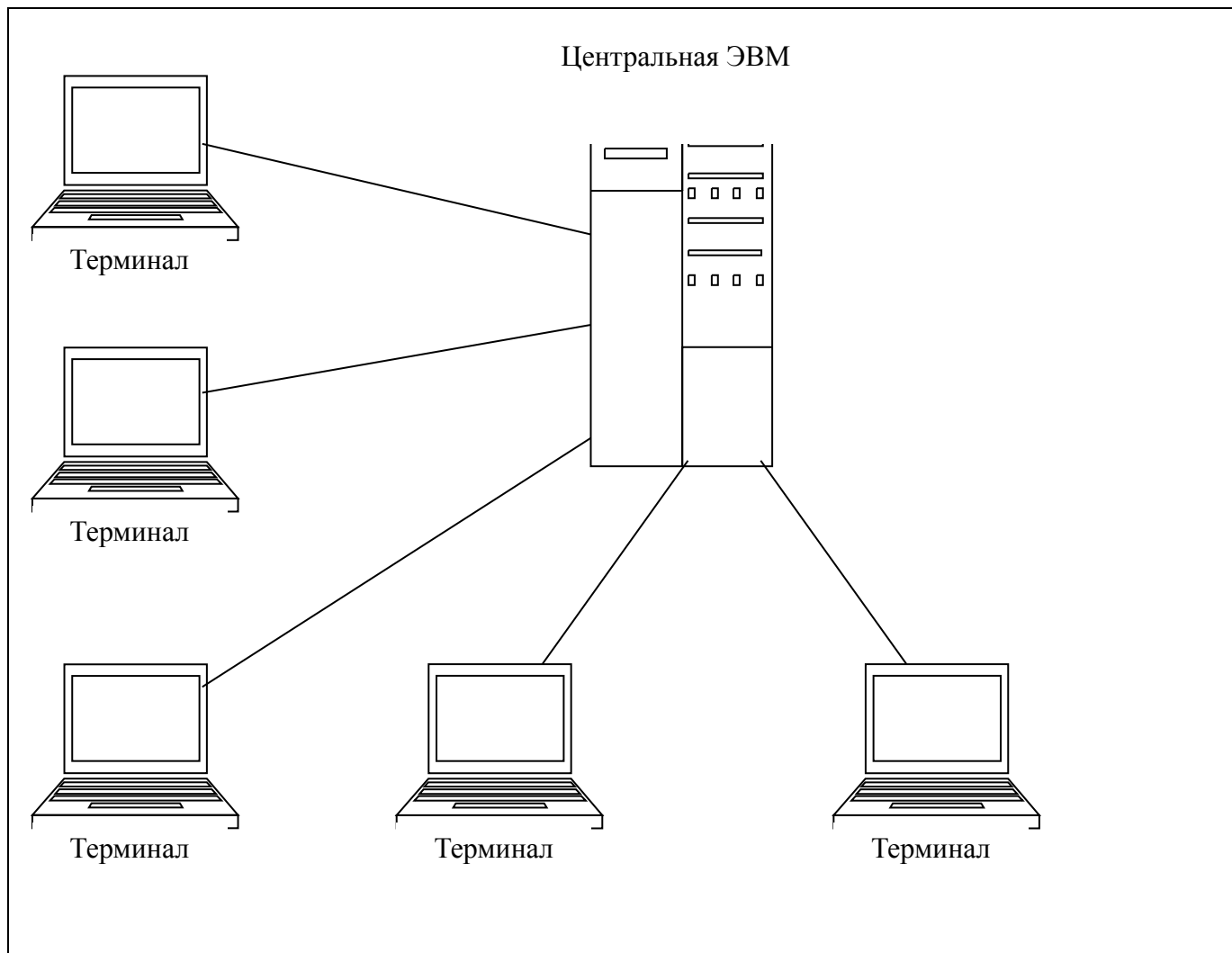
В настоящее время существуют такие *основные типовые архитектурные решения*, используемыми при реализации *многопользовательских СУБД*, а именно:

- Телеобработка
- Файловый сервер
- Технология "клиент-сервер".

# Телеобработка

**Телеобработка** – схема, при которой один компьютер с единственным процессором соединен с несколькими терминалами. При этом вся обработка выполняется с помощью этого компьютера.

# Топология архитектуры телеобработки





# Введение в технологию клиент-сервер

База данных представляет собой набор файлов на сетевом сервере. Характерная особенность архитектуры клиент-сервер является перенос вычислительной нагрузки на сервер БД(SQL сервер), а также максимальная разгрузка клиента от вычислительной работы и существенное укрепление данных.

Взаимодействие сервера и клиента реализуется с помощью SQL запросов, которые формирует и отправляет серверу клиент. Сервер, приняв запрос, выполняет его и возвращает результат клиенту. В клиентском приложении в основном осуществляется интерпретация полученных от сервера данных, реализация пользовательского интерфейса, а также реализация части бизнес - правил.

# Архитектура клиент-сервер

***"Клиент/сервер" означает такой способ взаимодействия программных компонентов, при котором они образуют единую систему.***

***Существует некий клиентский процесс, требующий определенных ресурсов, а также серверный процесс, который эти ресурсы предоставляет.***

# Архитектура клиент-сервер

**Клиент** принимает от пользователя запрос, проверяет синтаксис и генерирует запрос к базе данных на языке SQL или другом языке базы данных, который соответствует логике приложения. Затем он передает сообщение серверу, ожидает поступления ответа и форматирует полученные данные для представления их пользователю.

**Сервер** принимает и обрабатывает запросы к базе данных, а затем передает полученные результаты обратно клиенту. Такая обработка включает проверку полномочий клиента, обеспечение требований целостности, поддержку системного каталога, а также выполнение запроса и обновление данных, при этом поддерживается управление параллельностью и восстановлением.

В основе архитектуры клиент-сервер лежит идея разделения ресурсов, что приводит к функциональному выделению компонентов сети:

- ***Рабочая станция (клиент)*** предназначена для непосредственной работы пользователей, запрашивающий услуги у некоторого сервера.
- **Клиент в** типичной конфигурации клиент/сервер - это автоматизированное рабочее место, использующее графический интерфейс (Graphical User Interface - GUI).

- **Сервер** предназначен для хранения, передачи и распределения информации между клиентами. Сервер локальной сети предоставляет ресурсы (услуги) рабочим станциям и/или другим серверам.
- Чтобы прикладная программа, выполняющаяся на рабочей станции, могла запросить услугу у некоторого сервера, требуется некоторый интерфейсный программный слой, поддерживающий такого рода взаимодействие. Из этого и вытекают основные принципы системной архитектуры "клиент-сервер".

# *Преимущества архитектуры клиент – сервер:*

- Большинство вычислительных процессов происходит на сервере, что снижает требования к вычислительной мощности клиента.
- Снижается сетевой трафик, так как пересылаются не все данные, а только запрошенные.
- БД на сервере представляет единый файл, в котором содержатся таблицы, ограничения целостности и другие компоненты БД.
- Взломать, похитить или испортить такую БД значительно труднее; существенно увеличивается защищенность БД от ввода неправильных значений.
- Кроме того для каждого пользователя устанавливаются свои уровни доступа.
- Сервер реализует управление транзакциями и предотвращает попытки одновременного изменения одних и тех же данных.

- Технология “клиент-сервер” применительно к СУБД сводится к разделению системы на две части – приложение-клиент (front-end) и сервер базы данных (back-end).
- Графический интерфейс пользователя стал стандартом для систем “клиент-сервер”.
- Кроме того, архитектура “клиент-сервер” значительно упрощает и ускоряет разработку приложений за счет того, что правила проверки целостности данных, находятся на сервере. Неправильно работающее клиентское приложение не может привести к потере или искажению данных.



# Логическая модель РБД -

строится на 3-х уровнях (компонентах) абстракции данных:

- представления информации;
- обработки (бизнес-логики);
- хранения.

Компоненты образуют строгую иерархию: слой бизнес -- логики взаимодействует со слоями хранения и представления.

Физически, слои могут входить в состав одного программного модуля, или же распределяться на нескольких параллельных процессах в одном или нескольких узлах сети.

Каждый слой обрабатывается соответственно функциями стандартного интерактивного приложения

На верхнем уровне абстрагирования взаимодействия клиента и сервера достаточно четко можно выделить следующие компоненты:

- функции ввода и отображения данных или презентационная логика (Presentation Layer- PL);  
-- Обеспечивает интерфейс с пользователем. Как правило, получение информации от пользователя происходит посредством различных форм. А выдача результатов запросов - посредством отчетов.

- прикладные функции, характерные для данной предметной области или бизнес-логика (Business Layer- BL); именно он определяет функциональность и работоспособность системы в целом. Блоки программного кода распределены по сети и могут использоваться многократно для создания сложных распределенных приложений.

- фундаментальные функции хранения и управления информационными ресурсами (базами данных, файловыми системами и т. д.) или логика доступа к ресурсам (Access Layer – AL) -- Обеспечивает физическое хранение, добавление, модификацию и выборку данных. На данный компонент также возлагается проверка целостности и непротиворечивости данных, а также реализацию разделенных транзакций.
- Служебные функции, играющие роль связок между функциями первых трех групп.

В соответствии с этим в любом приложении выделяются следующие логические компоненты:

- прикладной компонент, поддерживающий функции второй группы;
- компонент доступа к информационным ресурсам, поддерживающий функции третьей группы;
- компонент представления, реализующий функции первой группы;
- протокол взаимодействия.

Различия в реализациях технологии определяются четырьмя факторами.

- в какие виды программного обеспечения интегрирован каждый из этих компонентов.
- тем, какие механизмы программного обеспечения используются для реализации функций всех четырех групп.
- как логические компоненты распределяются между компьютерами в сети.
- какие механизмы используются для связи компонентов между собой.

# Архитектура клиент-сервер

## Операции сервера

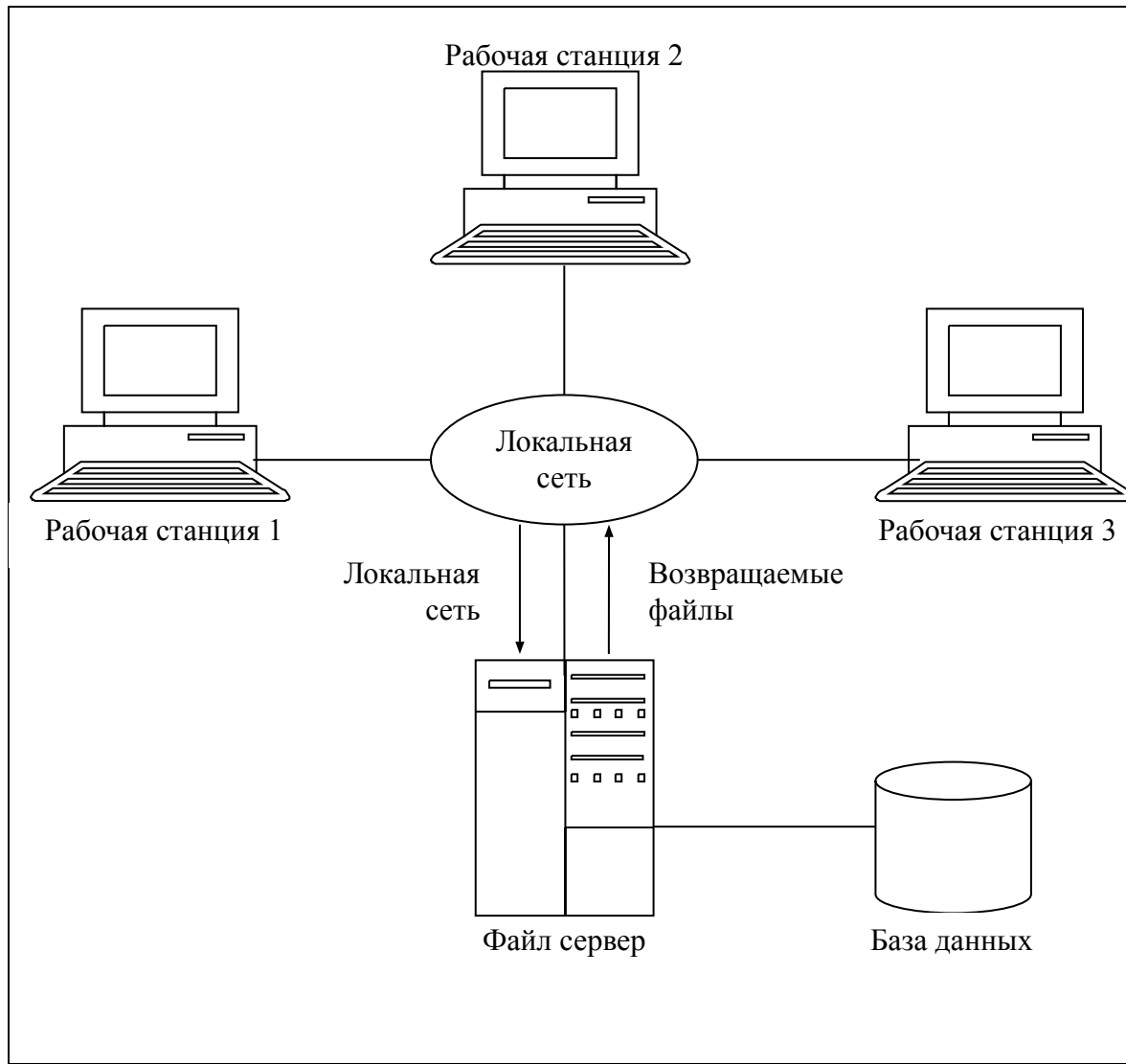
- Принимает и обрабатывает запросы к БД от клиента
- Проверяет полномочия пользователей
- Гарантирует соблюдение ограничений целостности
- Выполняет запросы обновления и возвращает результат клиенту
- Поддерживает системный каталог
- Обеспечивает параллельный доступ к БД

# Файловый сервер

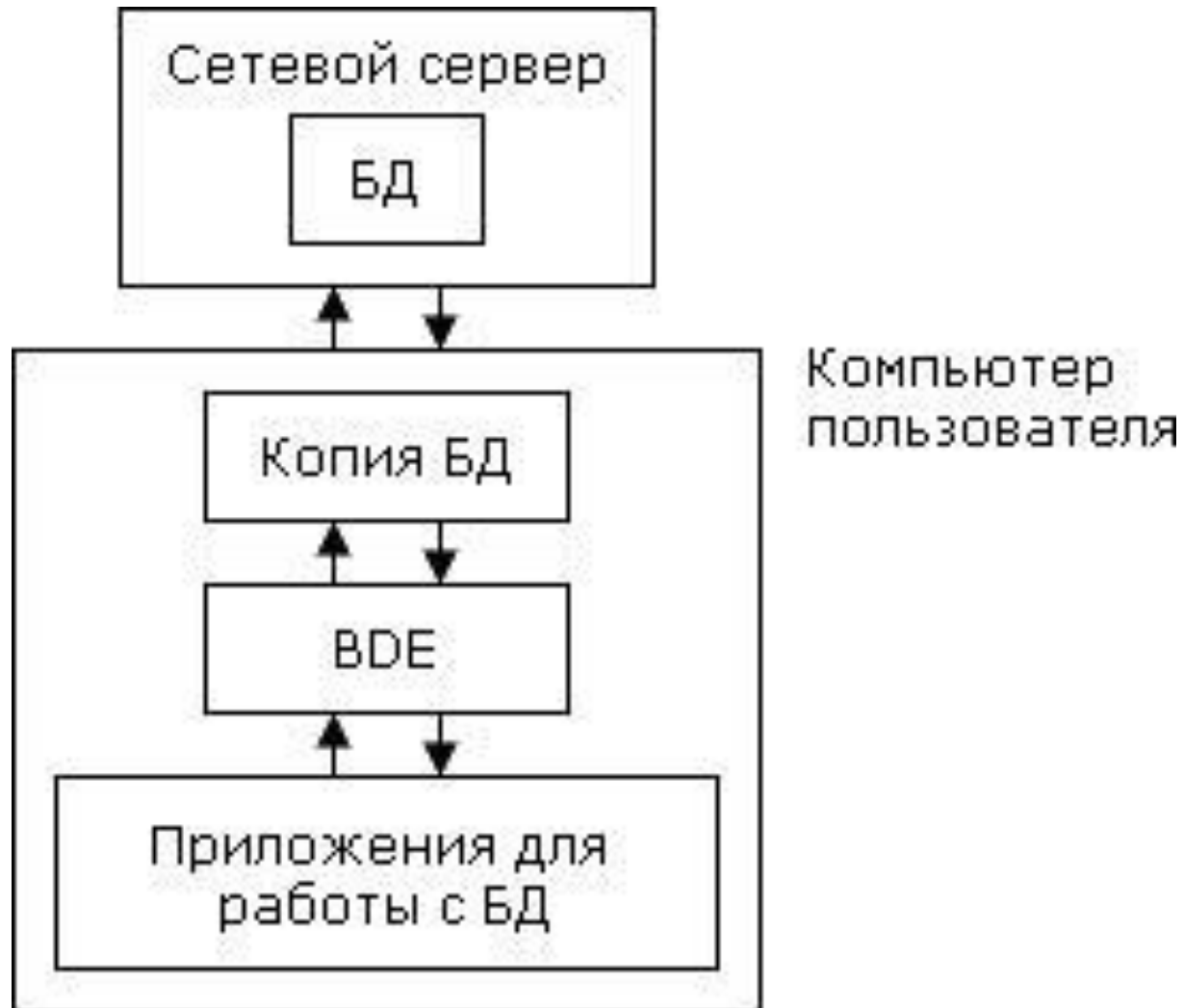
В среде файлового сервера обработка данных распределена в сети, обычно представляющей собой локальную вычислительную сеть (ЛВС). Файловый сервер содержит файлы, необходимые для работы приложений и самой СУБД. Эти приложения и СУБД размещены и функционируют на отдельных рабочих станциях.



# Архитектура с использованием файлового сервера



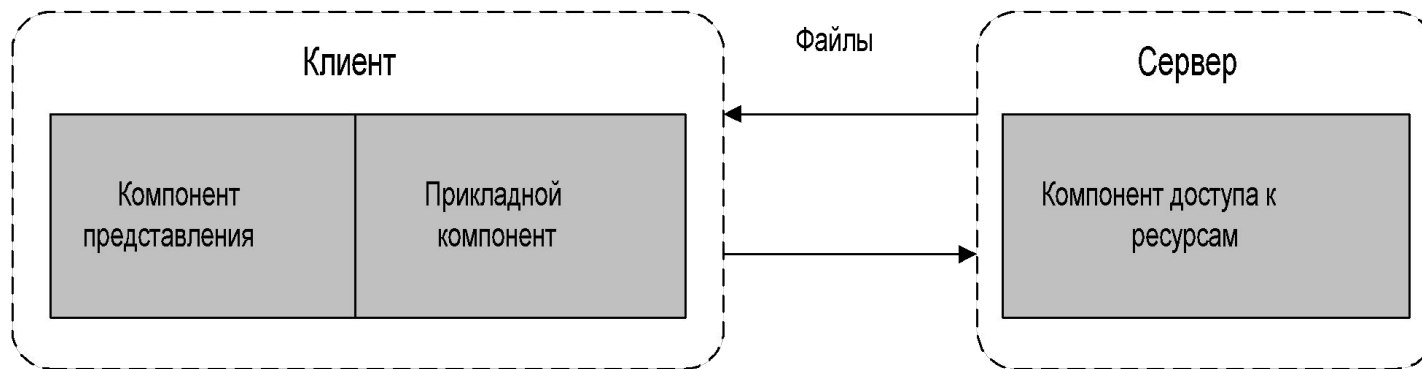
# Архитектура "файл-сервер"



# Архитектура с использованием файлового сервера

Архитектура с использованием файлового сервера обладает следующими *основными недостатками*:

- *большой объем сетевого трафика,*
- *на каждой рабочей станции должна находиться полная копия СУБД,*
- *управление параллельностью, восстановлением и целостностью усложняется, поскольку доступ к одним и тем же файлам могут осуществлять сразу несколько экземпляров СУБД.*

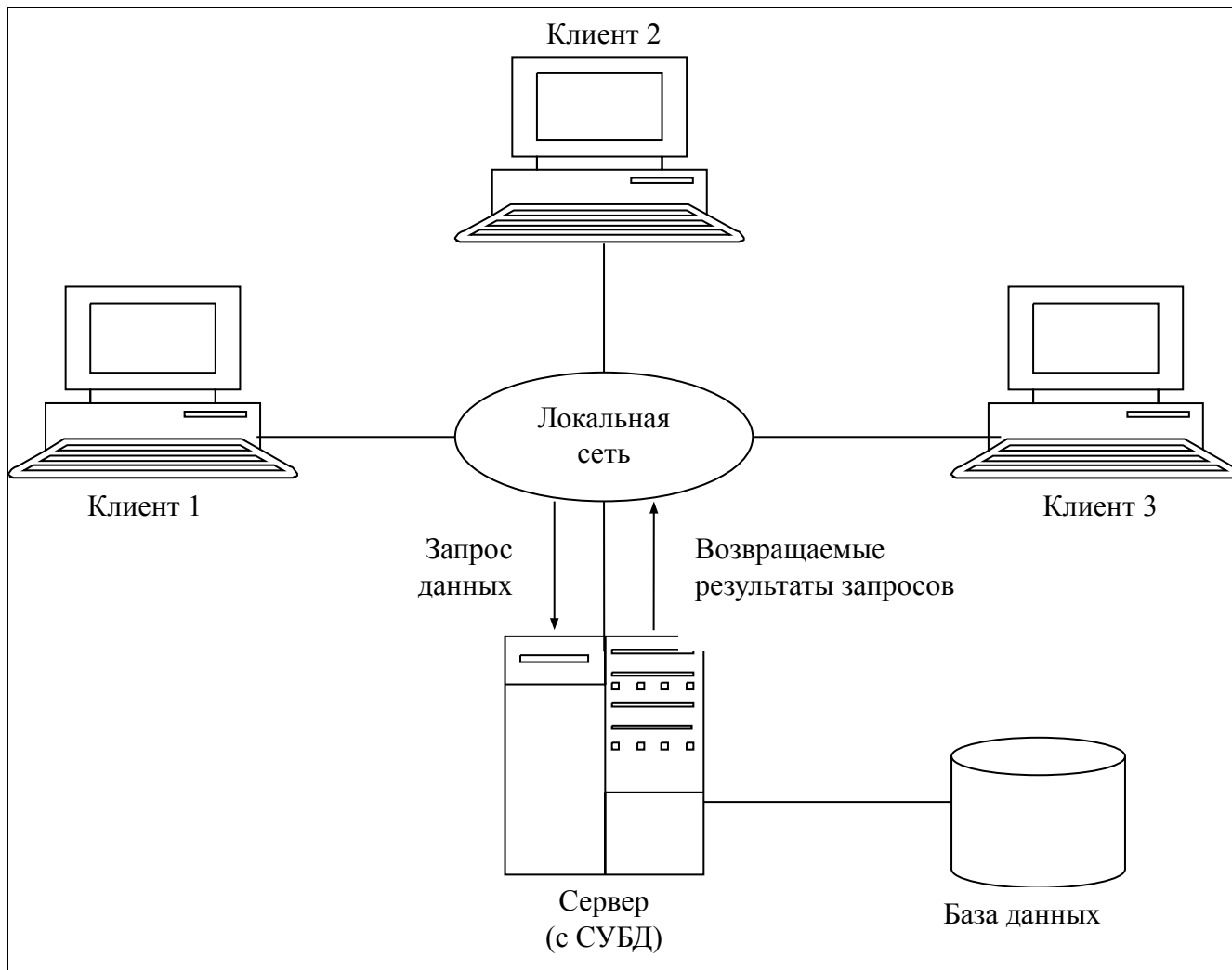


### **Система совместного использования файлов – FS-модель**

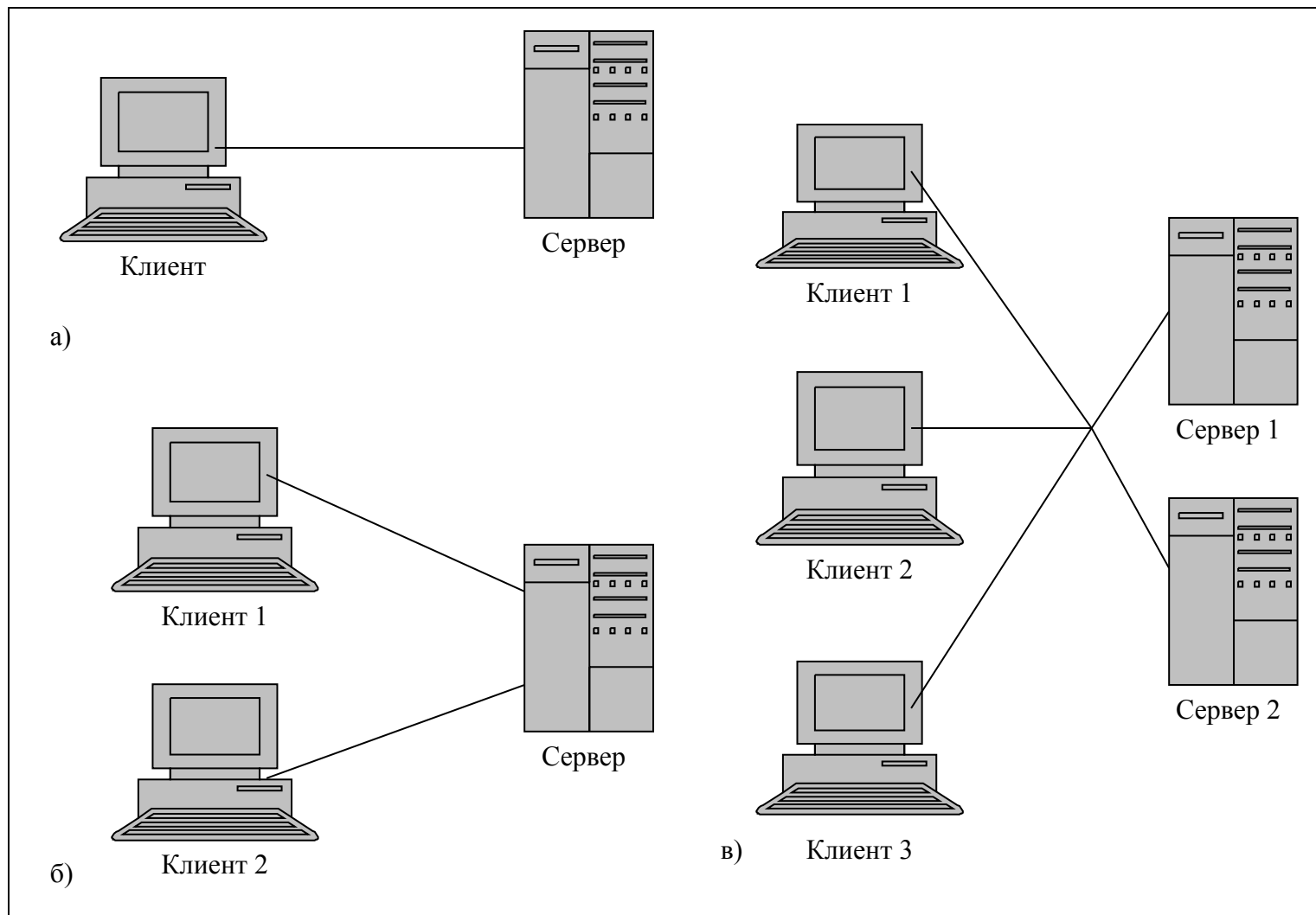
В соответствии с этой моделью один из компьютеров в сети считается файловым сервером и предоставляет услуги по обработке файлов другим компьютерам и играет роль компонента доступа к информационным ресурсам (то есть к файлам). Файл-сервер представляет собой разделяемое всеми РС комплекса расширение дисковой памяти (На других компьютерах в сети функционирует приложение, в кодах которого совмещены компонент представления и прикладной компонент. Протокол обмена представляет собой набор низкоуровневых вызовов, обеспечивающих приложению доступ к файловой системе на файл-сервере.

В целом, в файл-серверной архитектуре имеем "толстого" клиента и очень "тонкий" сервер в том смысле, что почти вся работа выполняется на стороне клиента, а от сервера требуется только достаточная емкость дисковой памяти.

# Общая схема построения систем с архитектурой «клиент-сервер»



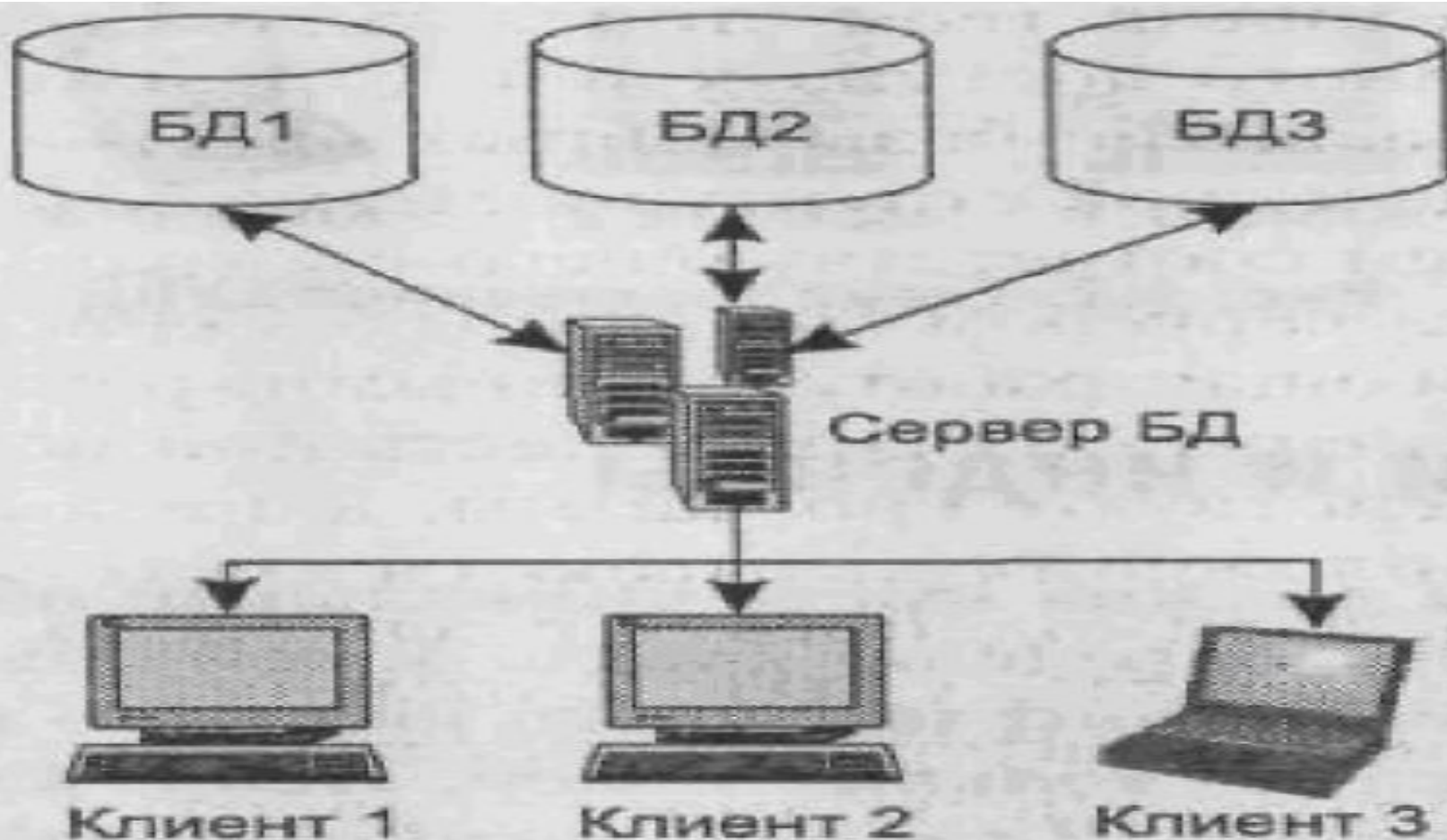
# Альтернативные топологии систем с архитектурой «клиент-сервер»



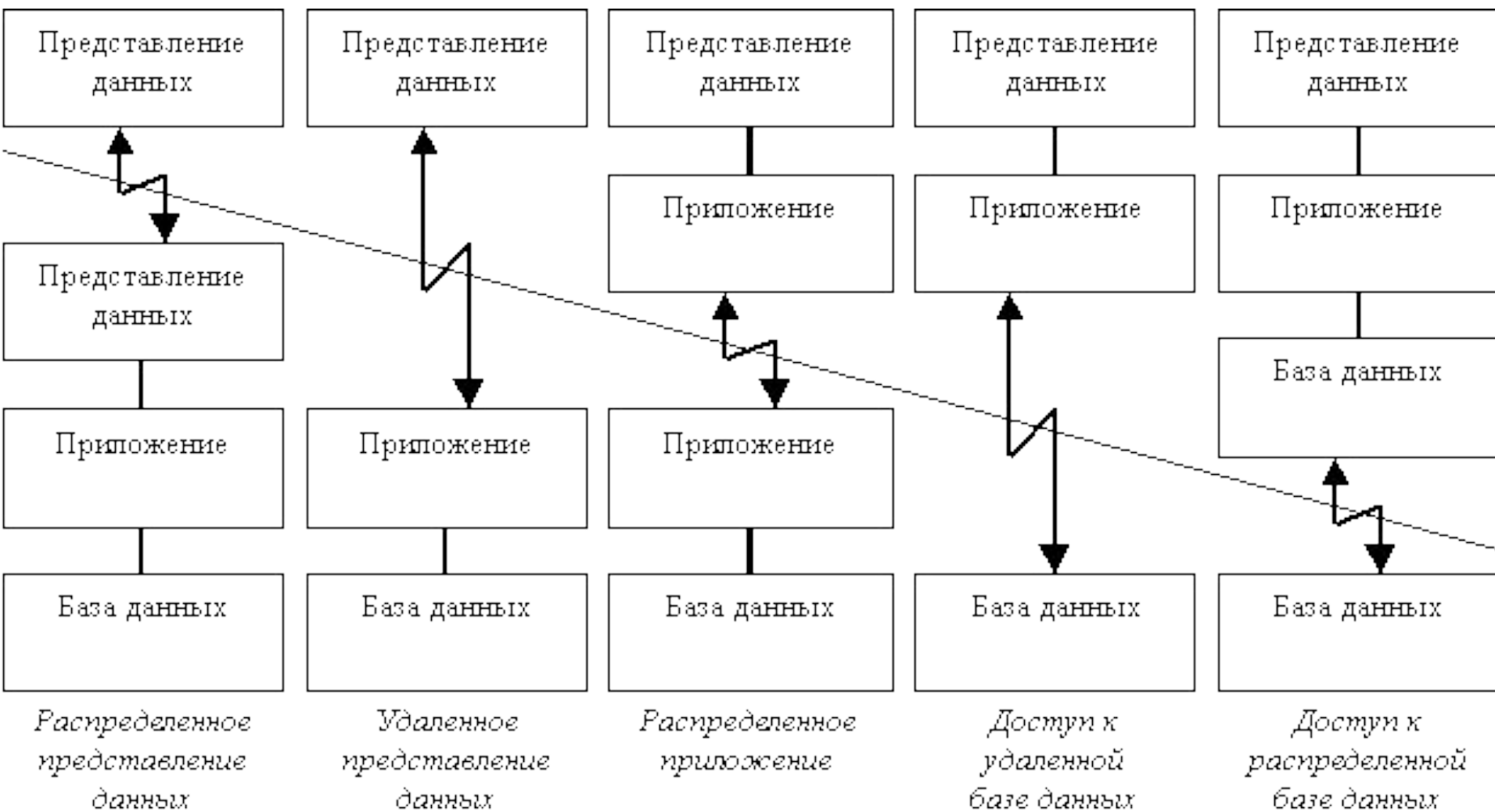
# Функции, выполняемые в среде “клиент/сервер”

Клиент	Сервер
Управляет пользовательским интерфейсом	Принимает и обрабатывает запросы к базе данных со стороны клиентов
Принимает и проверяет синтаксис введенного пользователем запроса	Проверяет полномочия пользователей
Выполняет приложение	Гарантирует соблюдение ограничений целостности
Генерирует запрос к базе данных и передает его серверу	Выполняет запросы/обновления и возвращает результаты клиенту
Отображает полученные данные пользователю	Поддерживает системный каталог Обеспечивает параллельный доступ к БД Обеспечивает управление восстановлением





**Рис. 1.1. Схема двухзвенной СУБД**



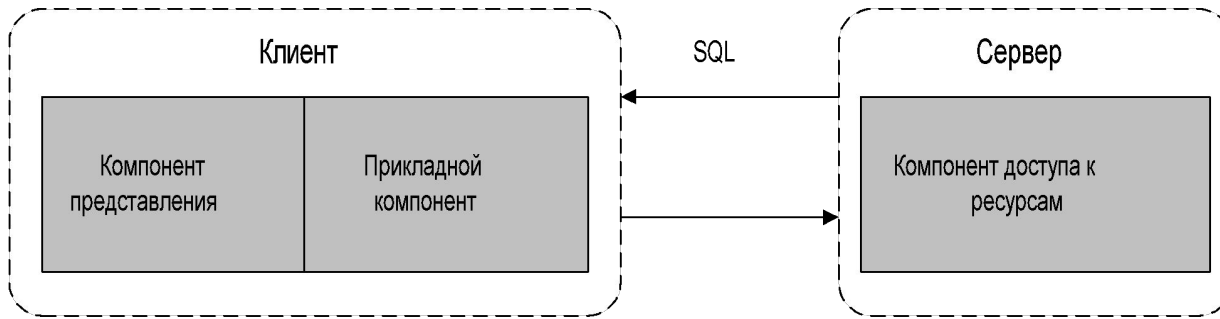
*Распределенное представление данных*

*Удаленное представление данных*

*Распределенное приложение*

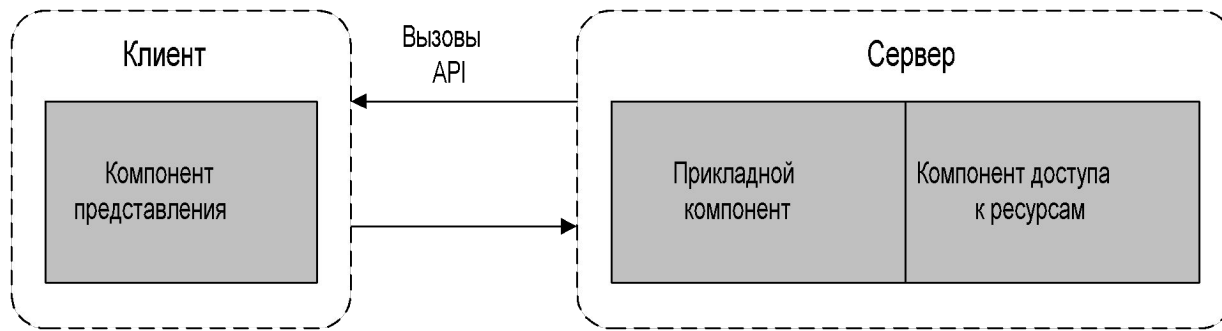
*Доступ к удаленной базе данных*

*Доступ к распределенной базе данных*



Удаленный доступ к данным (Remote Data Access -- RDA) отличается от FS-модели характером компонента доступа к информационным ресурсам. Это, как правило, SQL-сервер.

В RDA-модели программы реализующие функции представления информации(коды компонента представления) и логику прикладной обработки(прикладная компонента), совмещены и выполняются на компьютере-клиенте. Последний поддерживает как функции ввода и отображения данных, так и чисто прикладные функции. Обращение за сервисом управления данными происходит через среду передачи с помощью операторов языка SQL или вызовов функций специальной библиотекой API (Application Programming Interface- интерфейса прикладного программирования).



### Удаленное представление сервера БД ( Data Base Server- DBS)

В DBS-модели компонент представления выполняется на компьютере-клиенте, в то время как прикладной компонент оформлен как набор хранимых процедур и функционирует на компьютере-сервере БД. Там же выполняется компонент доступа к данным, то есть ядро СУБД.

#### Достоинства

- возможность централизованного администрирования приложений (прикладных функций);
- эффективное использование вычислительных и коммуникационных ресурсов.
- снижение трафика (вместо SQL-запросов по сети направляются вызовы хранимых процедур),

#### Недостатки

- ограничение средств разработки хранимых процедур - сильная привязка операторов хранимых процедур к конкретной СУБД
- ограниченность средств, используемых для написания хранимых процедур
- отсутствуют возможности отладки и тестирования разработанных хранимых процедур.

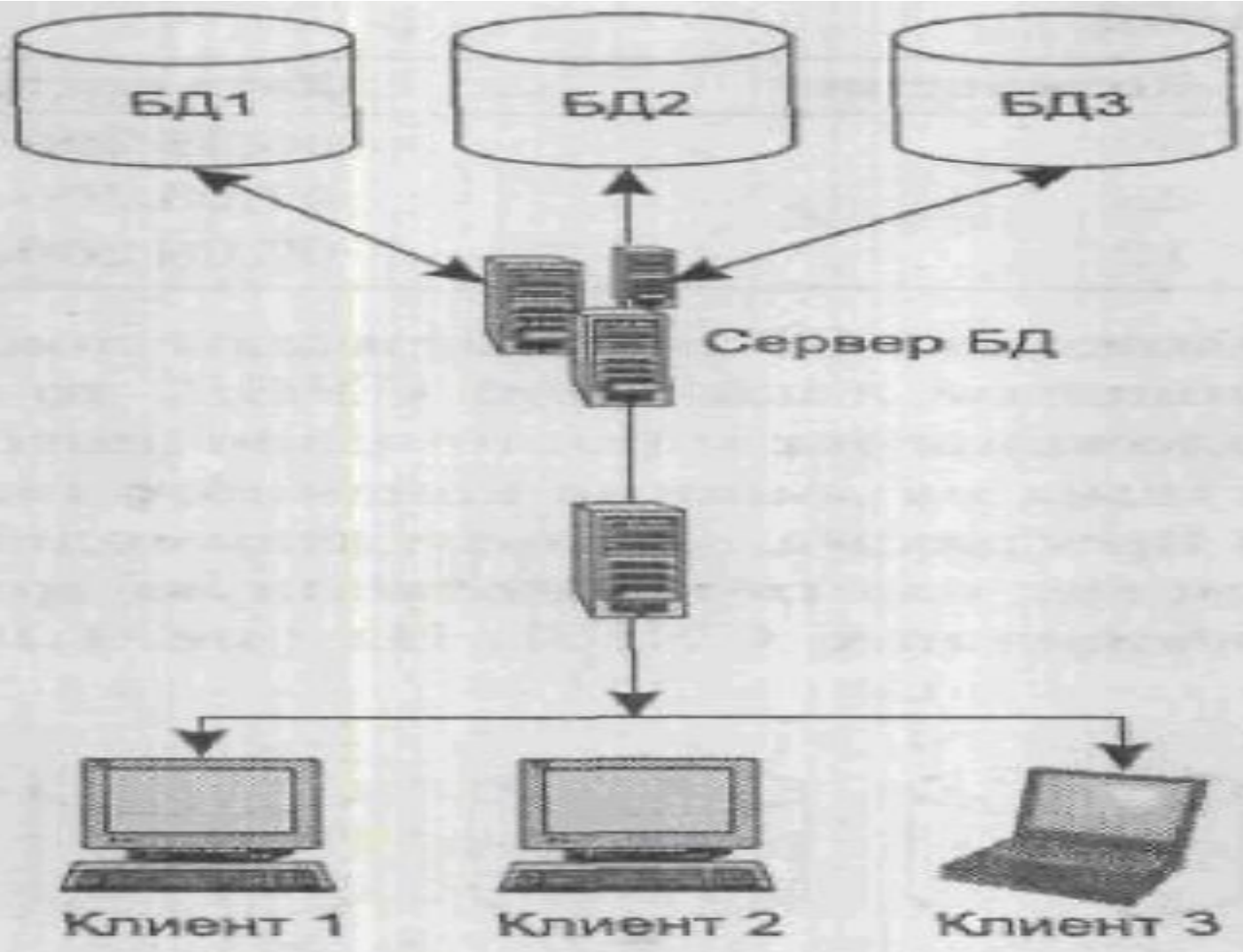
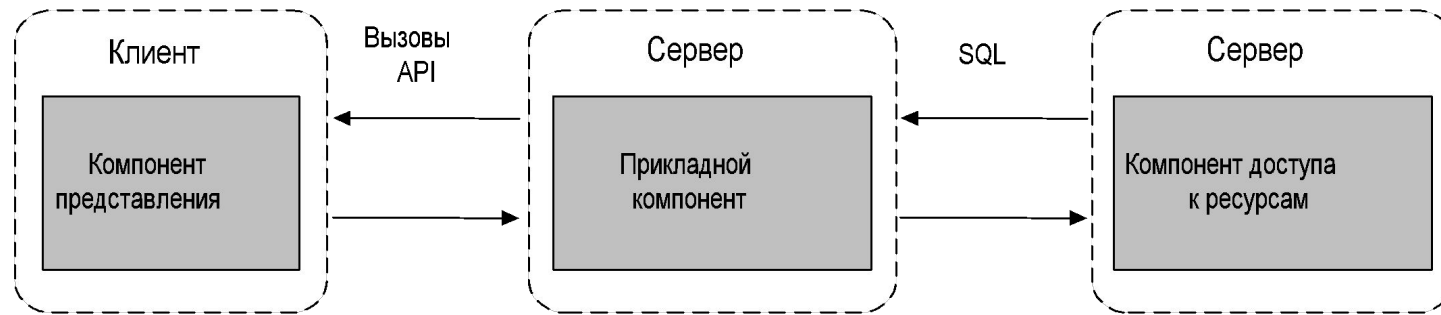


Рис. 1.2. Схема трехзвенной СУБД



## Трехзвенная модель

**В AS-модели** (Application Server) модель сервер-приложения. Каждая из трех функций приложения реализуется на отдельном компьютере. Процесс, выполняющийся на компьютере-клиенте (Application Client - AC), отвечает за интерфейс с пользователем (то есть реализует функции первой группы). Обращаясь за выполнением услуг к прикладному компоненту, этот процесс играет роль клиента приложения. Прикладной компонент реализован как группа процессов, выполняющих прикладные функции и называется сервером приложения (Application Server - AS). Все операции над информационными ресурсами выполняются соответствующим компонентом, по отношению к которому AS играет роль клиента. Из прикладных компонентов доступны ресурсы различных типов - базы данных, очереди, почтовые службы и др.

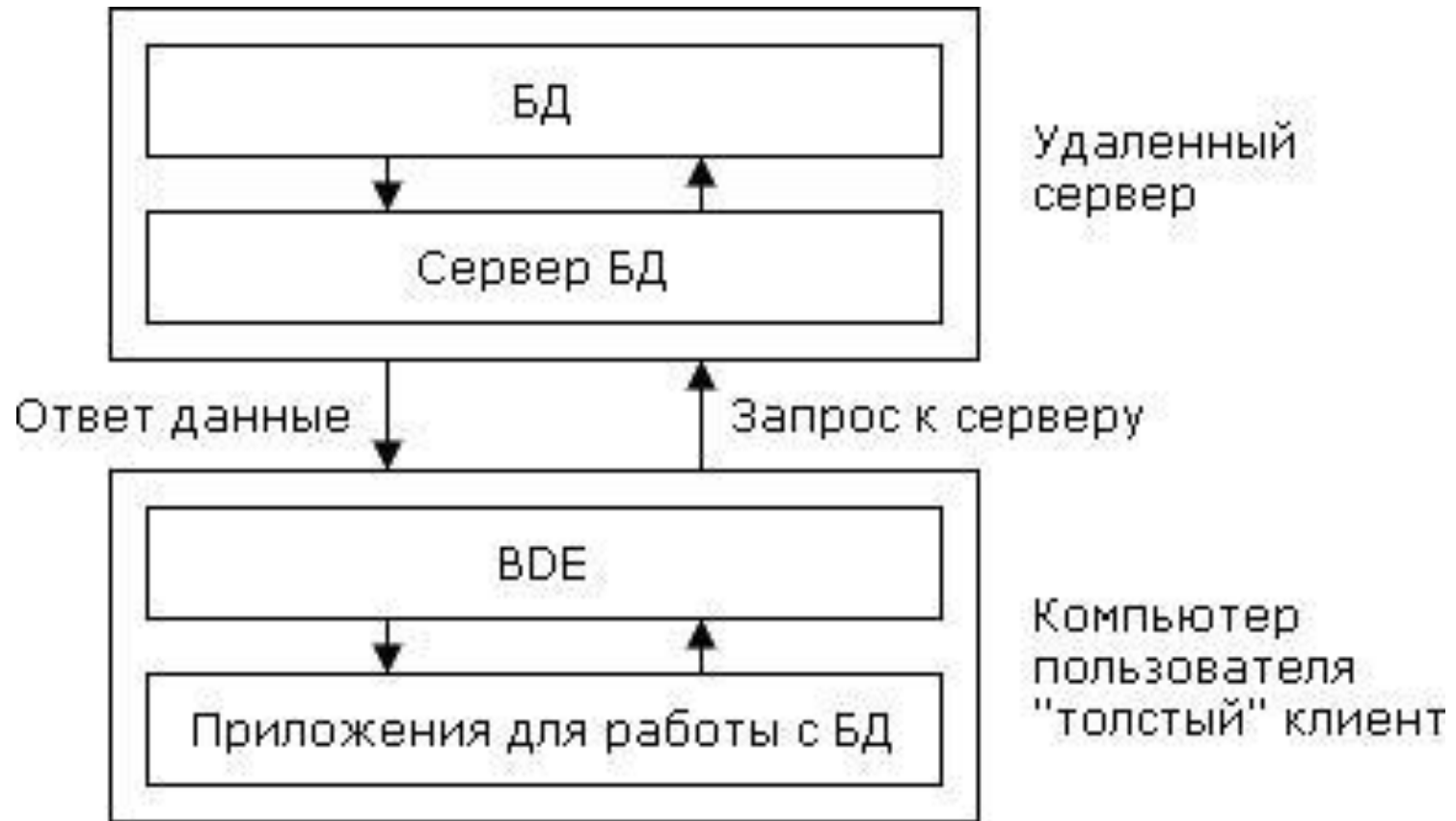
- RDA- и DBS-модели опираются на двухзвенную схему разделения функций.
- В AS-модели реализована трехзвенная схема разделения функций, где прикладной компонент выделен как изолированный элемент приложения, для его определения используются универсальные механизмы многозадачной операционной системы, и стандартизованы интерфейсы с двумя другими компонентами. AS-модель является фундаментом для мониторов обработки транзакций (Transaction Processing Monitors - TPM), или, проще, мониторов транзакций, которые выделяются как особый вид программного обеспечения.

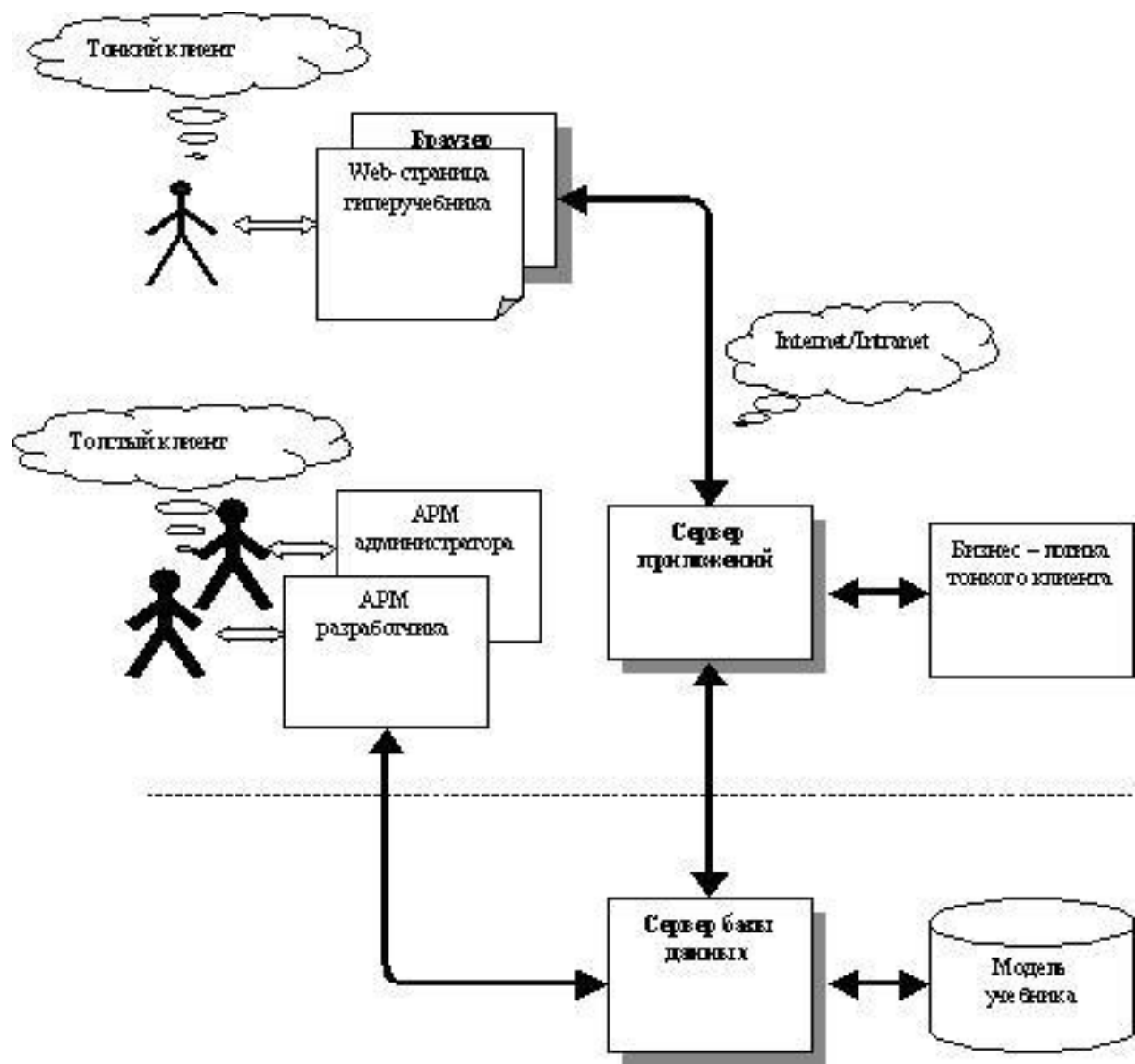
**Компоненты распределенной системы могут быть по-разному реализованы и исполняться в разных узлах сети. Обычно**

Компонент \ Тип архитектуры	Файл-сервер	Клиент-сервер (Бизнес-логика на клиенте)	Клиент-сервер (бизнес-логика на сервере)	N-уровневая архитектура
Представления	Клиент	Клиент	Клиент	Клиент
Бизнес-логики	Клиент	Клиент	Сервер БД	Сервер приложений (комп. кластер)
Хранения	Файл-сервер (или клиент) Все три слоя образуют единый программны	Сервер БД Пользоват. Интерфейс и бизнес-логика образуют единый модуль. Данные хранятся на сервере	Сервер БД Вся бизнес логика реализована в виде хранимых процедур, исполняемых на сервере БД	Сервер БД Все слои исполняются на разных



# Архитектура удаленных БД ("клиент-сервер")





# Системные каталоги

**Системный каталог** – это хранилище данных, которые описывают сохраняемую в базе данных информацию, т.е. метаданные, или "данные о данных".

## Система словаря данных может быть:

**Активной** - система всегда согласуется со структурой базы данных, поскольку она автоматически поддерживается этой системой.

**Пассивной** - система может противоречить состоянию базы данных из-за инициируемых пользователями изменений.

Если словарь данных является частью базы данных, то он называется **интегрированным словарем данных**.