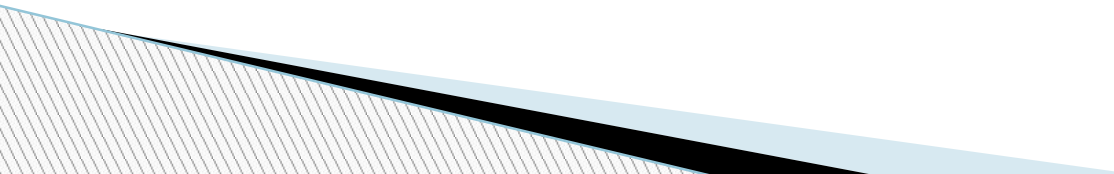


АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ

Тема 4/1 Архитектуры информационных систем

- 1. Классификация информационных систем по архитектуре.**
 - 2. Методы моделирования информационных систем.**
- 

1. Классификация информационных систем по архитектуре

Централизованная архитектура

Архитектура "файл-сервер"

Двухзвенная архитектура "клиент-сервер"

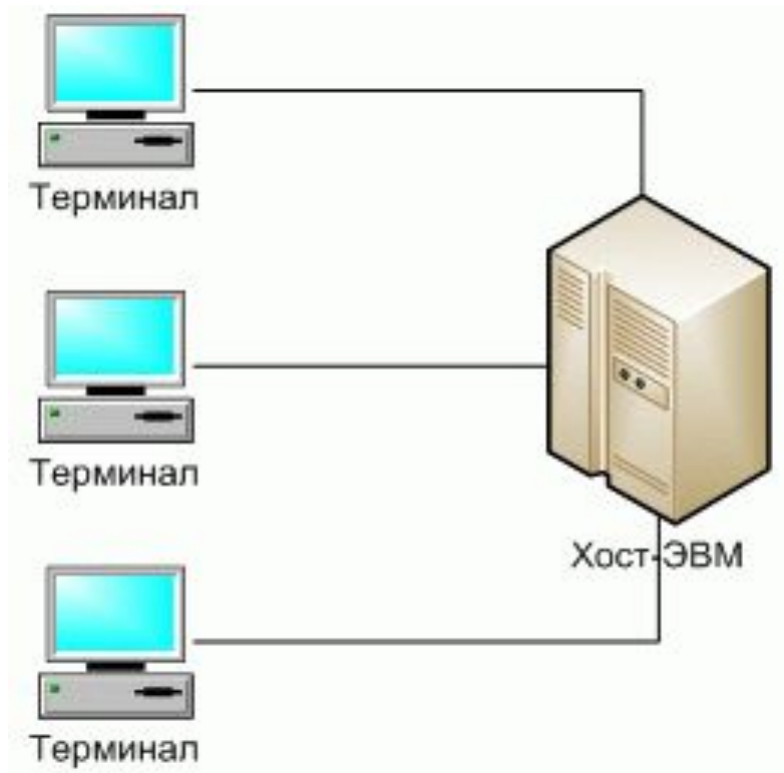
Многозвенная архитектура "клиент-сервер"

Архитектура распределенных систем

Архитектура Веб-приложений

Сервис-ориентированная архитектура

Централизованная архитектура

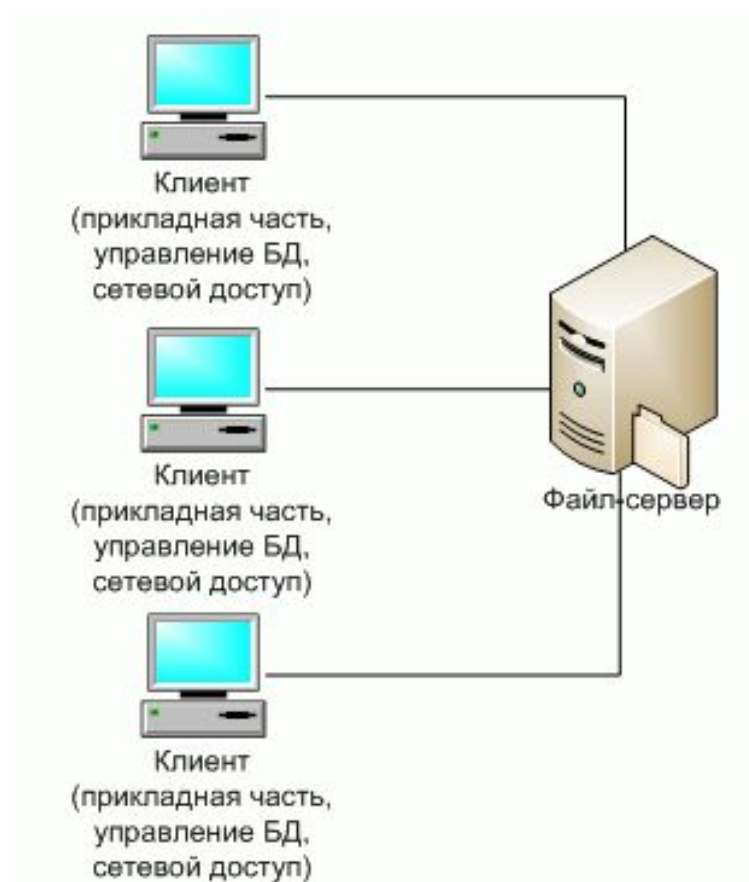


Архитектура "файл-сервер"

Файл-серверные приложения – приложения, схожие по своей структуре с локальными приложениями и использующие сетевой ресурс для хранения программы и данных.

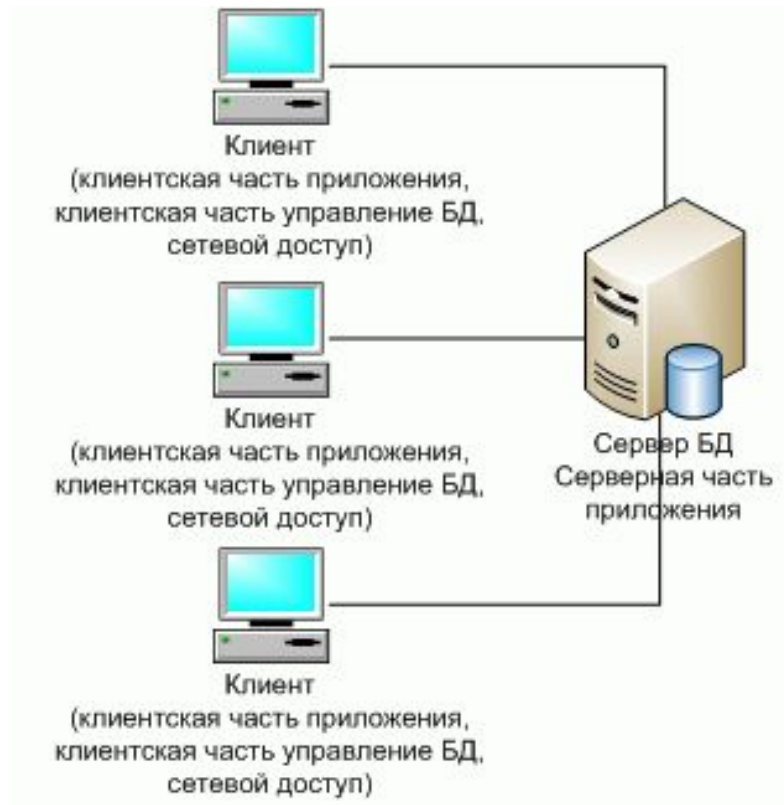
Функции сервера: хранение данных и кода программы.

Функции клиента: обработка данных происходит исключительно на стороне клиента.



Двухзвенная архитектура "клиент-сервер"

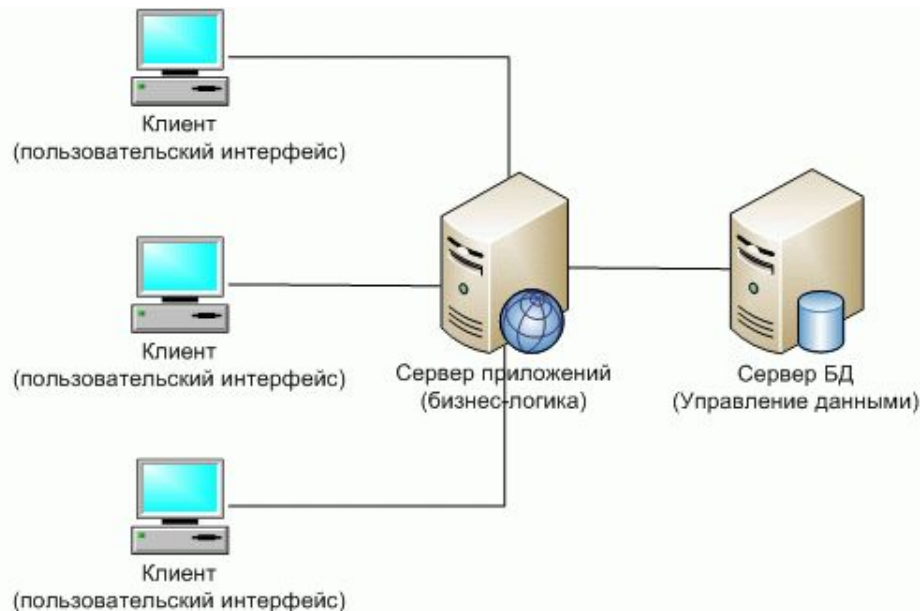
Клиент-сервер (*Client-server*) – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и заказчиками услуг, называемых клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением.



Многозвенная (многоуровневая) архитектура "клиент-сервер"

Многоуровневая архитектура клиент-сервер (*Multitier architecture*) – разновидность архитектуры клиент-сервер, в которой функция обработки данных вынесена на один или несколько отдельных серверов.

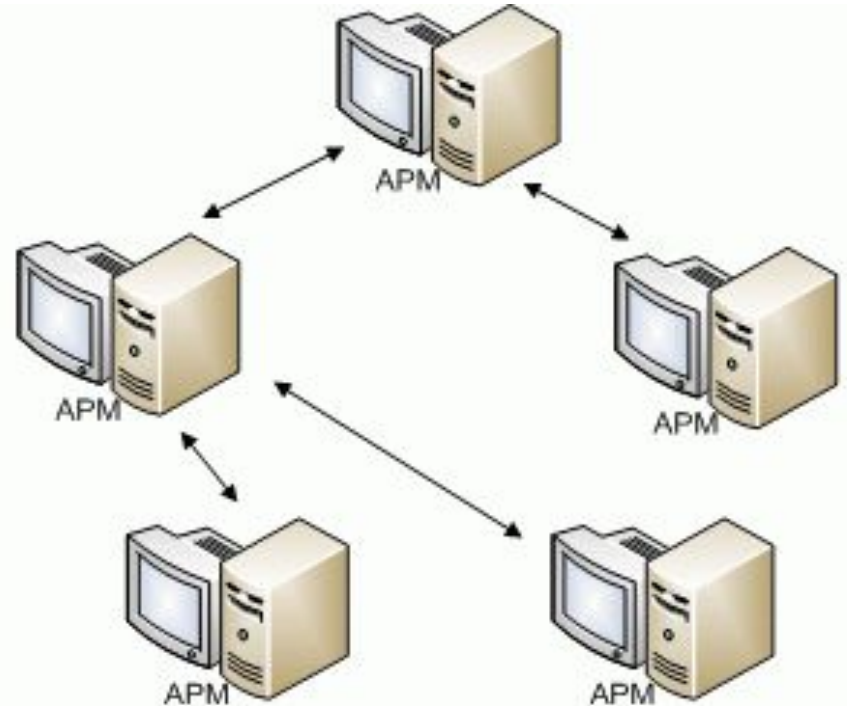
Это позволяет разделить функции хранения, обработки и представления данных для более эффективного использования возможностей серверов и клиентов.



Архитектура распределенных систем

Каждый АРМ независим, содержит только ту информацию, с которой должен работать, а актуальность данных во всей системе обеспечивается благодаря непрерывному обмену сообщениями с другими АРМами.

Обмен сообщениями между АРМами может быть реализован различными способами, от отправки данных по электронной почте до передачи данных по сетям.

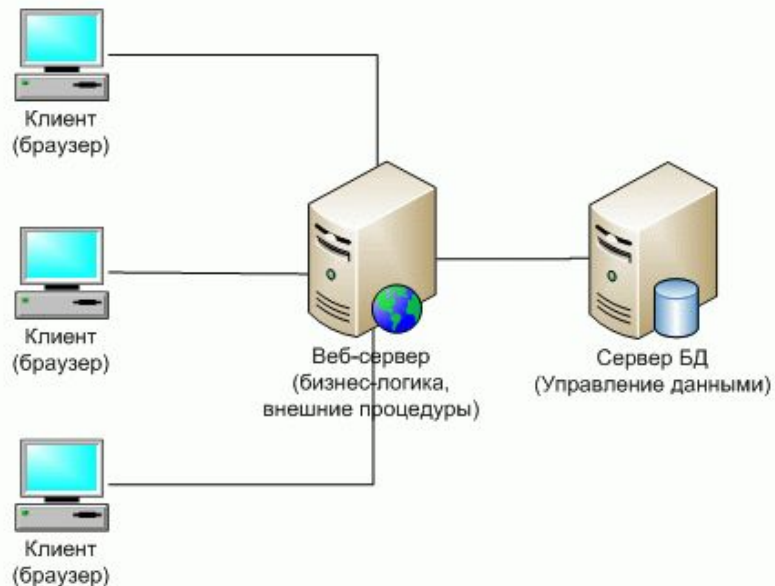


Архитектура Веб-приложений

Обычно Веб-приложения создаются как приложения в архитектуре "клиент-сервер", но серверная часть имеет различные архитектурные решения.

Основные особенности веб-архитектуры:

- отсутствие необходимости использовать дополнительное ПО на стороне клиента – это позволяет автоматически реализовать клиентскую часть на всех платформах;
- возможность подключения практически неограниченного количества клиентов;
- благодаря единственному месту хранения данных и наличия системы управления базами данных обеспечиваются минимальные требования для поддержания целостности данных;
- доступность при работоспособности сервера и каналов связи;
- достаточно низкая скорость Веб сервера и каналов передачи данных.



Сервис-ориентированная архитектура

Сервис-ориентированная архитектура (SOA, service-oriented architecture) – модульный подход к разработке программного обеспечения, основанный на использовании сервисов (служб) со стандартизированными интерфейсами.

Сервис-ориентированная архитектура – это парадигма организации и использования распределенных информационных ресурсов таких как: приложения и данные, находящихся в сфере ответственности разных владельцев, для достижения желаемых результатов потребителем, которым может быть: конечный пользователь или другое приложение.

В основе SOA лежат принципы многократного использования функциональных элементов ИТ, ликвидации дублирования функциональности в ПО, унификации типовых операционных процессов, обеспечения перевода операционной модели компании на централизованные процессы и *функциональную организацию* на основе промышленной платформы интеграции.

Компоненты программы могут быть распределены по разным узлам сети, и предлагаются как независимые, слабо связанные, заменяемые сервисы-приложения.

Программные комплексы, разработанные в соответствии с SOA, часто реализуются как набор веб-сервисов, интегрированных при помощи известных стандартных протоколов (SOAP, WSDL, и т. п.)

Таким образом, **SOA** предоставляет гибкий и элегантный способ комбинирования и многократного использования компонентов для построения сложных распределенных программных комплексов.

SOA хорошо зарекомендовала себя для построения крупных корпоративных программных приложений.

2. Методы моделирования информационных систем

На сегодняшний день существует достаточно большое количество методов моделирования бизнес-процессов, которые могут использоваться для моделирования соответствующих им информационных систем. Эти методы относятся к разным видам моделирования и позволяют сфокусировать внимание на различных аспектах. Они содержат как графические, так и текстовые средства, за счет которых можно наглядно представить основные компоненты процесса и дать точные определения параметров и связей элементов. Наиболее часто в *менеджменте качества* моделирование бизнес-процессов выполняют с помощью следующих методов:

Flow Chart Diagram (диаграмма потока работ) – это графический метод представления процесса в котором операции, данные, оборудование процесса и пр. изображаются специальными символами. Метод применяется для отображения логической последовательности действий процесса. Главным достоинством метода является его гибкость. Процесс может быть представлен множеством способов.

Data Flow Diagram (диаграмма потока данных). Диаграмма потока данных или DFD применяется для отображения передачи информации (данных) от одной операции процесса к другой. DFD описывает взаимосвязь операций за счет информации и данных. Этот метод является основой структурного анализа процессов, т.к. позволяет разложить процесс на логические уровни. Каждый процесс может быть разбит на подпроцессы с более высоким уровнем детализации. Применение DFD позволяет отразить только поток информации, но не поток материалов. Диаграмма потока данных показывает, как информация входит и выходит из процесса, какие действия изменяют информацию, где информация хранится в процессе и пр.

Role Activity Diagram (диаграмма ролей). Она применяется для моделирования процесса с точки зрения отдельных ролей, групп ролей и взаимодействия ролей в процессе. Роль представляет собой абстрактный элемент процесса, выполняющий какую-либо организационную функцию. Диаграмма ролей показывает степень «ответственности» за процесс и его операции, а также взаимодействие ролей.

IDEF (Integrated Definition for Function Modeling) – представляет собой целый набор методов для описания различных аспектов бизнес-процессов (IDEF0, IDEF1, IDEF1X, IDEF2, IDEF3, IDEF4, IDEF5). Эти методы строятся на базе методологии SADT (Structured Analysis and Design Technique). Для моделирования бизнес-процессов наиболее часто применяют методы IDEF0 и IDEF3.

IDEF0 – позволяет создать модель функций процесса. На диаграмме IDEF0 отображаются основные функции процесса, входы, выходы, управляющие воздействия и устройства, взаимосвязанные с основными функциями. Процесс может быть декомпозирован на более низкий уровень.

IDEF3 – этот метод позволяет создать «поведенческую» модель процесса. IDEF3 состоит из двух видов моделей. Первый вид представляет описание потока работ. Второй – описание состояний перехода объектов.

Цветные сети Петри – этот метод представляет модель процесса в виде графа, где вершинами являются действия процесса, а дугами события, за счет которых осуществляется переход процесса из одного состояния в другое. Сети Петри применяют для динамического моделирования поведения процесса.

Unified Modeling Language (UML) - представляет собой объектно-ориентированный метод моделирования процессов. Он состоит из 9-ти различных диаграмм, каждая из которых позволяет моделировать отдельные статические или динамические аспекты процесса.

Большинство из указанных методов реализованы в виде программного обеспечения. Оно позволяет осуществлять поддержку бизнес-процессов или проводить их анализ. Примерами такого ПО являются различные **CASE средства** моделирования процессов.