Кто изобрел интернет?

-Тим Бернерс-Ли

Инфраструктура ИС.

ИС – организационно упорядоченная совокупность документов, массивов документов и информационных технологий в том числе использованием средств вычислительной техники и связи, реализующих информационные процессы. ИС предназначены для хранения, обработки, поиска, распространения, передачи и предоставления информации. Архитектура ИС – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов ИС.

Разница между передачей и распространением информации. Также под архитектурой можно понимать абстрактное понятие из каких элементов, компонент состоит приложение и как эти части между собой взаимодействуют.

Составные части программы или программные модули, выполняющие отдельные изолированные задачи. Приложение, реализующее ИС состоит из UI, БЛ(Бизнес-логика) и УД, которые взаимодействуют между собой. Эти компоненты называют слоями ИС. Пользовательский интерфейс – слой представления, отвечает за взаимодействие системы с пользователем.

Документ содержит реквизиты, раскрывающие содержимое документа. Отчеты хранятся в БД тогда, когда надо хранить документы.

Бизнес-логика – правила алгоритма реакции приложения на действия пользователя или на внутренние события, по факту – правила обработки данных.

Способы доступа к данным – хранение, выбор, модификацию и удаление данных, связанных с решаемой прикладной задачей.

Классификация архитектур ИС.

Путь развития архитектуры:

1. Файл-серверная архитектура превратилась в клиент-серверную, которая была представлена однозвенной, двухуровневой, 2,5 уровневой, трехуровневой привела к горизонтальным и вертикальным распределениям.

Все общедоступные файлы хранятся на выделенном компьютере (Файл-сервере). Файл-серверное приложение – приложение-ресурс для хранения программ и данных. Функции сервера – хранения данных и кода программы. Функции клиента – обработка данных. Количество клиентов, использующим сервер ограничено десятками.

При файл-серверной архитектуре UI, БЛ (реализуемая через оператора обращения к СУБД), а также алгоритмы бизнес-логики и выполнение операторов выполняется на компьютере клиента.

Сервер отвечает за хранение и управление файлами.

Положительные стороны: многопользовательский режим работы с данными, удобство централизованного управления доступом, низкая стоимость разработки.

Отрицательные стороны: низкая производительность, низкая надежность, слабые возможности расширения (данные хранятся в одном месте, а обрабатываются в другом -> высокая нагрузка на передачу данных), децентрализованное решение целостности и согласованности данных.

2. Клиент-серверная архитектура ключевое отличие – абстрагирование от физической схемы данных и манипулирование данными клиентскими программами на уровне логической схемы. Слой доступа к данным полностью реализуется на сервере, выполнение данных и транзакции – серверная часть. Транзакция – последовательность операций, которое имеет смысл только после всех транзакционных операций (Commit). Её отмена означает полную отмену всех операций.

Особенности – базовые возможности разделены между клиентом и сервером.

Положительные стороны – полная поддержка, гарантия целостности данных.

Обратная сторона – бизнес-логика на клиенте, при изменении алгоритмов необходимо обновить пользовательское ПО на каждом клиенте, слабая защита данных от взлома, сложность администрирования настройки. Для устранения этих недостатков клиент-серверная структура перешла в двух-с-половиной слойную архитектуру. Алгоритм обработки данных разбивается на части, связанные с отображением информации. Часть, связанная с первичной обработкой и отображением, оставалась на клиенте, реальная функциональность – серверная часть.   
Особенности 2.5 слойности – хранение и вычисление данных на стороне сервера. Специализированные СУБД не позволяют реализовать бизнес-логику -> её часть реализуется на стороне клиента. Физическая сторона состоит из 2х компонентов: положительные стороны: снижается напряга на сеть, пользователь имеет доступ к функциям, а не к данным.

Отрицательная сторона: ограниченная масштабируемость, зависимость от программной платформы. Ограниченное использование сетевых вычислительных ресурсов, реализация бизнес-логики в виде хранения процедур, низкое быстродействие, высокая стоимость ПО и аппаратной части. Трехуровневая клиент-серверная архитектура – бизнес-логика реализуется ... (где?) Пользователь взаимодействует с сервером приложения. Клиент отвечает за UI и вызов функций сервером приложения.

Положительные стороны: тонкий клиент (между клиентом и сервером приложения минимальный обмен информации). Сервер приложения может быть запущен в одном или нескольких экземплярах + дешевый траффик между сервером приложения и сервером СУБД (зачастую траффик локальной сети)  
Отрицательные стороны: повышенные расходы на администрирование

Особенности: широкое масштабирование, упрощение функционала.

Многозвенные архитектуры – вертикальные распределенные системы (с логической точки зрения компоненты различаются) горизонтальные распределенные системы (клиент или сервер могут содержать физически разделенные части логически однородного модуля, причем работа каждой из частей может происходить независимо). Пример горизонтального распределения: использование рейд-массивов различного уровня.