



Московский авиационный институт  
(национальный исследовательский университет)



Факультет №4 «Радиоэлектроника летательных аппаратов»

Направление подготовки: Безопасность информационных технологий в правоохранительной сфере

Специализация подготовки: Информационно-аналитическое обеспечение правоохранительной деятельности с использованием космических технологий

Дисциплина:

**БАЗЫ ДАННЫХ**

СЕРПУХОВ 2013



Московский авиационный институт  
(национальный исследовательский университет)



**Факультет №8** Прикладная математика и физика

Направление подготовки: Прикладная математика и информатика

Специализация подготовки: Информатика

Дисциплина: **ПРОЕКТИРОВАНИЕ БАЗ  
ДААННЫХ**

СЕРПУХОВ 2016



## Лекция 10 Распределённые базы данных

### Учебные вопросы:

10.1 Общие понятия распределённых баз данных

10.2 Архитектура распределённых СУБД

### Литература:

Швецов В.И., Визгунов А.Н., Мееров И.Б. Базы данных. Учебное пособие.  
Нижний Новгород: Изд-во ННГУ, 2004.



## 10.1 Общие понятия распределённых баз данных



Мы можем определить распределенные базы данных как совокупность множества логически взаимосвязанных баз данных, распределенных в вычислительной сети (на различных вычислительных установках);

- Системы управления распределенными базами данных (РСУБД - distributed DBMS) определяются как программные системы, которые позволяют управлять распределенными базами данных и обеспечивать прозрачность (transparent) для пользователей;
- Важнейшими понятиями РБД являются «логическая взаимосвязь» и «распределение по компьютерной сети»;
- В значительном числе случаев физически распределенные данные не являются наиболее важным признаком РБД, т.к. распределенными базами данных могут быть две связанные по данным БД расположенные на одной и той же вычислительной системе (сервере);
- Физически распределенные БД создают проблемы специфические только для них и отличающиеся от особенностей БД размещенных на одной установке;
- Распределенная обработка возможна на различных процессорах использующих одну и ту же оперативную память и системы ввода/вывода;



## Способы реализации распределенных БД:

- распределенные многопроцессорные системы;
- разделяемая общая память для нескольких вычислительных систем;
- многомашинные комплексы;
- централизованные базы данных в сети;
- распределенные базы данных в сети;
- распределенные БД в ЛВС.

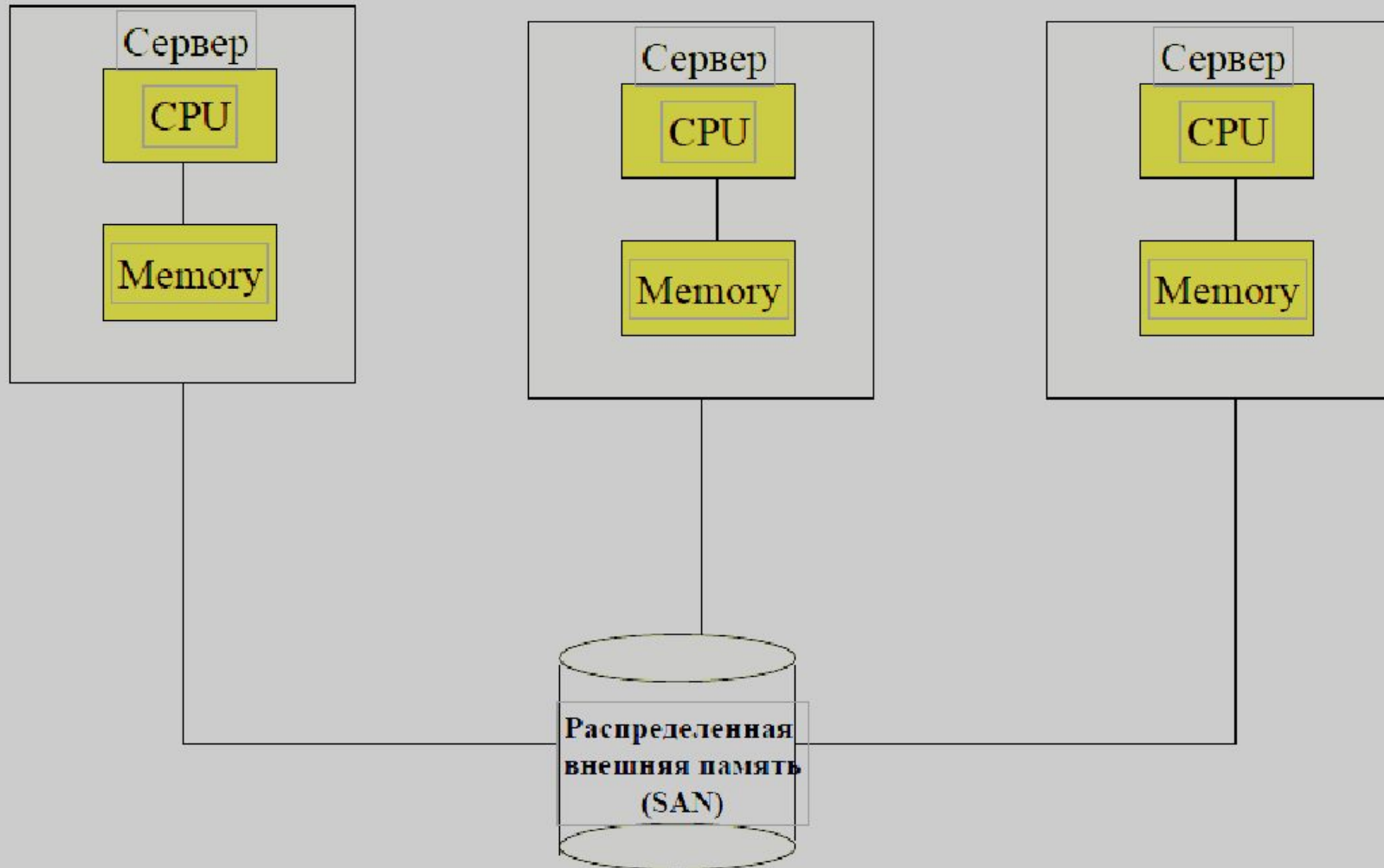


# Распределенные многопроцессорные системы





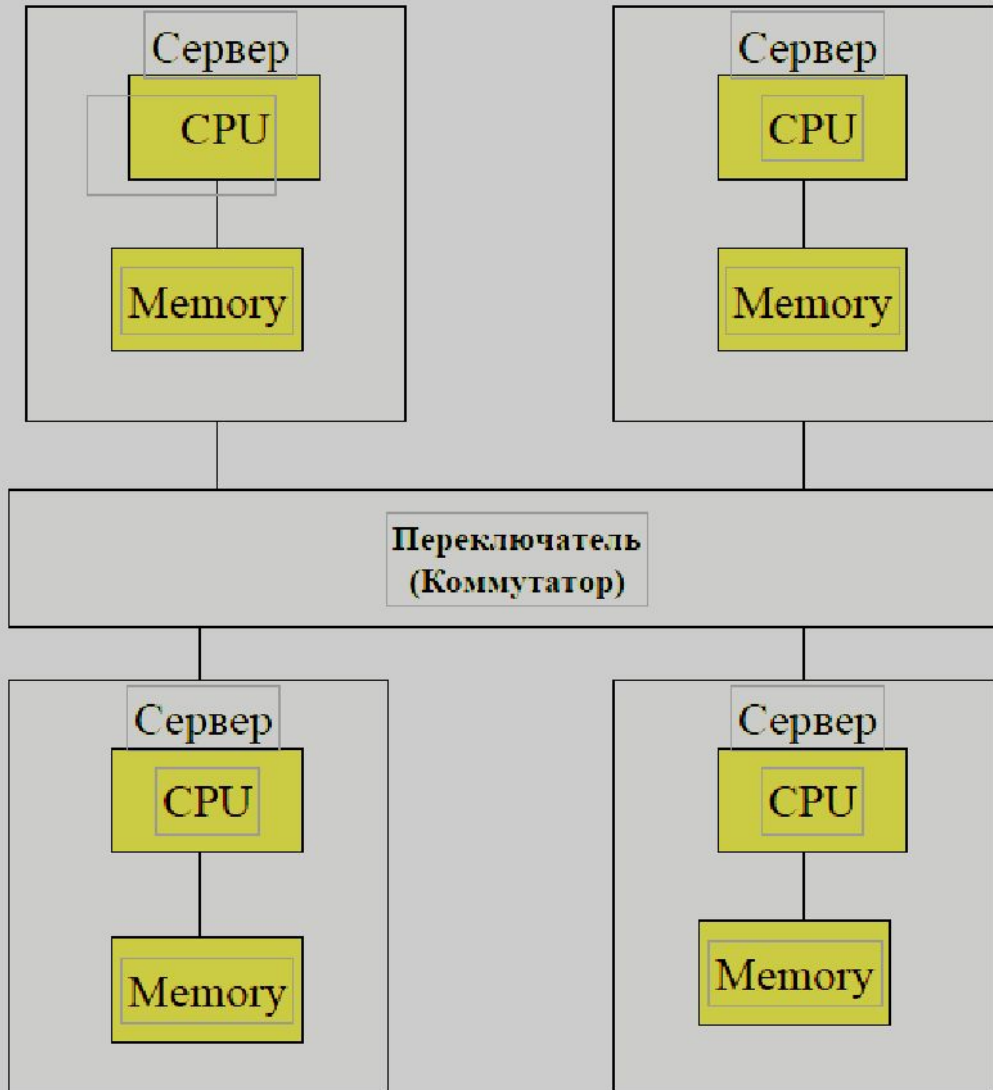
# Разделяемая общая память для нескольких вычислительных систем





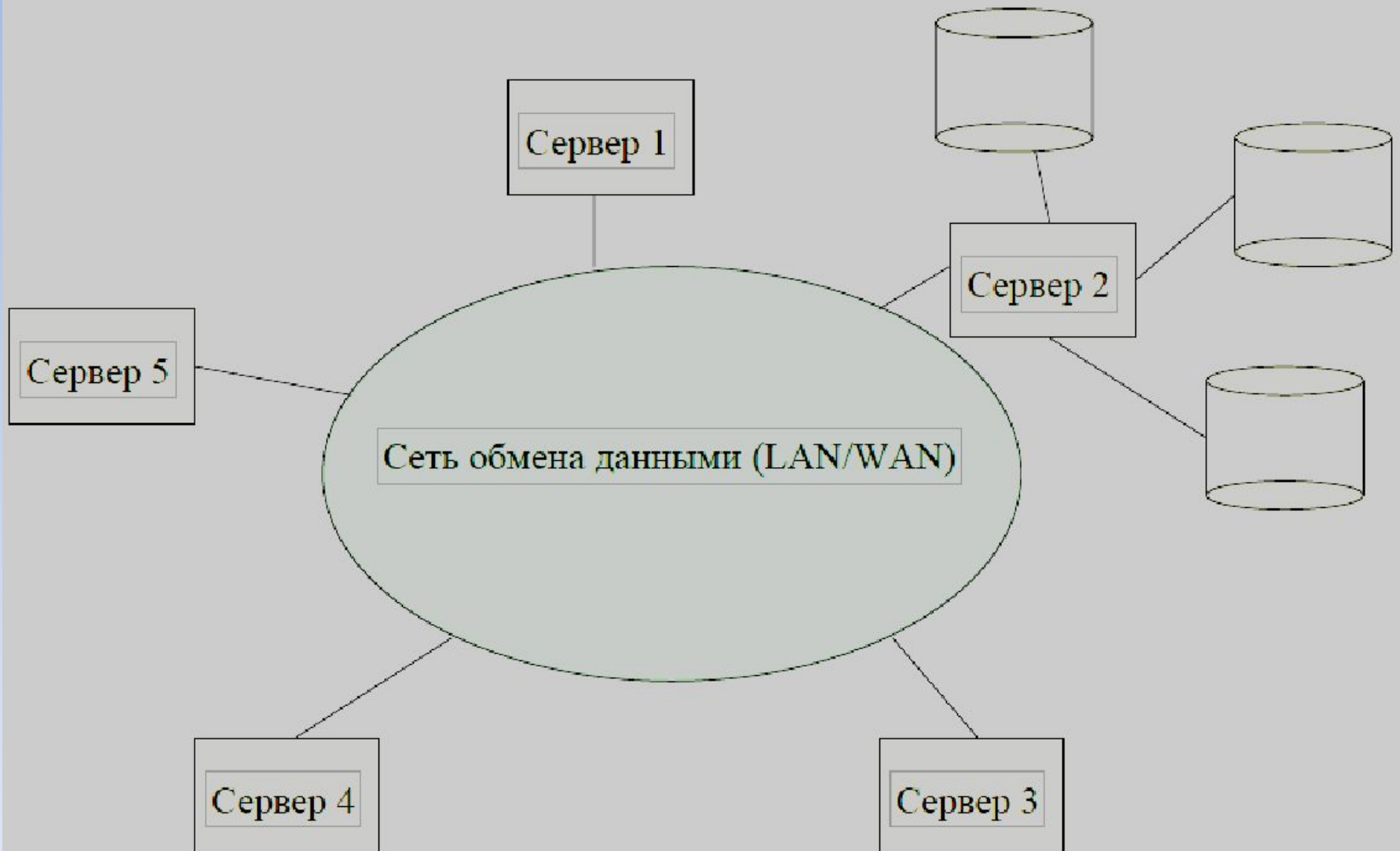


# Многомашинные комплексы



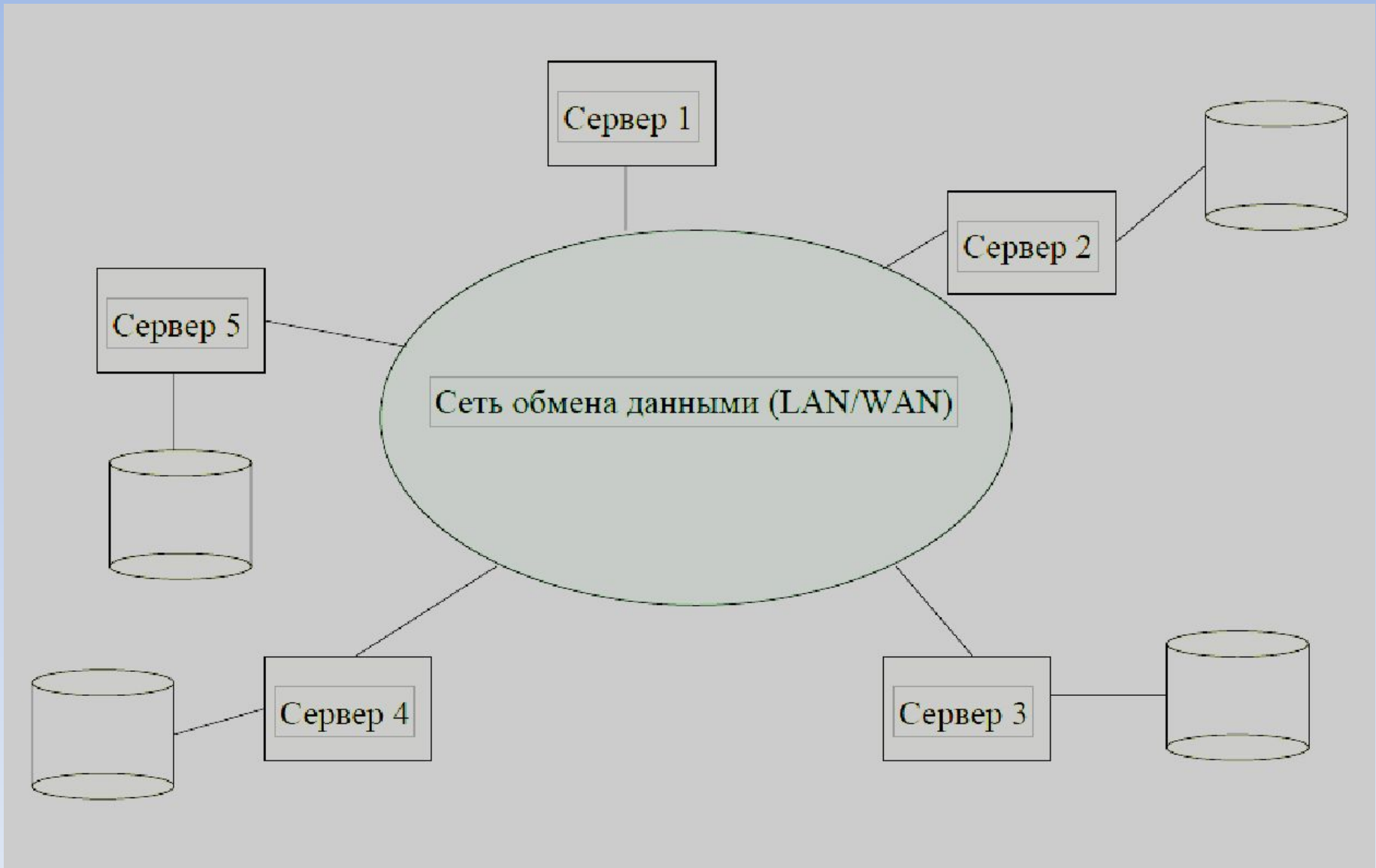


# Централизованные базы данных в сети



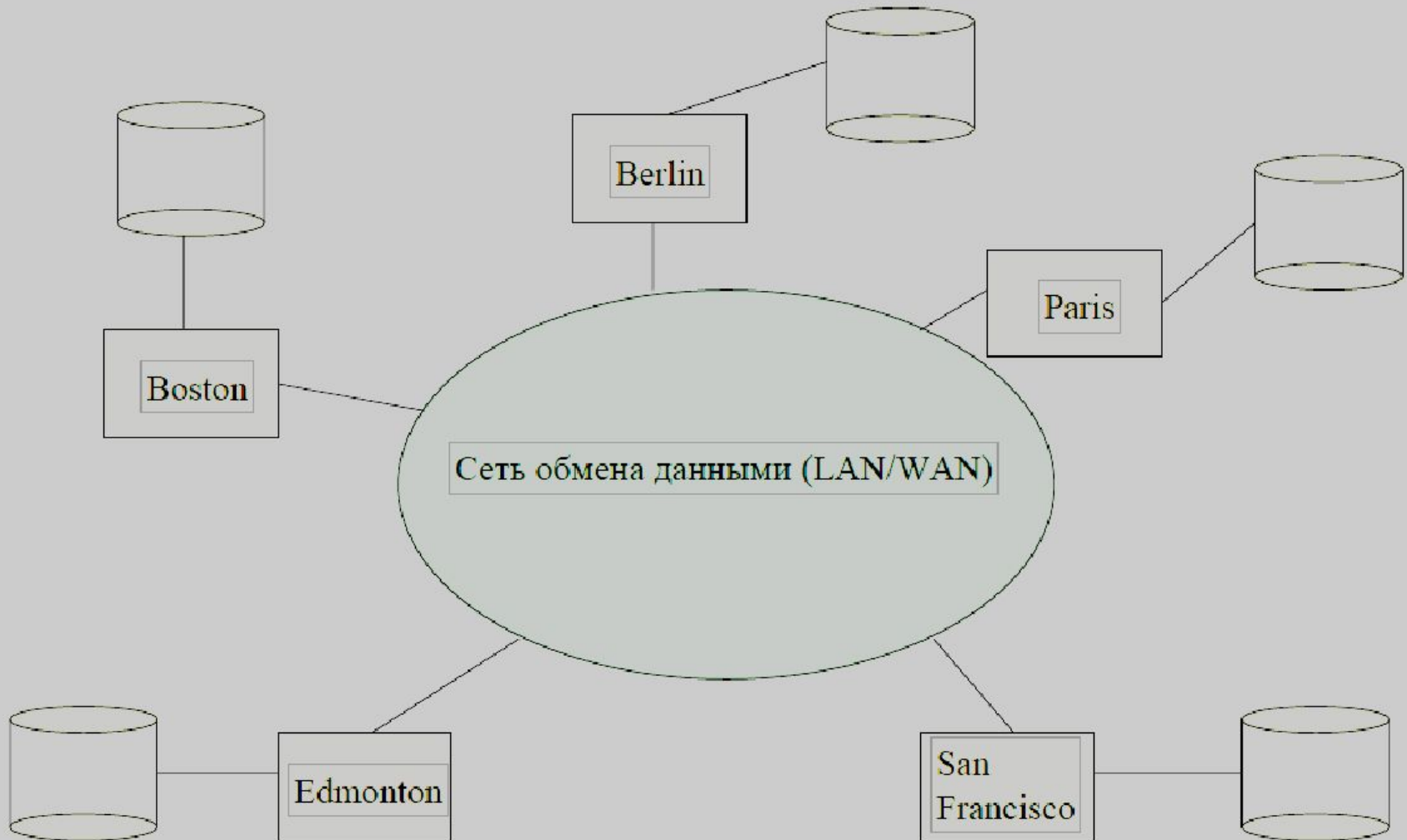


# Распределённые базы в локальной сети





# Распределенные приложения





## Особенности РБД

- Управление распределенными базами данных и репликациями
- Распределенные приложения
- Сетевая независимость (прозрачность)
- Репликационная переносимость (независимость)

Уровни независимости:

- Языковая независимость
- Фрагментационная
- Репликационная
- Сетевая
- Независимость данных



# Факторы сложности РБД

- **Три основных фактора добавляют сложности реализации РБД:**

- Возможное решение на основе репликации данных;
- Технические проблемы с доступностью сайтов (серверов хранения) и каналов связи во время обновления и как следствие необходимость восстановления доступа и данных после устранения проблемы;
- Проблема выгрузки информации на другие сайты РБД путем синхронизации транзакций на множестве сайтов, что существенно труднее, чем в случае централизованной системы

- **Сложность.** Проблемы РБД существенно сложнее, чем у централизованных БД;

- **Стоимость.** РБД требуют дополнительного оборудования и каналов связи, кроме того стоимость растет за счет усложнения ПО. Наиболее сложным и дорогим компонентом является поддержка репликации, в т.ч. производительность оборудования и ПО, и восстановления распределенных данных. При этом возрастает и потребность в дополнительном персонале на удаленных сайтах, однако необходимо анализировать дополнительные затраты на оборудование и персонал в сравнении с достижением эффективности и своевременности доступа к информации;

- **Распределенное управление.** Оно облегчается, если использует соответствующие политики адекватные требованиям к РБД;

- **Безопасность.** Одно из основных преимуществ централизованных БД более высокая степень контроля доступа к данным. Естественно обеспечение



# Проблемные области

- **Разработка распределенных баз данных:**

- не реплицированные (распределение без повторов на различных сайтах фрагментированные);
- реплицированные (полностью дублированные на сайтах) или с частичной репликацией, когда БД хранится более чем на одном сайте, но не на всех с оптимальным распределением фрагментов;
- распределенная обработка запросов:
- разработка алгоритмов анализа запросов и преобразования их в серию операторов манипулирования данными, при этом главная проблема в стратегии выполнения каждого запроса в сети наиболее эффективным способом;
- структура является оптимизированной, если используется параллельная обработка для повышения производительности транзакций.

- **Управление распределенными структурами.**

Словарь РБД содержит информацию об элементах (сущностях) данных их описание и размещение. Словарь может быть глобальным для РБД в целом и локальным для каждого сайта.

- **Управление распределенной обработкой.**

Управление это включает синхронизацию доступа к распределенной БД и поддержание целостности. Это наиболее сложная проблема РБД, т.к. необходимо поддерживать целостность множества копий элементов данных – множественную консистентность.



## Проблемные области

- **Управление распределенными блокировками.**

Механизм синхронизации управления ресурсами (данными) построен на блокировках.

- **Надежность распределенных БД.**

Важно, что механизмом обеспечивающим целостность БД, является алгоритм определения ошибок и восстановления данных на основе механизма двухфазной транзакции, журнализации операций и откатов.

- **Поддержка операционных систем.**

Большинство операционных систем поддерживают часть функций по взаимодействию с СУБД (управление памятью, файловыми системами и методами доступа, восстановления после сбоев и управления процессами), в распределенных системах кроме того добавляются проблемы распределенного в сети ПО.

- **Гетерогенные базы данных.**

Когда БД на разных сайтах не являются гомогенными (однородными) в терминах логических структур (моделей) данных или в терминах механизмов доступа к данным (ЯМД), необходимо обеспечить механизм трансляции операторов между СУБД. Такой механизм обычно включает в себя нормальные формы для обеспечения трансляции данных и шаблоны программ для трансляции операторов ЯМД.





## 10.2 Архитектура распределенных СУБД



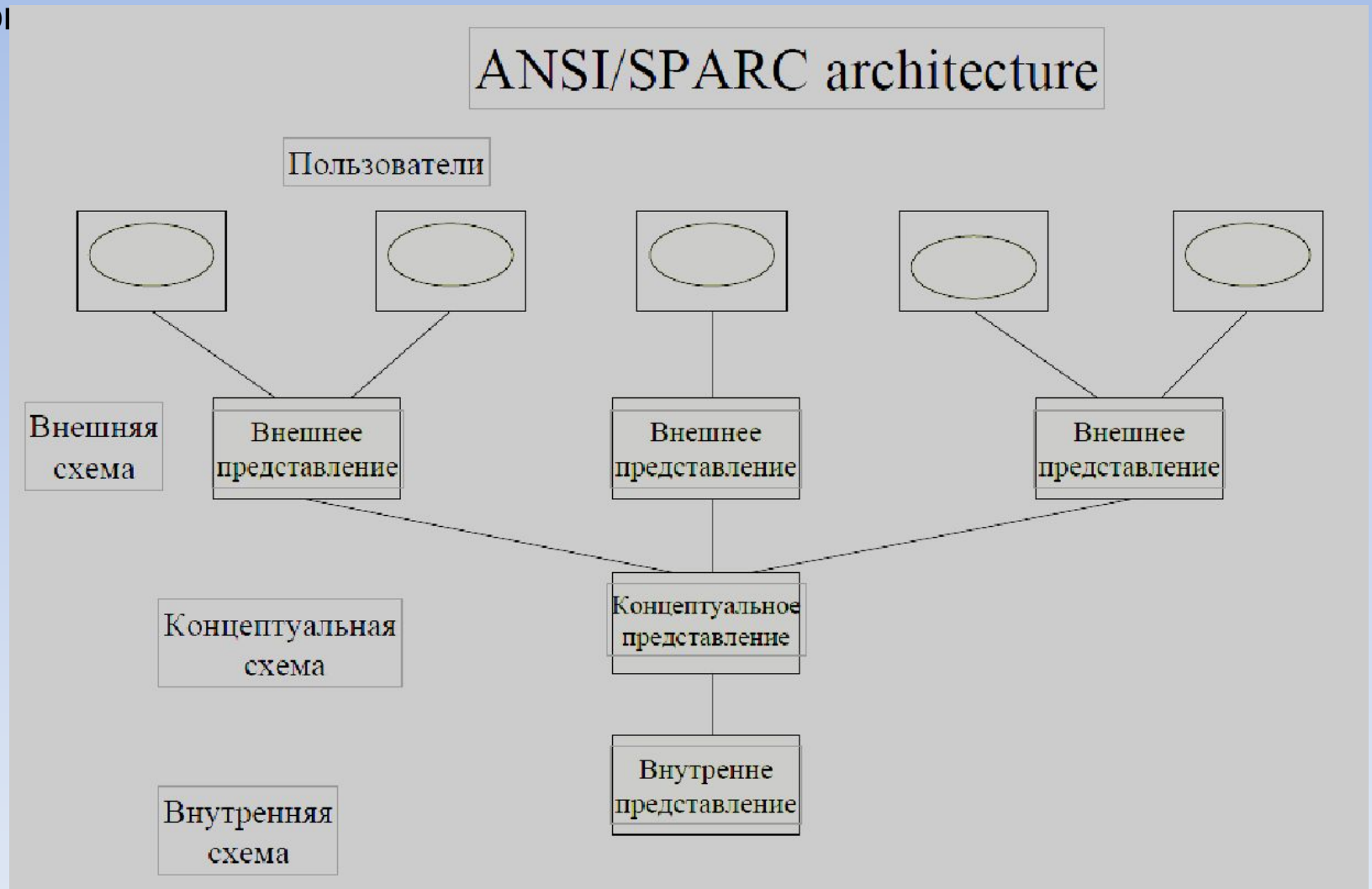
# Архитектура распределенных СУБД

Взаимосвязь между архитектурой систем и референтной моделью этой системы, основывается на трех подходах:

- основывается на компонентах и взаимосвязях между ними;
- основывается на функциях, которые исполняют системы каждого класса, обычно речь идет о иерархической архитектуре систем с хорошо определенными интерфейсами между функциональностями на различных уровнях;
- основывается на данных, определяются различные типы данных и определяются функциональные блоки, которые используют данные согласно различным представлениям (даталогический подход). Этот подход особенно важен, т.к. основным ресурсом СУБД являются данные.

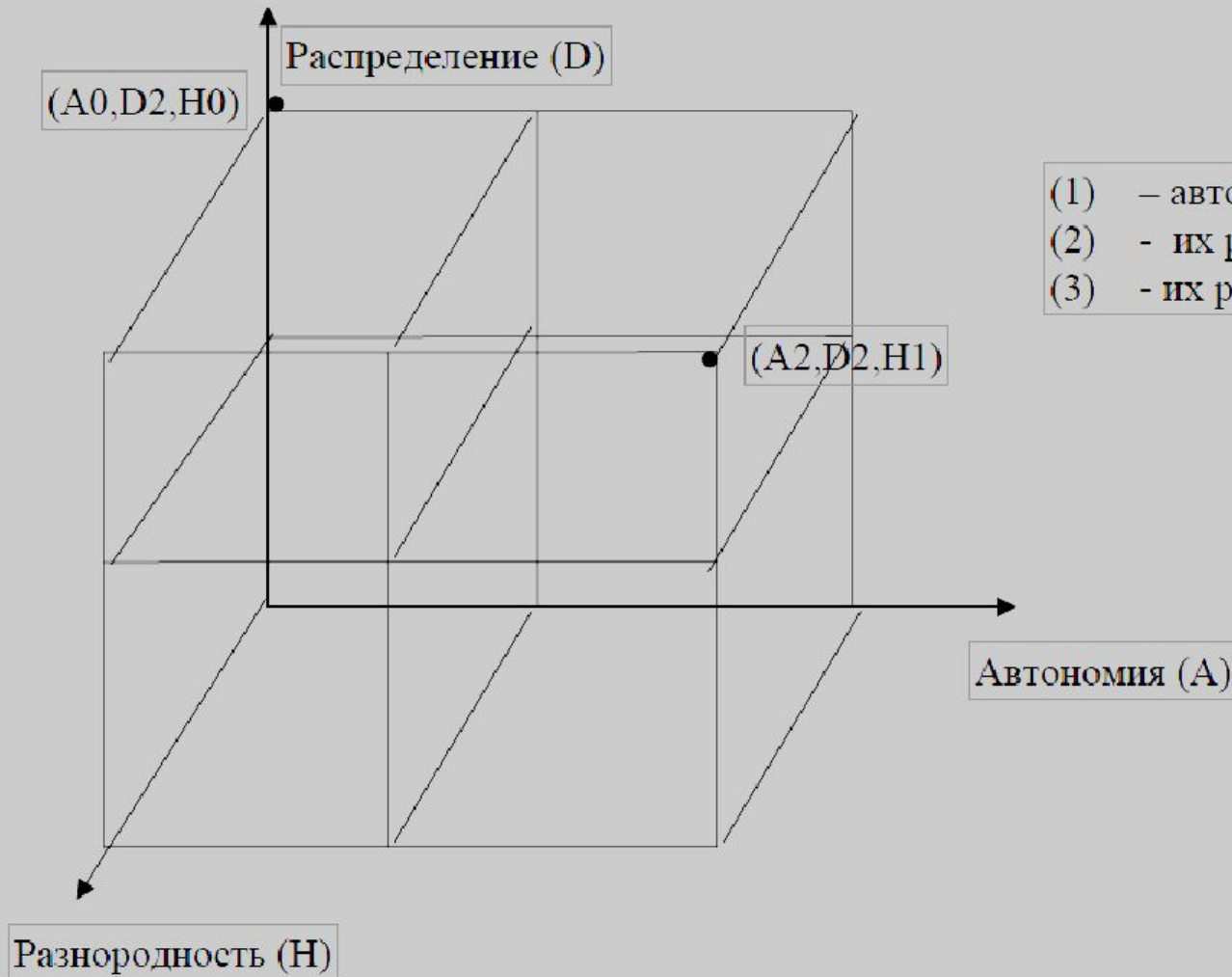


**ANSI/SPARC architecture** – отчет 1978 года, который определил основную архитектуру СУБД и БД в области стандартизации, разработав 43 интерфейса, 14 из которых связаны с физической структурой хранения. Именно тогда были введены понятия: внутренняя, внешняя и КОИ





# Альтернативы внедрения СУБД



- (1) – автономия локальных систем
- (2) - их распределенность
- (3) - их разнородность



**Автономия.** Этот термин относится к управлению распределением, а не к данным. Он показывает степень в рамках которой СУБД может функционировать независимо (обмениваются ли компоненты системы информацией, независимо ли выполняются транзакции, и допустимо ли модифицировать их)

**Распределение.** Этот термин относится к данным. Существует два способа организации распределенного хранения данных: клиент/серверное распределение и полное распределение (по равноправным узлам). Вместе с опцией нераспределенной базы данных они составляют три альтернативных архитектуры.

**Разнородность.** Термин может использоваться в различных формах в распределенных системах, от технической разнородности и различий в сетевых протоколах до различий в методах управления данными. Наиболее существенным является различия в моделях данных, языках запросов и протоколах управления транзакциями. При этом могут быть использованы различные парадигмы доступа к моделям данных, а также различия в языковых средствах при доступе к одинаковым моделям данных.



## Альтернативы архитектуры

Для идентификации архитектуры мы используем нотацию основанную на трех измерениях: **A** (автономия), **D** (распределенность), **H** (разнородность).

Альтернативы по каждой оси нумеруются как 0, 1 или 2.

По оси Автономия 0 представляет полное интеграцию, 1 представляет полуавтономную систему и 2 представляет полную изоляцию.

По оси Распределение 0 - не распределенная система, 1 – клиент/серверная система, 2 полное сетевое распределение.

И по оси Разнородность 0 – однородная система, 1 – устанавливается для разнородных систем



## Альтернативы архитектуры

- **(A0, D2, H0)** – полностью распределенная однородная СУБД;
- **(A2, D2, H1)** – распределенная в сети, разнородная мультибазовая система;
- **(A0, D0, H0)** – первый класс систем – логически интегрированные (сложные системы) мультибазовые СУБД без распределения и разнородности;
- **(A0, D0, H1)** – Разнородность присутствует с множественным управлением данными, но обеспечивает интегрированное представление пользователей на структур данных (логические структуры данных любые – иерархическая, сетевая и реляционная);
- **(A0, D1, H0)** – БД распределенная, но пользователям обеспечен интегрированное представление о данных. Это альтернатива распределенным представлениям клиент/сервер;
- **(A1, D0, H0)** – федеративные СУБД [Heimbigner and McLeod, 1985] это полуавтономные системы, имеющие самостоятельность в своей работе, но координированы при выполнении пользовательских запросов к мультибазам. Обычно относятся к типу «открытых» СУБД;



- **(A1, D0, H1)** – полностью распределенная неоднородная СУБД, которую можно назвать еще и неоднородная федеративная СУБД. Обычно это реляционная СУБД, которая управляет структурированными данными и СУБД, которая управляет хранимыми образами и видео сервером. Если мы хотим обеспечить интегрированный доступ к данным пользователей, то следует скрыть автономность и разнородность сложных систем и установить общий интерфейс;
  - **(A1, D1, H1)** – распределенная сложная система, размещаемая на различных машинах, это может быть распределенная разнородная федеративная СУБД. При этом мы полагаем, что аспекты распределения в этих системах менее важны, чем автономность и разнородность;
  - **(A2, D0, H0)** – если мы движемся к полной автономии, мы называем такую архитектуру системы мультитазовой (MDBS). Элементы такой системы не имеют никакого взаимодействия и даже не знают как взаимодействовать друг с другом, т.е. без разнородной или распределенной MDBS – внутренне связанное множество автономных БД. Амультитазовые системы управления обеспечивают управление таким собранием автономных баз данных и прозрачность доступа к ним;
  - **(A2, D0, H1)** – Наиболее реалистичная архитектура, при которой строятся приложения которые имеют доступ к данным с множества систем хранения с различными характеристиками, возможно не являющимися СУБД, а только приложениями;
  - **(A2, D1, H1)** и **(A2, D2, H1)** – Подобные архитектуры рассматриваем совместно. Обе архитектуры представляют сложные распределенные мульти базы данных. Возможно основное различие, что в случае распределенной архитектуры клиент/сервер (A2, D1, H1) выполнение предпочтительно делегировать middleware системам трехуровневой архитектуры.





## Архитектура распределенных СУБД

- Клиент/серверные системы;
- Распределенная по равноправным узлам БД;
- Системы мультибазданных.



# Архитектура клиент/сервер





# Клиент/серверные системы

- Клиент – это любой процесс, который запрашивает определенные ресурсы или сервисы от других (серверных) процессов;
- Сервер – это процесс, который предоставляет необходимые сервисы (услуги) другому процессу (клиенту).
- Процессы клиента и сервера могут находиться на одном и том же компьютере или же на разных компьютерах, подключенных к сети;
- Когда процессы клиента и сервера находятся на двух или более независимых компьютерах сети, сервер может предоставлять сервисы для более чем одного клиента;
- Сеть связывает воедино серверы и клиенты, предоставляя им средства связи;
- Если клиент запрашивает данные с сервера БД, то фактически обработка запроса (выбор релевантных записей) осуществляется на сервере БД;
- Возможна распределенная обработка информации на различных по конструктиву вычислительных установках;
- Уровень распределения задач обработки данных – главное отличие клиент/серверных систем от систем с мэйнфреймом;
- Серверы и клиенты находятся в отношении многие-ко-многим;
- Тонкий (слабый) клиент выполняет минимум обработки на стороне клиента;
- Толстый (сильный) клиент берет на себя значительную часть обработки данных;



# Клиент/серверные системы

- Клиент/серверные системы делятся на двухзвенные и трехзвенные;
- Двухзвенные – клиент запрашивает сервисы непосредственно от сервера;
- Трехзвенные – клиентские запросы обрабатываются промежуточными серверами, которые координируют выполнение клиентских запросов с подчиненными им серверами;
- Факторы, влияющие на клиент/серверную архитектуру:
  - Изменения в инфраструктуре бизнеса;
  - Возросшие требования доступа к данным предприятия;
  - Необходимость повышения производительности конечных пользователей на базе эффективного использования информации;
  - Развитие технологий, обеспечивающих эффективность использования клиент/серверных моделей.



## Ожидания специалистов ИТ от клиент/серверных технологий:

- сокращение стоимости разработки и реализации;
- уменьшение времени на разработку и повышение производительности труда разработчиков;
- расширение жизненного цикла системы за счет масштабируемости и переносимости;
- уменьшение стоимости эксплуатации системы;
- передача части функций от разработчиков конечным пользователям;
- улучшение размещения информации.



## Ожидания бизнеса от клиент/серверных технологий:

- гибкость и адаптивность;
- повышение производительности труда сотрудников;
- оптимизация бизнес-процессов компании и их мобильности;
- повышение качества обслуживания клиентов.



# Клиент/серверные системы

- **Клиент** – любой процесс компьютера, который запрашивает сервис от сервера. Клиент называется интерфейсным приложением, что отражает факт взаимодействия конечного пользователя с клиентским процессом;
- **Сервер** – любой компьютерный процесс, представляющий сервис клиентам. Сервер называется серверным приложением, что отражает факт предоставления сервером сервиса клиентскому процессу;
- **Промежуточное ПО** передачи/обмена данными – любой компьютерный процесс, посредством которого клиенты и серверы взаимодействуют друг с другом (уровень коммуникаций, позволяющий передавать данные и управляющую информацию между клиентами и серверами) (middleware).



# Правила архитектуры «клиент/сервер»

- Независимость от оборудования;
- Независимость от ПО:
  - операционной системы;
  - сетевой системы;
  - приложений.
- Открытый доступ к сервисам.
- Распределение процессов:
  - автономность процессов (с определенными границами и функциями);
  - максимальное использование локальных ресурсов;
  - масштабируемость и гибкость;
  - способность к взаимодействию и интегрируемость.
- Стандартизация. Все правила клиент/серверной архитектуры основаны на стандартах (интерфейсы, сетевые протоколы, обмен между процессами)





# Компоненты клиента и сервера

## Компоненты клиента:

- Мощное оборудование;
- Операционная система с возможностью много задачной обработки информации;
- Графический интерфейс пользователя (GUI);
- Коммуникационные возможности.

## Компоненты сервера и обеспечиваемы им возможности:

- файловые серверы для ЛВС, в которых хост с быстрыми дисками большой емкости совместно используется несколькими пользователями;
- сервисы печати для локальной сети (принт-сервер);
- сервисы факсимильной связи (факс-сервер);
- сервисы передачи данных;
- сервисы баз данных;
- сервисы транзакций (сервера транзакций, подключенные к серверу БД);
- другие сервисы и т.д.



# Характеристики серверного оборудования

- Быстрый процессор;
- Высокая отказоустойчивость (двойное питание, резервный ИБП, обнаружение и исправлении ошибок, массив резервных дисков);
- Возможность модернизации (ЦП, памяти, диска и изменения периферии);
- Поддержка шины для подключения дополнительного оборудования;
- Различные коммуникационные возможности, обеспечивающие распределенную обработку;
- Независимость от местоположения процесса сервера в сети;
- Оптимизация ресурсов;
- Масштабируемость;
- Интеграция и способность к взаимодействию (plug-and-play).



# Компоненты ППО базы данных

- Программный интерфейс приложения (API). Открыт для клиентского приложения. Программист взаимодействует с ППО через API, поставляемый вместе с ППО;
- Транслятор базы данных. ТБД транслирует SQL- запросы;
- Сетевой транслятор. Управляет сетевыми межкоммутационными протоколами.
- Клиенту предоставляются следующие преимущества ППО:
  - Доступ к нескольким БД;
  - Независимость от сервера БД;
  - Независимость от сетевого протокола.



# Функциональные логические компоненты приложения



1. Логика представления:  
форматирование GUI
2. Логика обработки  
ввода/вывода:  
проверка ввода/вывода  
и выявление ошибок
3. Бизнес-логика
4. Логика управления  
данными
5. Логика манипулирования  
БД: хранение и извлечение



# Типичное размещение сервисов

- *логика представления* всегда размещается на стороне клиента, т.к. предполагает взаимодействие с конечным пользователем;
- *логика обработки ввода/вывода* может размещаться на стороне клиента или на стороне сервера («сильный сервер/слабый клиент») или на прикладном сервере;
- *бизнес-логика* может размещаться и на клиенте, и на сервере, но чаще на стороне клиента. Если используется трехзвенная клиент/серверная система, то промежуточные сервера обычно содержат все элементы бизнес-логики;
- *логика управления данными* может размещаться и на клиенте и на сервере. Обычно размещается на стороне клиента или на промежуточном сервере приложения с бизнес-логикой. Логика управления данными может подразделяться на клиентские и серверные подкомпоненты, выполняемые ППО базы данных, а в случае РБД, подкомпоненты могут размещаться на нескольких серверных компьютерах;
- *логика манипулирования данными*, как правило, размещается на стороне сервера, но она (логика манипулирования данными) может распределяться между несколькими серверами в среде распределенных баз данных.



# Проблемы реализации клиент/серверных систем

- От частных систем к открытым системам (системы должны объединяться и быть открыты для других систем);
- От обслуживания кодирования к анализу, проектированию и сервису;
- От локализации данных к распределению данных;
- От централизованного стиля к распределенной форме управления данными;
- От вертикального негибкого стиля к более горизонтальному, гибкому организационному стилю;



**Спасибо за внимание!**