

# Проектирование и эксплуатация информационных систем в медиаиндустрии

*Выломова Екатерина Алексеевна*  
*e-mail: [evylomova@gmail.com](mailto:evylomova@gmail.com)*

# 0. Лекция 1

Базовые понятия:

- Кибернетика
- Формы адекватности
- Мера информации, информация, энтропия, качество информации
- Информационная модель, технология, система. Классификация ИС

# 0. Принцип KISS

Keep it short and simple

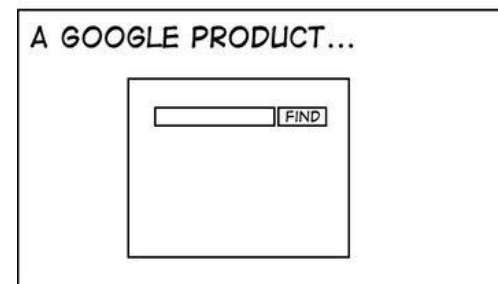
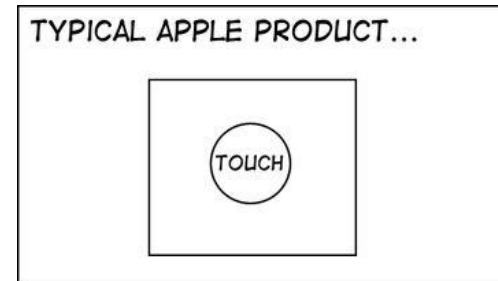
*or*

Keep it simple, stupid

Einstein: “everything should be made as simple as possible, but no simpler”

Главное – простота

*Пример: Unix*



YOUR COMPANY'S APP...

FIRST NAME: [ ] TYPE CD: [ ]  
LAST NAME: [ ] TQP STAT: [ ]  
SSN: [ ] FT/PT: [ ] VER: [ ]  
ID: [ ] CAT CD: [ ]  
PHONE 1: [ ] CITY: [ ]  
PHONE 2: [ ] STATE: [ ]  
ADDR 1: [ ] ZIP: [ ]  
ACCT #: [ ] ORD #: [ ]

4 - K  
AA2-  
DK9B  
KKA?  
CN3  
AA-9  
NEW  
DEL

OKAY APPLY SAVE UNDO HELP DELETE EDIT  
SELECT BROWSE ERRORS

STUFFTHATHAPPENS.COM BY ERIC BURKE

# Лекция 2. Архитектура ИС

- Понятие архитектуры ИС
- Типы архитектур ИС
- Примеры архитектур и принципов работы ИС

# I. Архитектура ИС

**Архитектура ИС** – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.



Слой представления

- взаимодействие с пользователем

Бизнес-логика

- правила обработки данных

Слой доступа к данным

- хранение, выборка, модификация и удаление данных

**Архитектура ИС**

# I. Классификация архитектур

## По степени распределенности:

- Настольные(desktop) – все данные (БД, СУБД, клиентские приложения) хранятся на одном компьютере
- Распределенные (distributed) – компоненты распределены по нескольким компьютерам

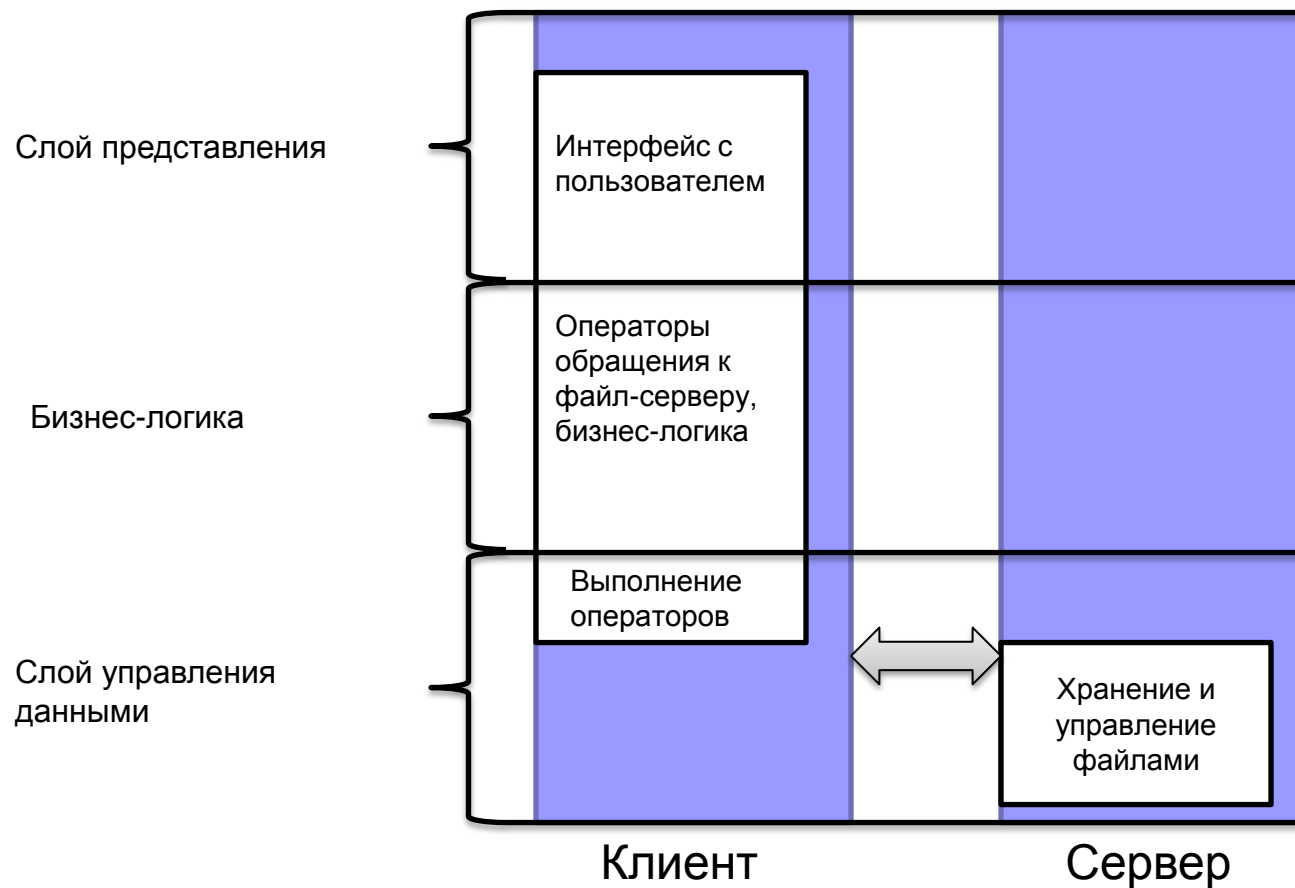
## Распределенные подразделяются на:

- Файл-серверные
- Клиент-серверные

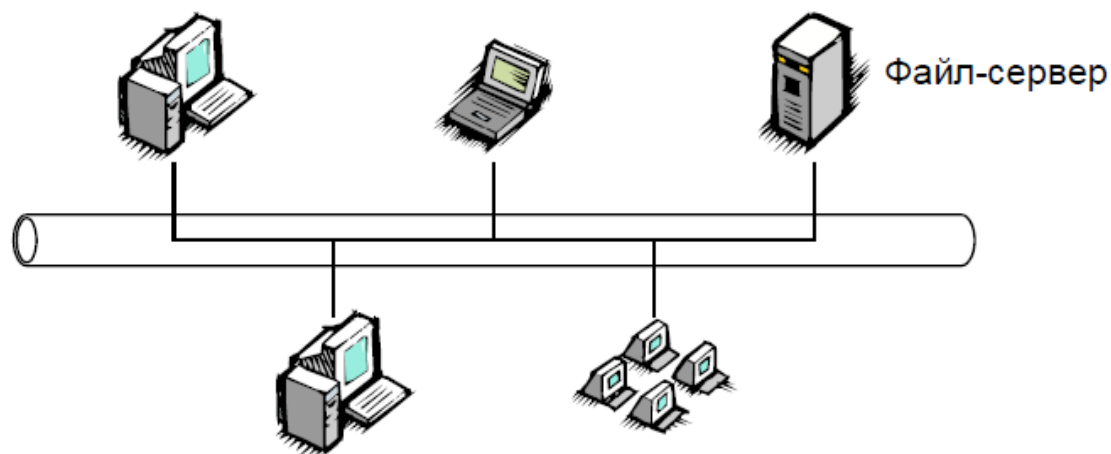
## Клиент-серверные:

- Двузвенные
- Многозвенные

# I. Файл-серверная архитектура



# I. Файл-серверная архитектура



**СУБД, поддерживающие модель:**

- FoxPro
- MS Access
- Paradox
- dBase

## **Плюсы:**

- Многопользовательский режим работы с данными
- Централизованное управление доступом
- Низкая стоимость и высокая скорость разработки

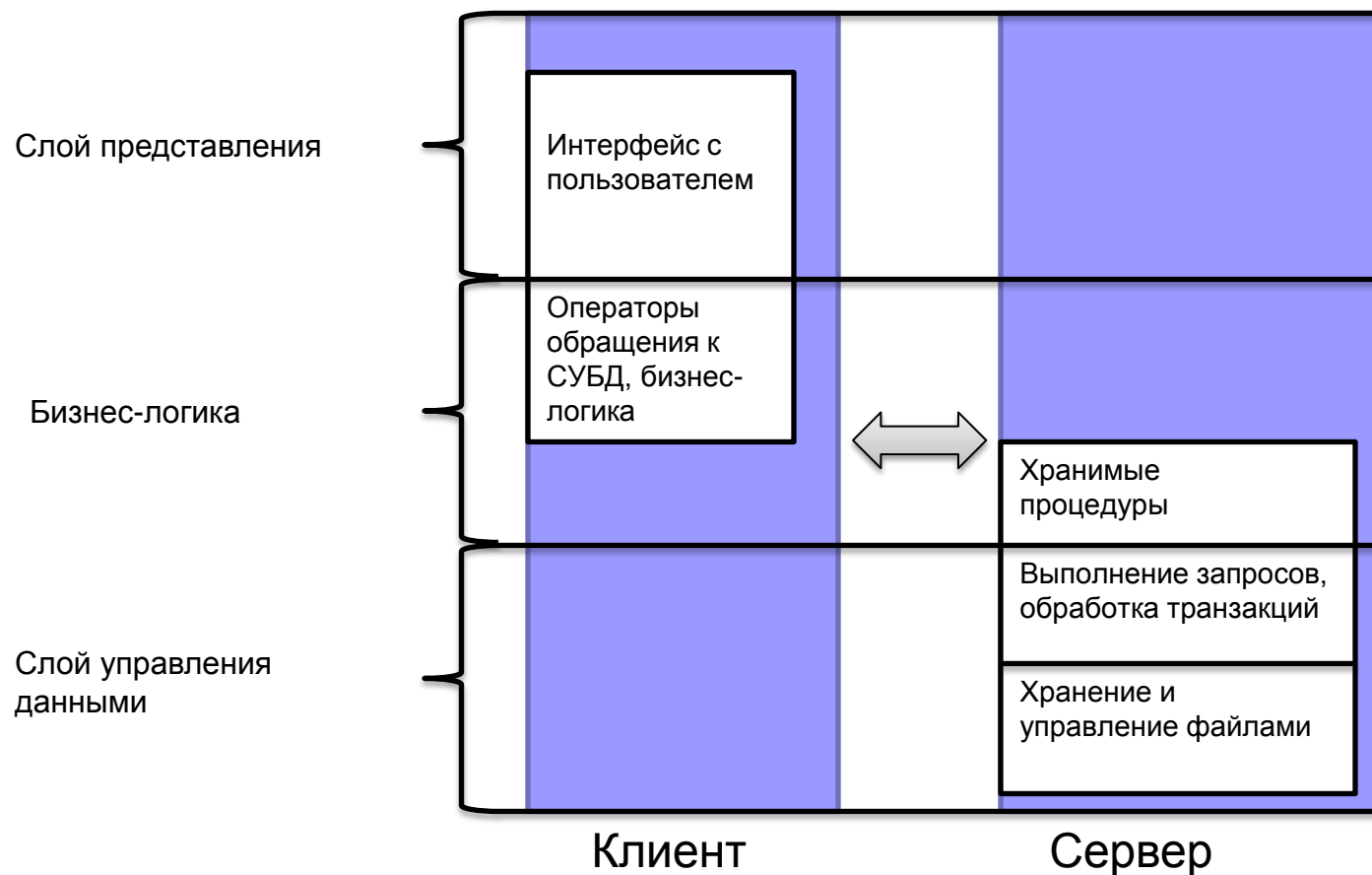
## **Минусы:**

- Низкая производительность; сильная перегрузка ЛВС
- Низкая надежность
- Слабая возможность расширения

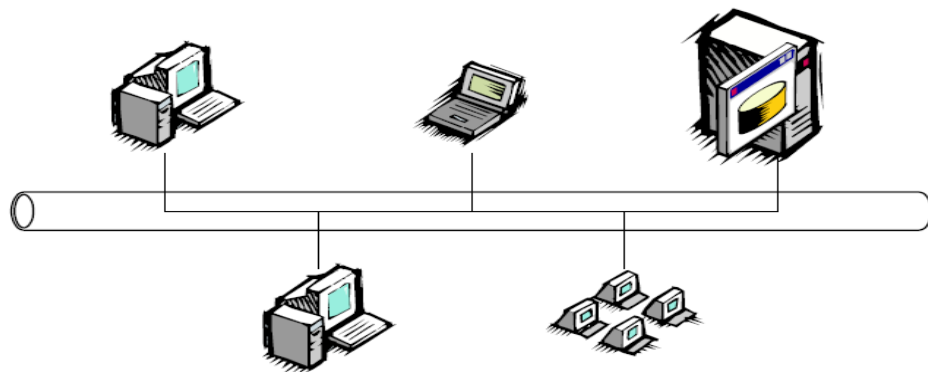
**Архитектуры ИС. Файл-сервер**



# I. Клиент-серверная архитектура с сервером СУБД



# I. Клиент-серверная архитектура с сервером СУБД



**СУБД, поддерживающие модель:**

- Oracle
- MS SQL Server
- SyBase
- Informix
- Centura
- Interbase

**Плюсы:**

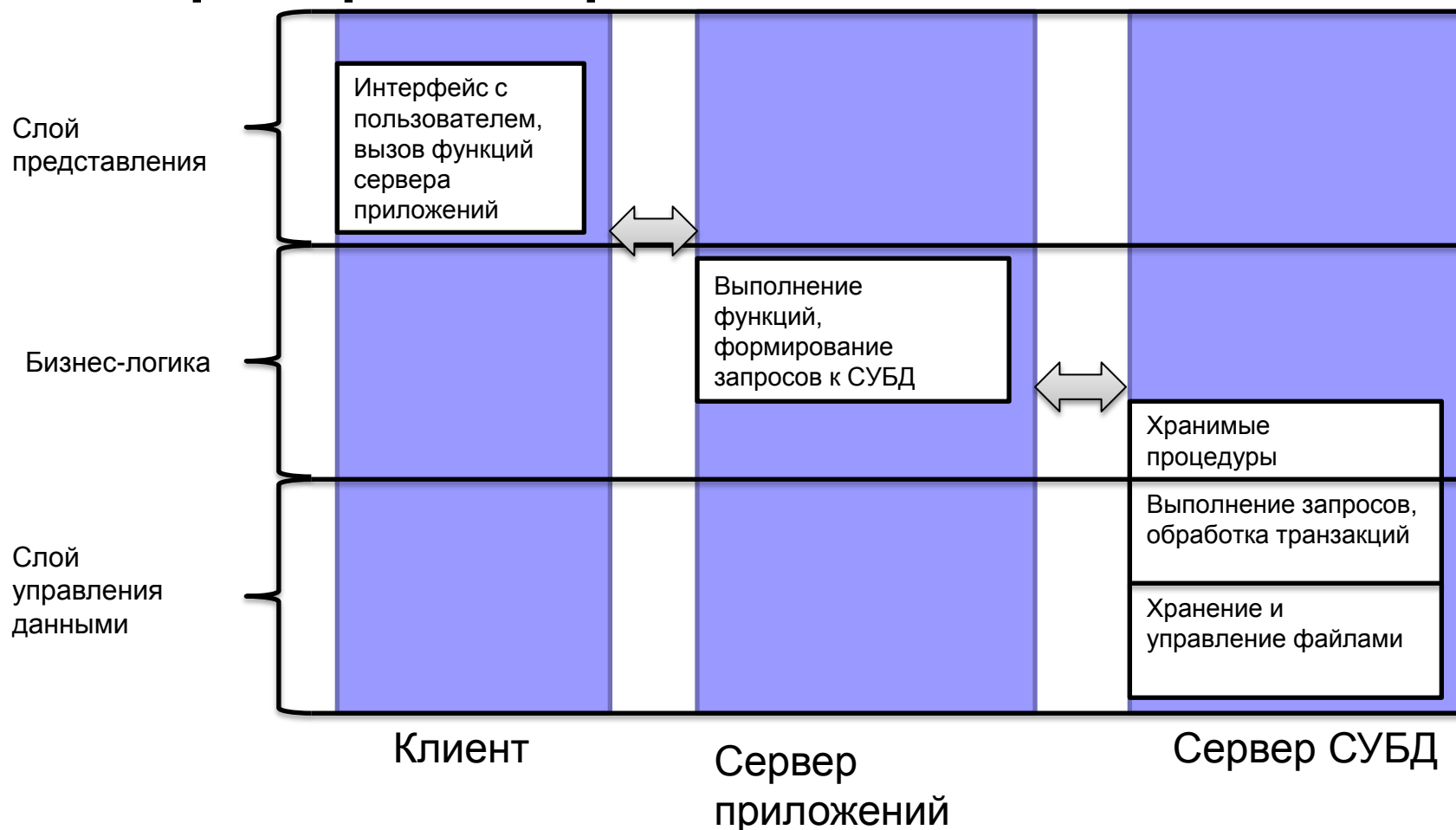
- Многопользовательский режим работы
- Гарантия целостности данных

**Минусы:**

- Бизнес-логика на клиенте, сложности изменения алгоритмов
- Слабая защита данных от взлома
- Высокие требования к пропускной способности, клиентским машинам
- Высокая сложность администрирования и разработки

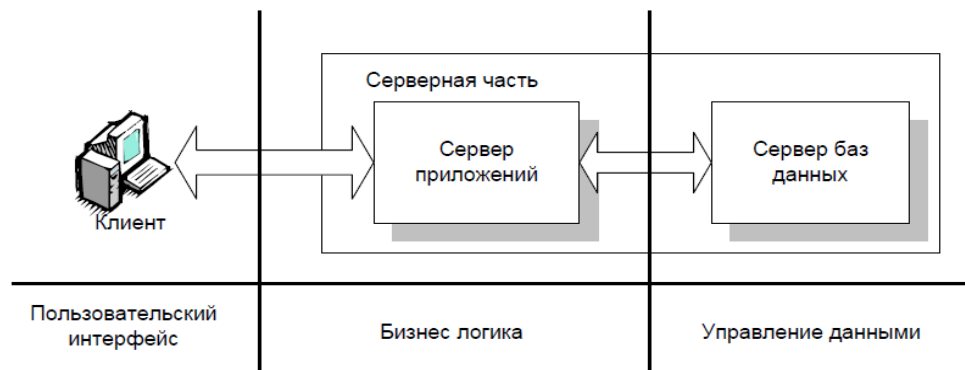
**Архитектуры ИС. Клиент-сервер с сервером СУБД**

# I. Клиент-серверная архитектура с сервером приложений



**Архитектуры ИС. Клиент-сервер с сервером приложений**

# I. Клиент-серверная архитектура с сервером приложений



**СУБД, поддерживающие модель:**

- MS SQL Server
- CICS

## **Плюсы:**

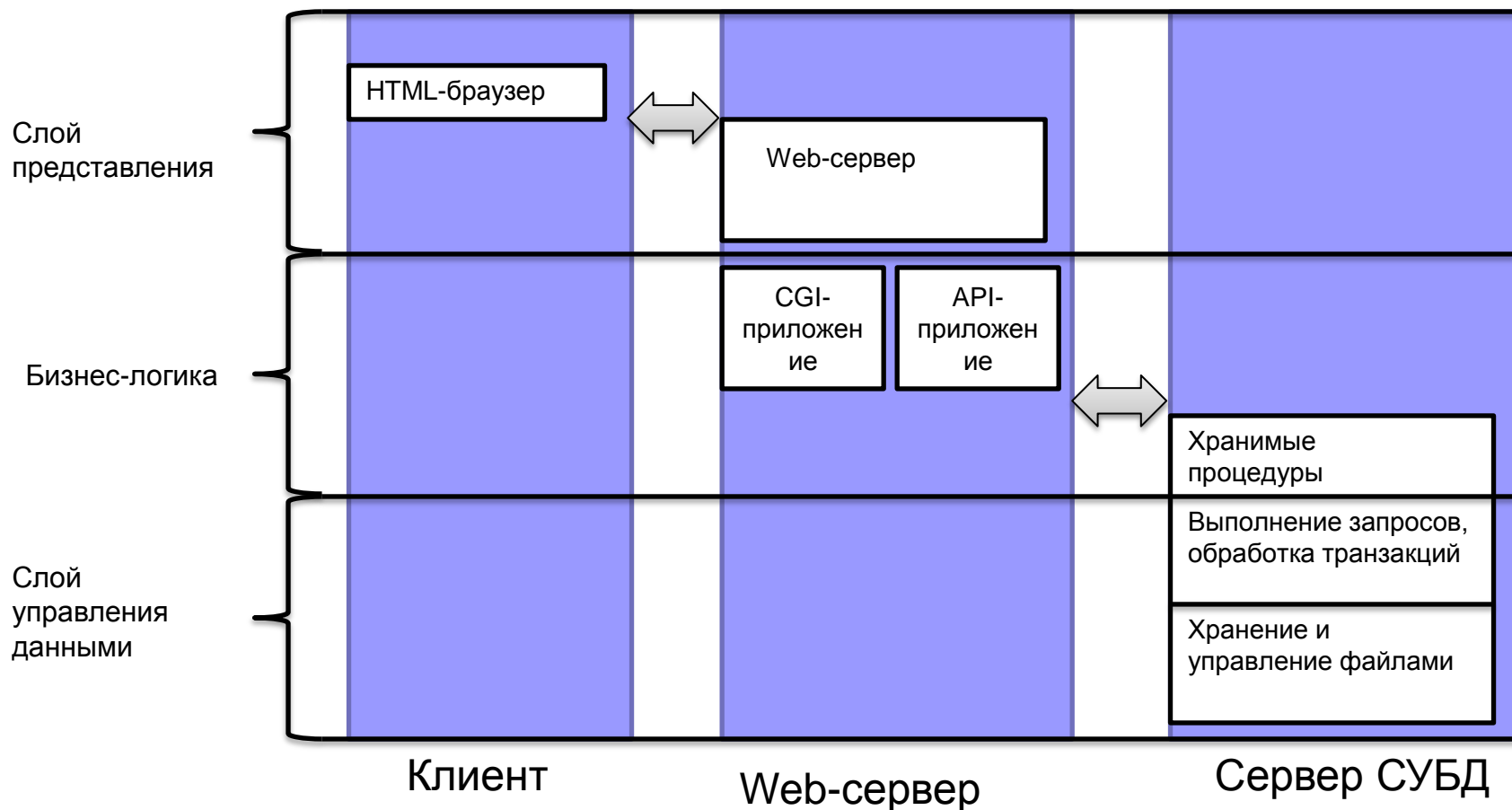
- Тонкий клиент
- Между клиентом и сервером передается минимум данных: аргументы функций и результаты
- Сервер приложения может быть запущен в 1 или М экземплярах на 1 или М компьютерах
- Дешевый трафик между СП и сервером СУБД, снижение нагрузки на сервер данных; дешевле наращивание функциональности и обновление ПО

## **Минусы:**

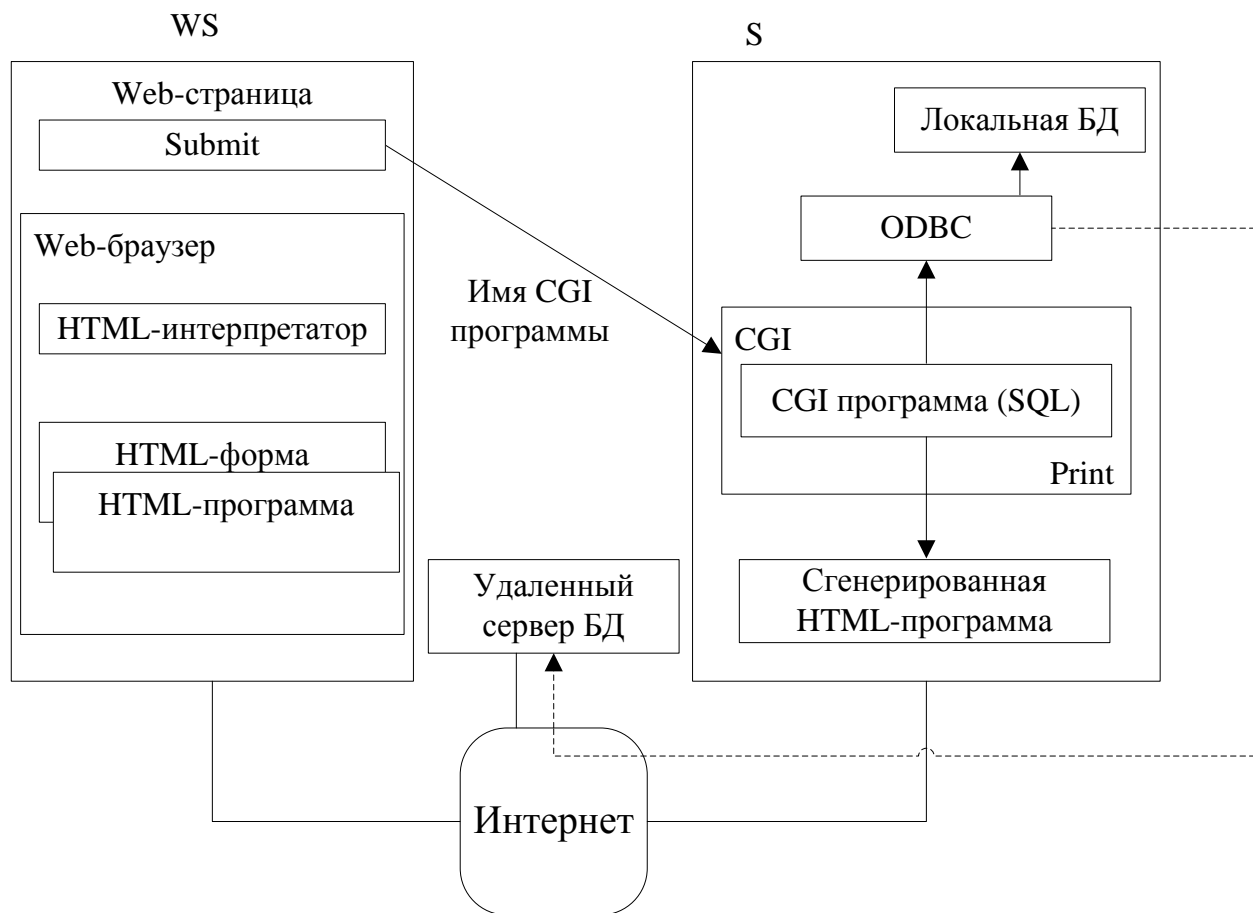
- Высокие расходы на администрирование и разработку серверной части

**Архитектуры ИС. Клиент-сервер с сервером приложений**

# I. Архитектура на основе Internet/Intranet и CGI/API



# I. Архитектура на основе Internet/Intranet и CGI/API



**Архитектуры ИС. Клиент-сервер с сервером приложений**

# I. CGI vs API

**CGI** (от англ. **Common Gateway Interface** — «общий интерфейс шлюза») — стандарт интерфейса, используемого для связи внешней программы с веб-сервером.

## **Плюсы:**

- Web-сервер выступает в качестве сервера приложения (администрирование выполняется централизованно).
- CGI интерфейс унифицирован и реализован во всех серверах.
- Для доступа к БД можно использовать любой web-браузер.

## **Минусы:**

- Каждая CGI программа выполняется как процесс ОС. Занимает много времени.
- CGI программа не поддерживает контекст связи с БД, т.е. БД открывается при каждом вызове CGI программы.
- Генерируемая форма имеет небольшие выразительные возможности.

# I. CGI vs API

**API** – (от англ. **Application programming interface** — «интерфейс программирования приложений») — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах

## **Плюсы:**

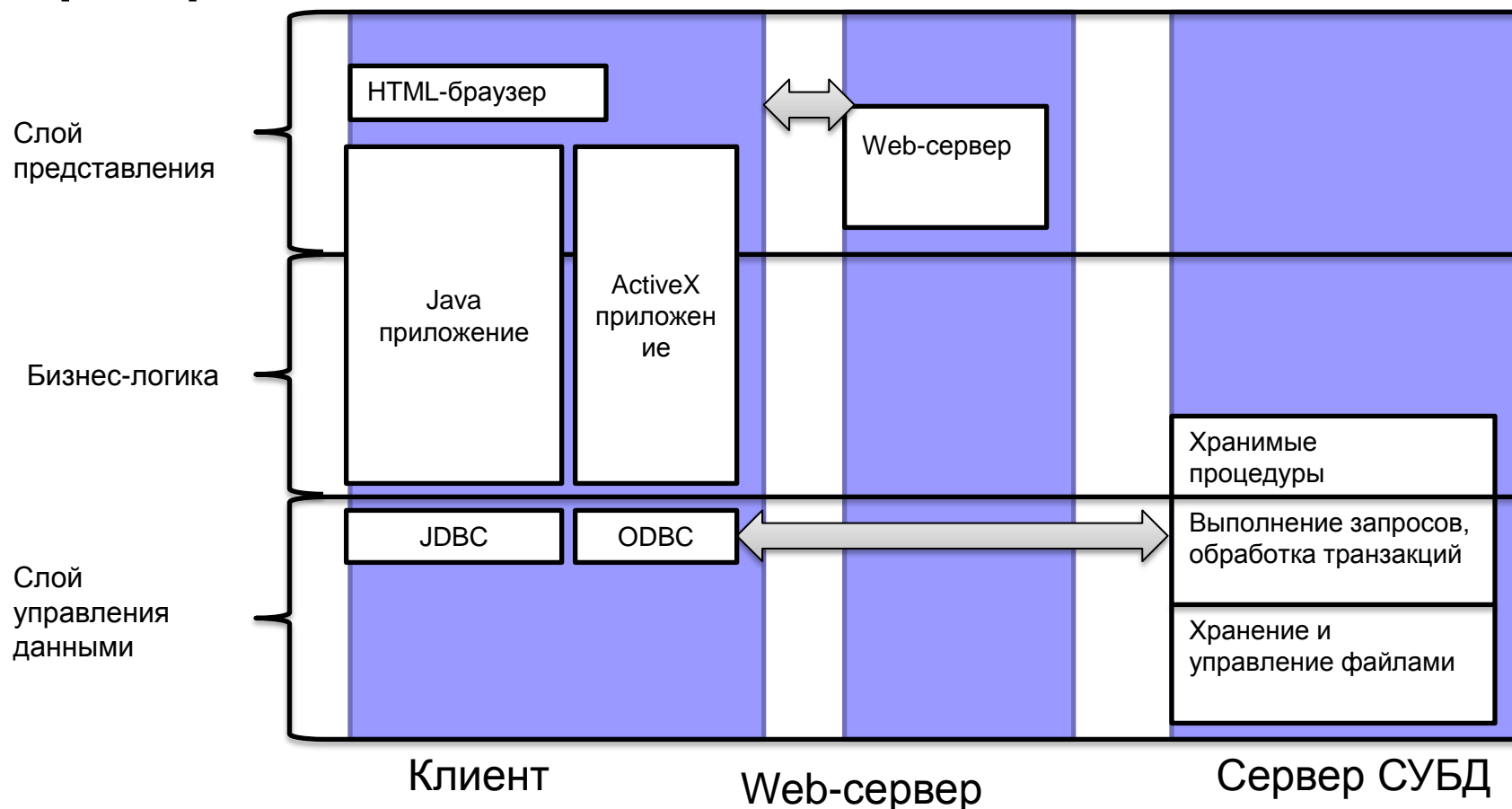
- Они выполняются быстрее, чем CGI программы (нет переключения между задачами ОС).
- ASP вместе с некоторыми дополнениями (Remote scripting, scriptlet) позволяют поддерживать контекст с БД

## **Минусы:**

- API программы разных производителей не совместимы между собой
- API интерфейсы и соответствующие API программы зависят от платформы

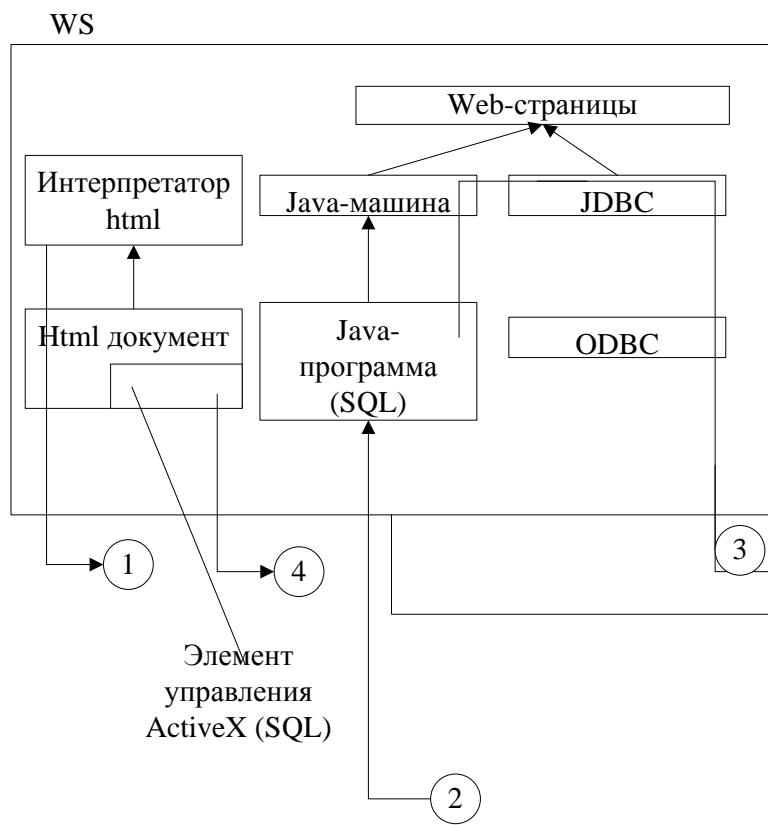


# I. Архитектура на основе Internet/Intranet с мигрирующими программами



**Архитектуры ИС. Архитектура на основе Internet/Intranet**

# I. Архитектура на основе Internet/Intranet и мигрирующих программ



1 – тег <applet>->браузер обращается к web-серверу с запросом на чтение java-программы

2 - java-программа передается на WS и там интерпретируется java-машиной

3 - в процессе выполнения java-программы встречается SQL-оператор->

через интерфейс JDBC-ODBC передается на удаленный сервер БД ->выполняется

Результаты возвращаются обратно в java-программу, там обрабатываются java-программа выводит их в web-страницу.

# I. Архитектура на основе Internet/Intranet и мигрирующих программ

## Плюсы:

- Эта технология позволяет существенно разгрузить web-сервер, т.к. java-апплеты выполняются на рабочих станциях
- Java-апплеты мобильны. Язык java достаточно гибкий для создания сложных программ
- JDBC является универсальным интерфейсом. Язык SQL не зависит от СУБД
- Существует множество java-программ, которые можно использовать. Их можно запускать с различных серверов и связывать на рабочей станции

## Минусы:

- Размеры java-апплетов должны быть небольшими. Это связано с ограничением времени передачи по сети
- Низкая производительность java-программ
- Относительная сложность разработки java-апплетов, выполняющих доступ к БД

# I. Распределенные ИС

Особенности распределенных систем с точки зрения проектировщика

- Ссылки
- Задержки выполнения запросов
- Активация/деактивация
- Постоянное хранение
- Параллельное исполнение
- Отказы
- Безопасность

## 1. Ссылки

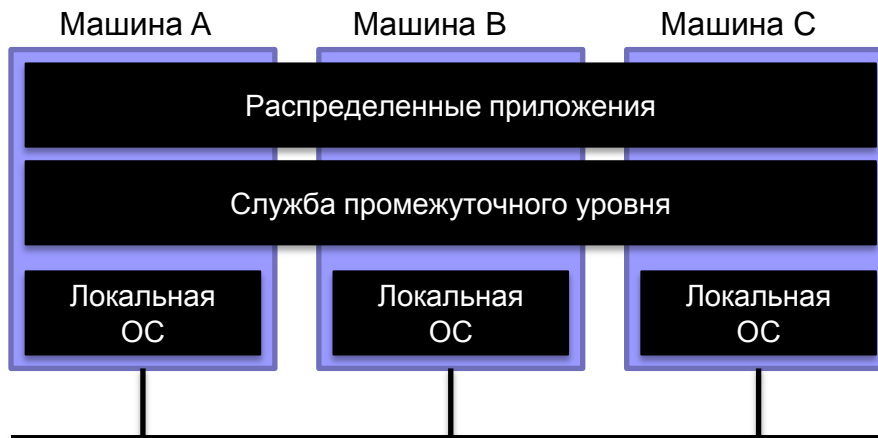
- Содержат информацию о размещении
- Содержат информацию о безопасности
- Содержат ссылки на объектные типы

## 2. Задержки выполнения запросов

- Снизить частоту обращения
- Укрупнить выполняемые функции

## 3. Активация/деактивация

- Большое число объектов
- Объекты могут не использоваться на протяжении долгого времени
- Объект переносится в память при активации
- Объект удаляется из памяти при деактивации



# I. Распределенные ИС

## 4. Постоянное хранение

- Объект может иметь или не иметь состояния
- Имеющие состояние объекты должны храниться между деактивацией и активацией
- Способы хранения:
  - Запись в файловую систему
  - Отображение в реляционную СУБД
  - Сохранение в объектную СУБД

## 5. Параллельное исполнение

- В нераспределенных системах выполнение последовательное либо конкурентное в разных потоках процесса.
- В распределенных системах выполнение всегда параллельное -> сложные схемы синхронизации

## 6. Отказы

- Большая вероятность отказов
- Клиент обязан проверять выполнение запроса сервером

## 7. Безопасность

- Кто запрашивает выполнение операции?
- Как можно удостовериться в личности субъекта?
- Как определить предоставлять ли субъекту сервис?
- Как можно доказать, что сервис был предоставлен?

## II. Примеры

Google™

Яндекс

facebook®

## II. Поисковые системы

**Поисковая система** – программно-аппаратный комплекс с веб-интерфейсом, предоставляющий возможность поиска информации в интернете.

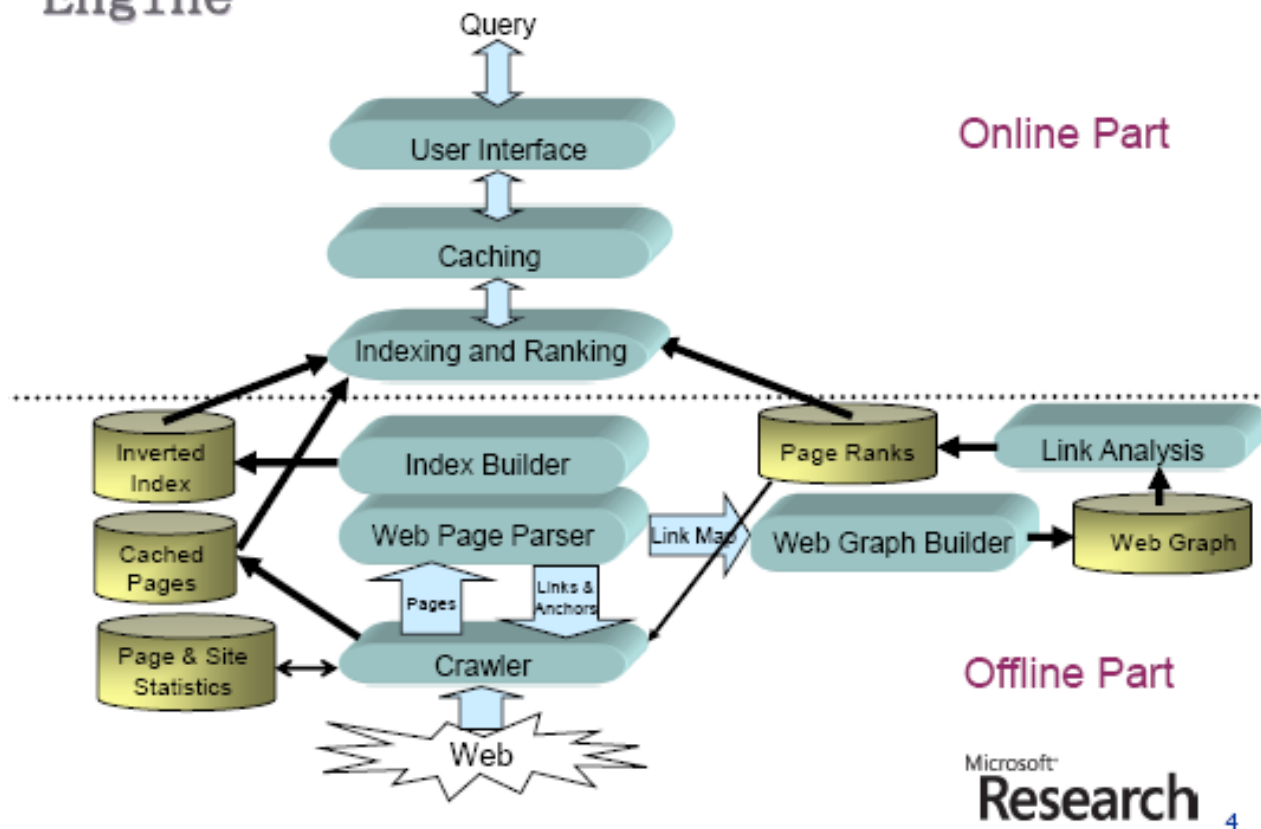
Программной частью поисковой системы является **поисковый движок** – комплекс программ, обеспечивающий функциональность поисковой системы

**Критерии качества поисковой машины:**

- Релевантность
- Полнота базы
- Учет морфологии

## II. Архитектура ПД

### Architecture of a Typical Search Engine



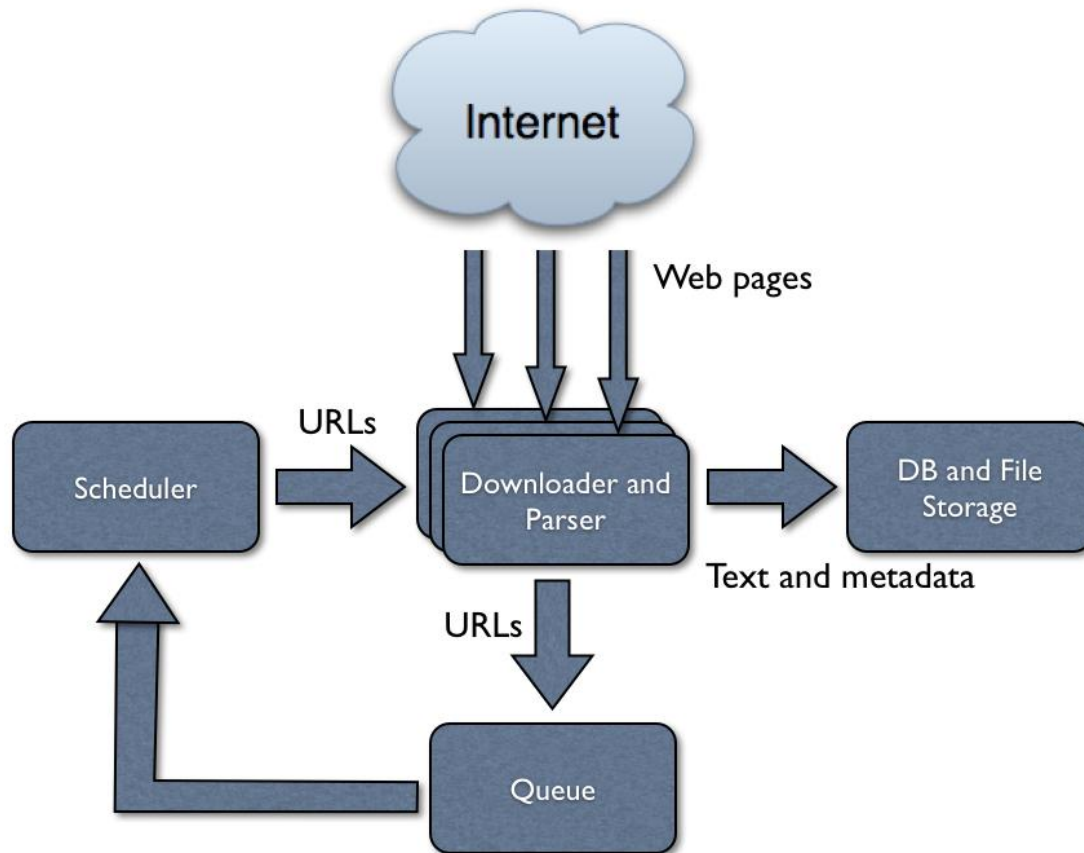




## II. Основные элементы

1. Web-crawling
2. Indexing
3. Searching

## II. Web-crawling



Web-краулер ака паук выполняет:

- Обход страниц по ссылкам
- Анализ содержимого страниц
- Сохранение содержимого страниц на сервере

*Robots.txt* – указания по обходу сайта

## II. Пример robots.txt

Запрет доступа всех роботов ко всему сайту:

*User-agent: \**

*Disallow: /*

Запрет доступа определенного робота к каталогу private:

*User-agent: googlebot*

*Disallow: /private/*

Разрешение доступа к одному файлу каталога:

*Allow: /album1/photo.html*

*Disallow: /album1/*

## II. Создание индекса

**Цель использования индекса** — в улучшении скорости и быстродействия при поиске релевантных документов по поисковому запросу



## II. Предметный указатель

### Б

- Бармаглот, 196, 198, 217
- Брандашмыг, 15, **18, 317**

### В

- Варкаться, 2, 18, 39

### З

- Зелюк, 3, 15, 47, 79, 115
- Злопастьность, **18, 45, 317**

### М

- Мова, 12, 16, 17

- Мюмзик, 8, 18, 191

### П

- Пыряться, 77, 128

### Р

- Рымать, 14

### С

- Свирлепость, 195, 278

### Х

- Хливкость, 33, 135
- Хрюкотать, 134, 156

## II. Поисковый индекс

- Заранее подготовленные данные для поисковой системы
- Все упоминания слов
- Номера предложений/слов
- Все слова, не только специальные термины
- Служебная информация

### **Задачи, решаемые при построении индекса:**

- Определение языка документа
- Определение формата и структуры документа
- Распознавание секций документа
- Токенизация (разбитие на слова)

## II. Типы индексов

- Прямой
- Обратный
- N-граммный
- Матрица «терм-документ»
- Суффиксное дерево
- Индекс цитирования

## II. Прямой и обратный индексы

- Прямой

Document	Words
Document 1	the,cow,says,moo
Document 2	the,cat,and,the,had
Document 3	the,dish,ran,away,with,the,spoon

- Обратный

Word	Documents
the	Document 1, Document 3, Document 4, Document 5
cow	Document 2, Document 3, Document 4
says	Document 5
moo	Document 7



## II. Матрица «терм-документ»

Матрица, содержащая частоты слов в документах.

Пример:

D1 = «I like databases»

D2 = «I hate hate databases»

	I	like	hate	databases
D1	1	1	0	1
D2	1	0	2	1

Более сложное – вес слова в документе  $tf*idf$

## II. TF\*IDF

TF\*IDF – мера важности слова в документе

Частота слова в документе: 
$$\text{TF} = \frac{n_i}{\sum_k n_k}$$

Количество документов, содержащих слово, в коллекции:

$$\text{idf}(t) = \log \frac{|D|}{|\{d : t \in d\}|}$$

Мера важности слова: 
$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t)$$

## II. Учет ссылок

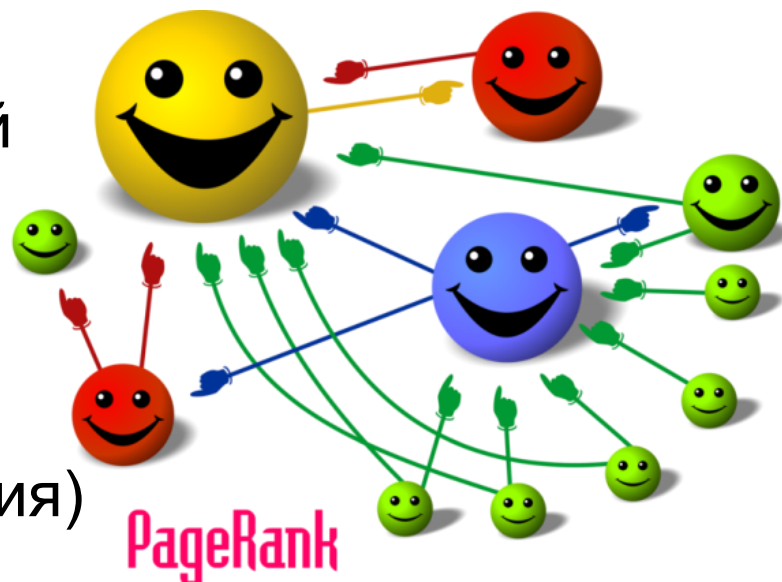
Вопрос: Как понять, что сайт хорош?

Ответ: Если на него ссылаются хорошие сайты, то, возможно, он и сам неплох? 😊

Если сайт указывает на плохой узел, то и он не слишком хорош.

Google: PageRank

Яндекс: ИЦ (индекс цитирования)



## II. SERP

Яндекс

Нашлось 5 млн ответов

Поиск [Почта](#) [Карты](#) [Маркет](#) [Новости](#) [Словари](#) [Блоги](#) [Видео](#) [Картинки](#) [ещё](#)

Ответ на главный вопрос жизни, вселенной и всего такого ...

☐ в найденном ☐ в Москве

Найти

расширенный поиск

[Мои находки](#) [Настройка](#) [Регион: Москва](#)

[Войти](#) [Помощь](#)

W [Ответ на главный вопрос жизни, вселенной...](#)

В книге Дугласа Адамса «Путеводитель для путешествующих автостопом по галактике» «**Ответ на главный вопрос жизни, вселенной и всего такого**» должен был решить все проблемы Вселенной. Этого ответа с нетерпением ждали все разумные расы.

[ru.wikipedia.org > ... Ответ... вопрос\\_жизни\\_вселенной и...](#)

[копия](#) [ещё](#)


Λ [42 — Lurkmore](#)

Дуглас Адамс, Серия «Путеводитель хитч-хайкера по Галактике» в библиотеке Альдебаран. Ответ знает даже калькулятор Гугля. Вольфрам альфа уверен. **Ответ на главный вопрос жизни, вселенной и всего такого** делить на стакан.

[lurkmore.ru > Жизни](#) [копия](#) [ещё](#)

[Разместить объявление по запросу «Ответ на главный ...»](#)

[Видео «Ответ на главный ...»](#)

 01:04

Белое солнце пустыни - Сорок два, 42

[Все видеоролики](#)

Весьма приблизительное значение

Сниппет

А как определяется порядок следования результатов?

**Архитектуры ИС. Примеры**

## II. Функция ранжирования

Функция сортировки найденных документов в соответствии их релевантности исходному запросу

Наиболее простые зависят от:

- TF
- TF-IDF

На данный момент используется машинное обучение со множеством параметров

## II. Функция ранжирования

Признаки, используемые при машинном обучении:

- Независимые от запроса (ИЦ, PageRank)
- Зависимые от запроса (TF, TF-IDF)
- Признаки самого запроса (н-р, количество слов)
- Различные их комбинации

## II. Меры качества

Точность (precision)

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Полнота (recall)

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Помимо этого, существуют также DCG, MAP и др.

## II. Еще немного о запросах

### Классификация запросов:

- Информационные. Цель – получение определенной информации, не важно, где именно

*Пример: текст песни о Чебурашке*

- Навигационные. Цель – навигация, поиск сайта, где по его предположению расположена необходимая информация

*Пример: вконтакте, сайт бауманки*

- Транзакционные. Цель – совершение сделки, купли-продаже, какой-либо транзакции

*Пример: купить наушники, продажа квартир, установка дверей*

**Архитектуры ИС. Примеры**



## II. Еще немного о запросах



статистика запросов

[Помощь](#)

[Статистика посещений](#)

[по словам](#)

[по регионам](#)

[по месяцам](#)

Ключевые слова:

новый год

[Показать](#)

Шкала графика:

[абсолютная](#)

[относительная](#)

Запросов за последние 30 дней: 864864, за период: 3345379

