

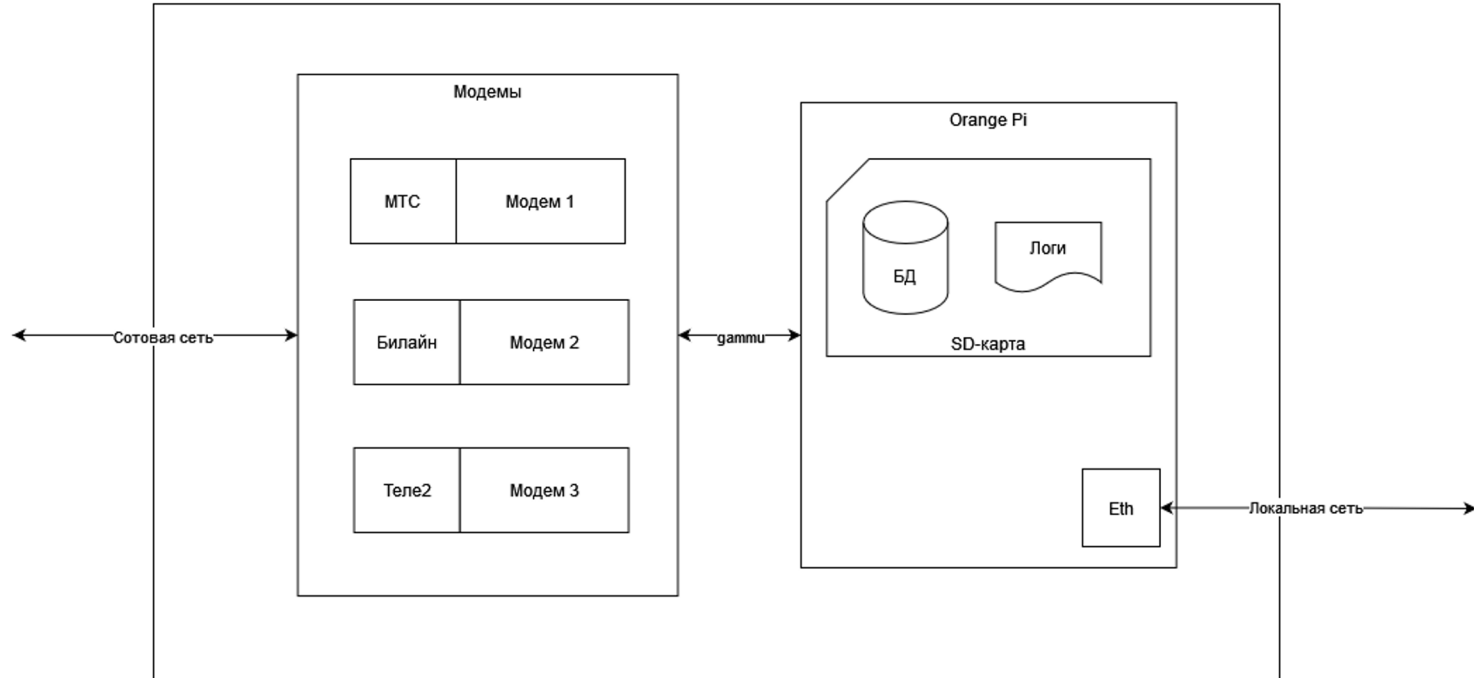
Архитектуры ИС

Архитектура ИС

Архитектура информационной системы - концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

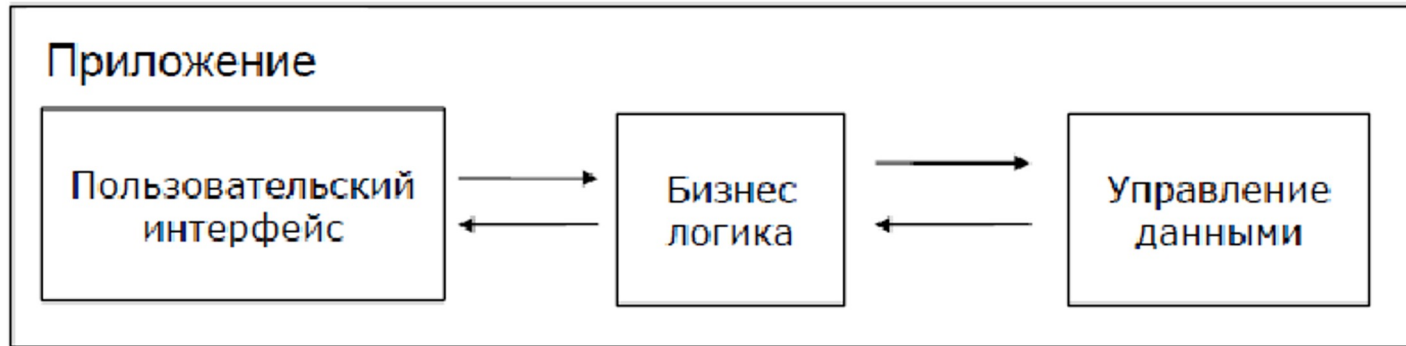


Архитектура ИС



Архитектура ИС

С точки зрения программно-аппаратной реализации можно выделить ряд типовых архитектур ИС.



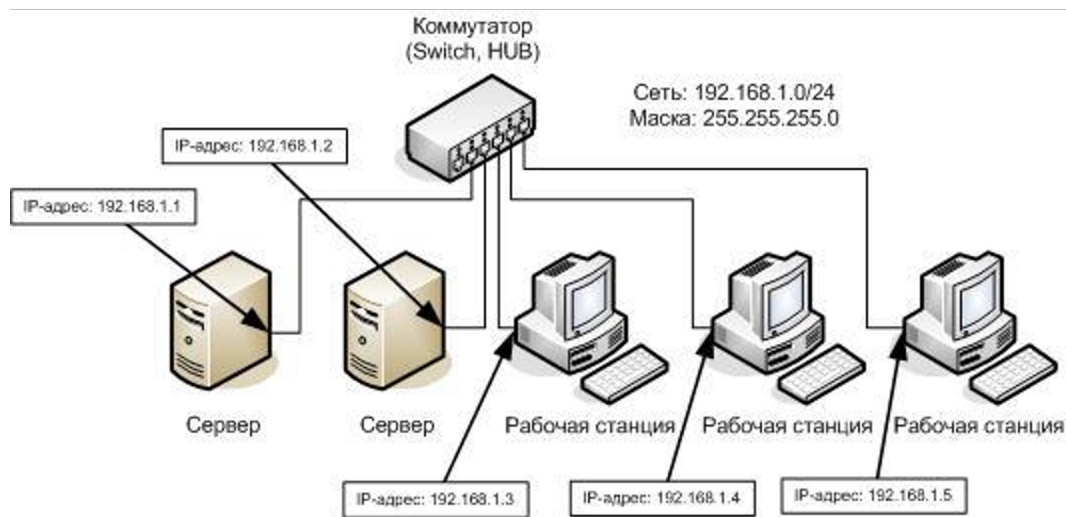
Архитектура ИС

Компоненты информационной системы по выполняемым функциям можно разделить на три слоя: слой представления, слой бизнес-логики и слой доступа к данным.

- **Слой представления** - все, что связано с взаимодействием с пользователем: нажатие кнопок, движение мыши, отрисовка изображения, вывод результатов поиска и т.д.
- **Бизнес логика** - правила, алгоритмы реакции приложения на действия пользователя или на внутренние события, правила обработки данных.
- **Слой доступа к данным** - хранение, выборка, модификация и удаление данных, связанных с решаемой приложением прикладной задачей

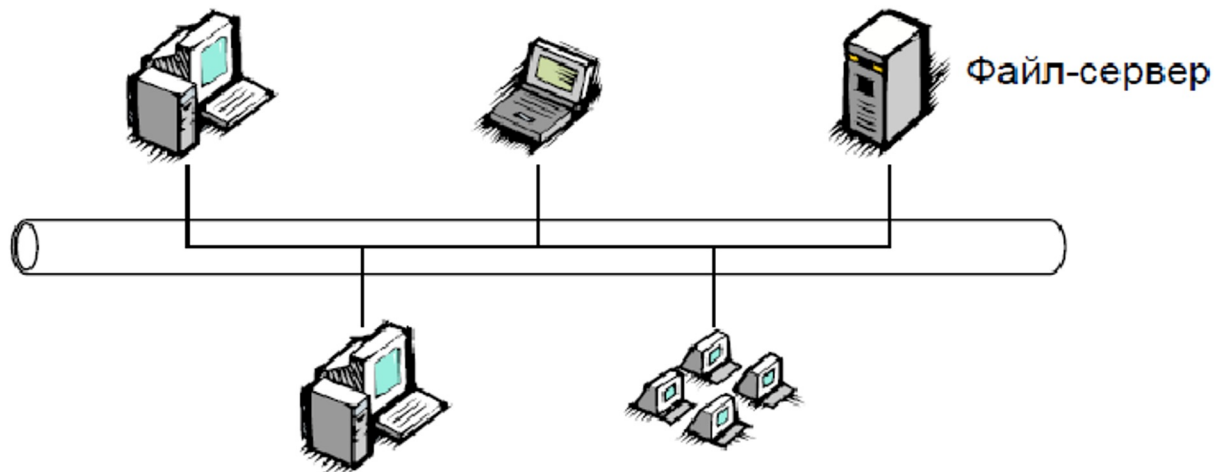
Файл-серверная архитектура

Появились локальные сети. Файлы начали передаваться по локальной сети между различными компьютерами.

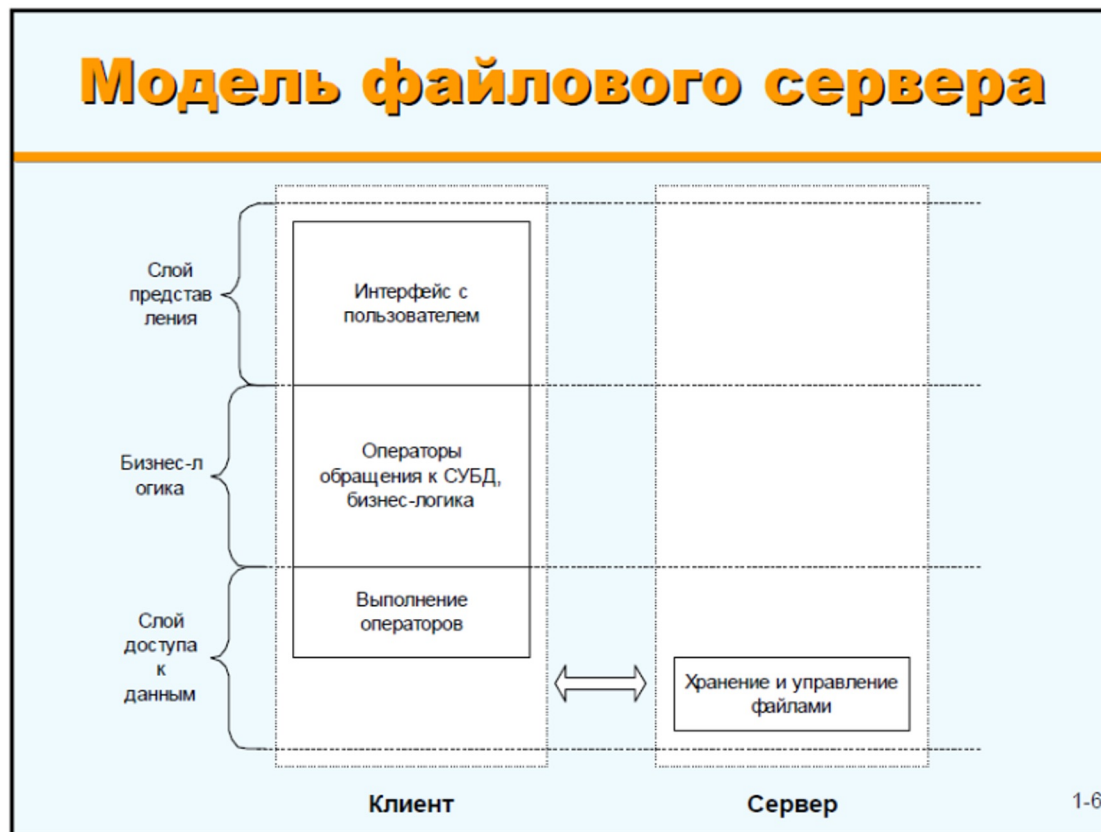


Файл-серверная архитектура

Потом возникла идея хранения всех общедоступных файлов на выделенном компьютере в сети - файл-сервере.



Файл-серверная архитектура



Файл-серверная архитектура

Файл-серверные приложения — приложения, схожие по своей структуре с локальными приложениями и использующие сетевой ресурс для хранения программы и данных. Функции сервера: хранения данных и кода программы. Функции клиента: обработка данных происходит исключительно на стороне клиента.

Количество клиентов ограничено десятками.

Файл-серверная архитектура

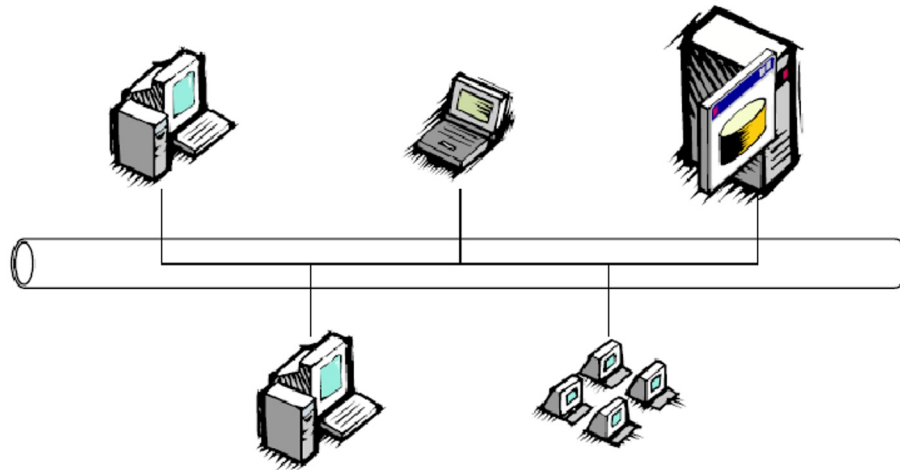
- + Многопользовательский режим работы с данными
 - + Удобство централизованного управления доступом
 - + Низкая стоимость разработки
-
- Низкая производительность
 - Низкая надежность
 - Слабые возможности расширения

Файл-серверная архитектура

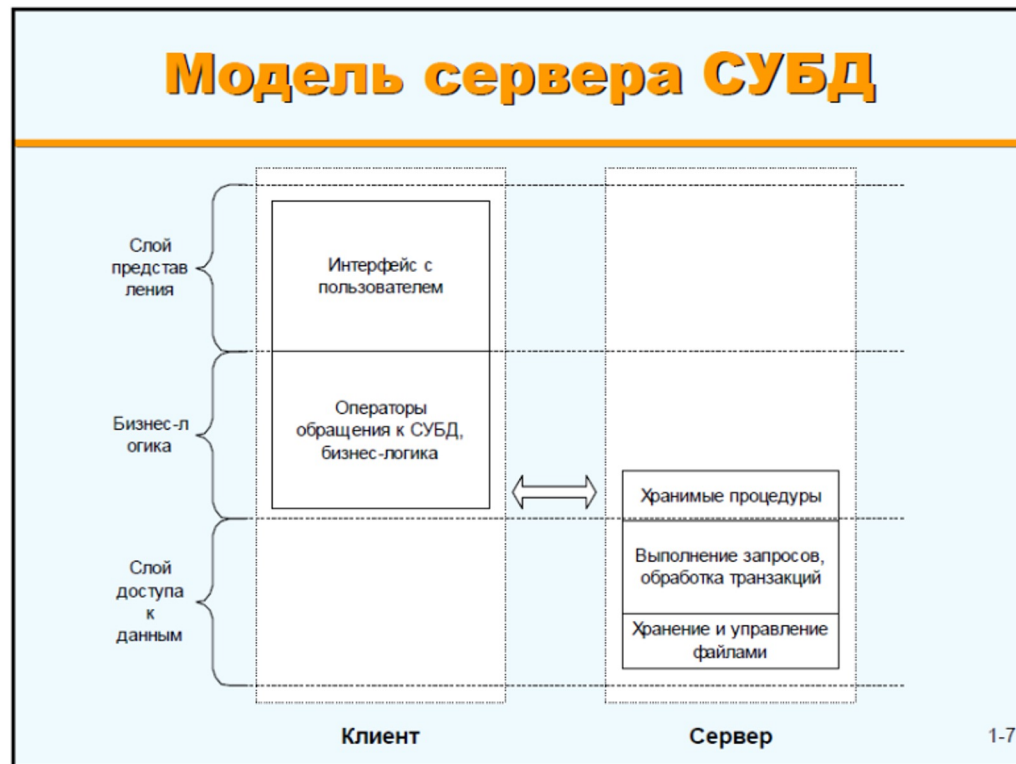
Недостатки архитектуры с файловым сервером очевидны и вытекают главным образом из того, что данные хранятся в одном месте, а обрабатываются в другом. Это означает, что их нужно передавать по сети, что приводит к очень высоким нагрузкам на сеть и, вследствие этого, резкому снижению производительности приложения при увеличении числа одновременно работающих клиентов. Вторым важным недостатком такой архитектуры является децентрализованное решение проблем целостности и согласованности данных и одновременного доступа к данным. Такое решение снижает надежность приложения.

Клиент-серверная архитектура

Ключевым отличием архитектуры клиент-сервер от архитектуры файл-сервер является абстрагирование от внутреннего представления данных (физической схемы данных). Теперь клиентские программы манипулируют данными на уровне логической схемы.



Клиент-серверная архитектура



Клиент-серверная архитектура

- + Полная поддержка многопользовательской работы
- + Гарантия целостности данных

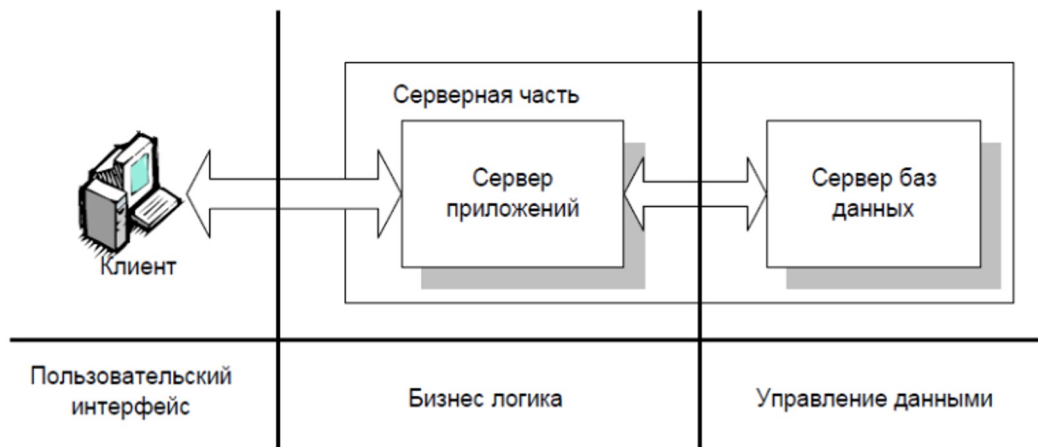
- Бизнес логика приложений осталась в клиентском ПО. При любом изменении алгоритмов, надо обновлять пользовательское ПО на каждом клиенте
- Высокие требования к пропускной способности коммуникационных каналов с сервером, что препятствует использованию клиентских станций иначе как в локальной сети
- Слабая защита данных от взлома, в особенности от недобросовестных пользователей системы
- Высокая сложность администрирования и настройки рабочих мест пользователей
- системы
- Необходимость использовать мощные ПК на клиентских местах
- Высокая сложность разработки системы из-за необходимости выполнять бизнес-логику и обеспечивать пользовательский интерфейс в одной программе

Клиент-серверная архитектура

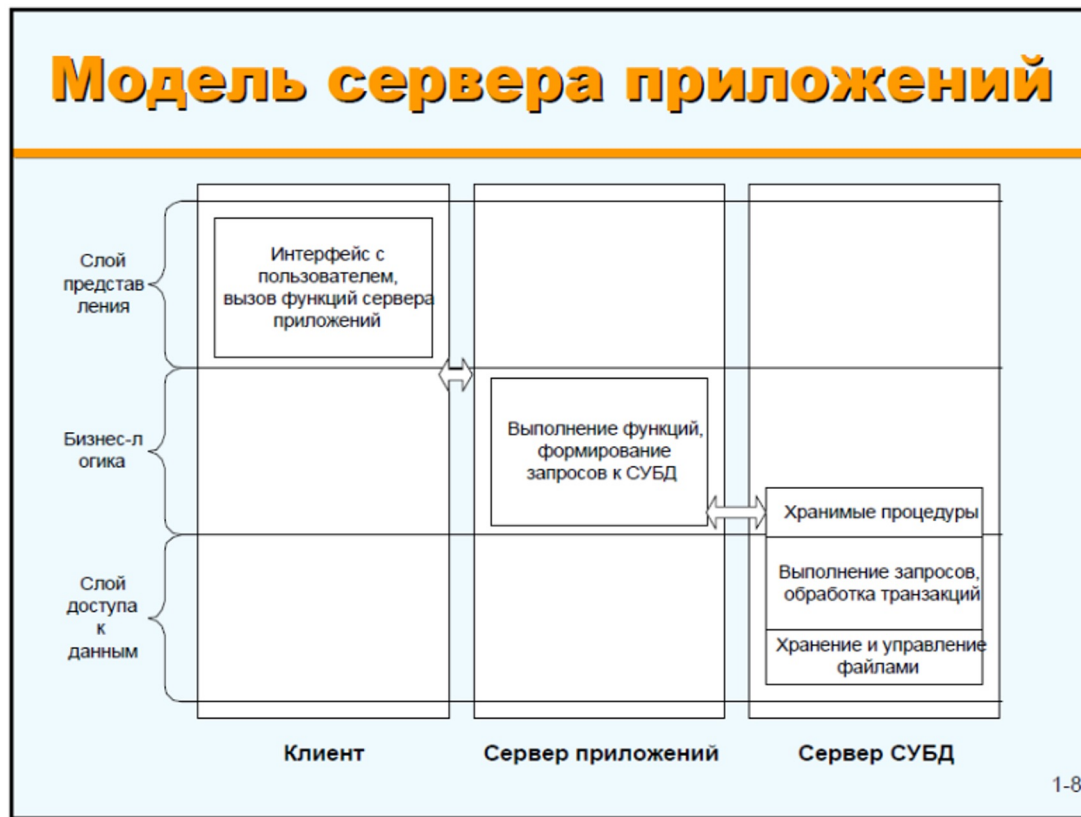
Нетрудно заметить, что большинство недостатков классической или 2-х слойной архитектуры клиент-сервер проистекают от использования клиентской станции в качестве исполнителя бизнес-логики ИС. Поэтому очевидным шагом дальнейшей эволюции архитектур ИС явилась идея "тонкого клиента", то есть разбиения алгоритмов обработки данных на части связанные с выполнением бизнес-функций и связанные с отображением информации в удобном для человека представлении. При этом на клиентской машине оставляют лишь вторую часть, связанную с первичной проверкой и отображением информации, перенося всю реальную функциональность системы на серверную часть

Трехуровневая клиент-серверная архитектура

Для решения этих проблем и была предложена так называемая 3-х слойная архитектура клиент-сервер. Основным ее отличием является физическое разделение программ, отвечающих за хранение данных (СУБД) от программ. Такое разделение программных компонент позволяет оптимизировать нагрузки как на сетевое, так и на вычислительное оборудование комплекса.



Трехуровневая клиент-серверная архитектура



Трехуровневая клиент-серверная архитектура

- + Между клиентской программой и сервером приложения передается лишь минимально необходимый поток данных - аргументы вызываемых функций и возвращаемые от них значения
- + Сервер приложения ИС может быть запущен в одном или нескольких экземплярах на одном или нескольких компьютерах, что позволяет использовать вычислительные мощности организации эффективно
- + Дешевый трафик между сервером приложений и СУБД. Трафик между сервером приложений и СУБД может быть большим, однако это всегда трафик локальной сети, а их пропускная способность достаточно велика и дешева
- + Дешевле наращивать функциональность и обновлять ПО
- Выше расходы на администрирование и обслуживание серверной части

Трехуровневая клиент-серверная архитектура

Масштабируемость систем выполненных в 3-х слойной архитектуре впечатляет. Одна и та же система может работать как на одном отдельно стоящем компьютере, выполняя на нем программы СУБД, СП и клиентской части, так и в сети, состоящей из сотен и тысяч машин. Как уже было отмечено, единственным фактором, препятствующим бесконечной масштабируемости, является лишь требование ведения единой базы данных.