

# Client Server Architecture

На того ли архитектора я пошел учиться

# Системная Архитектура

## Архитектура КС

Высокоуровневый дизайн, который лежит в основе системы

### Что входит?

- Компоненты
- Коллаборация
- Коммуникация

# Классические шаблоны

- Клиент-сервер
- Слои
- Peer-to-peer
- Каналы и фильтры
- И так далее

# Клиент - Сервер

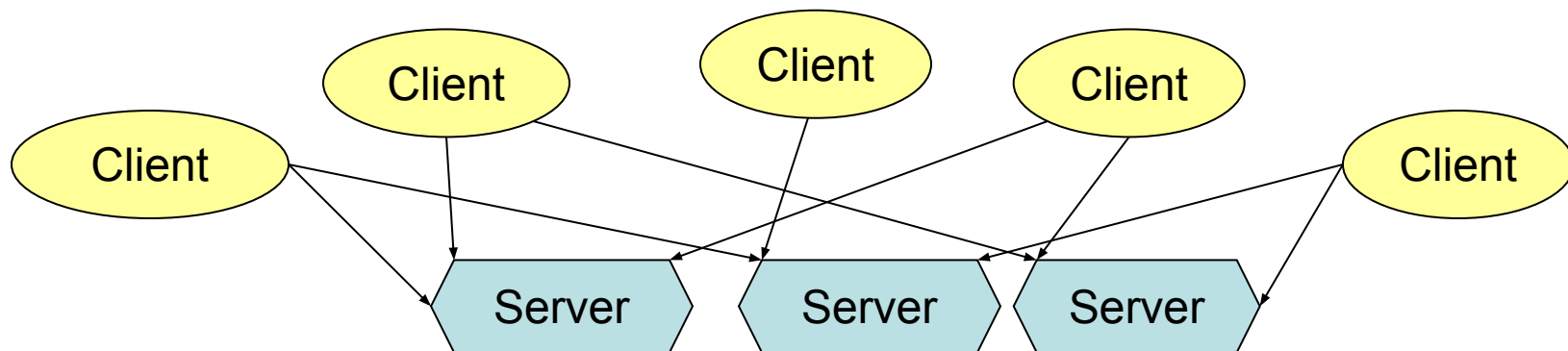
Каждый компонент системы или клиент или сервер

- Клиент - компонент, который делает запросы
- Сервер - компонент, который обрабатывает запросы

# Чувак, это Вебчик

Так получилось, что Веб - это клиент-серверная система

Клиентом выступает браузер, а сервера отдают странички



# Уровни архитектуры Клиент-Сервер

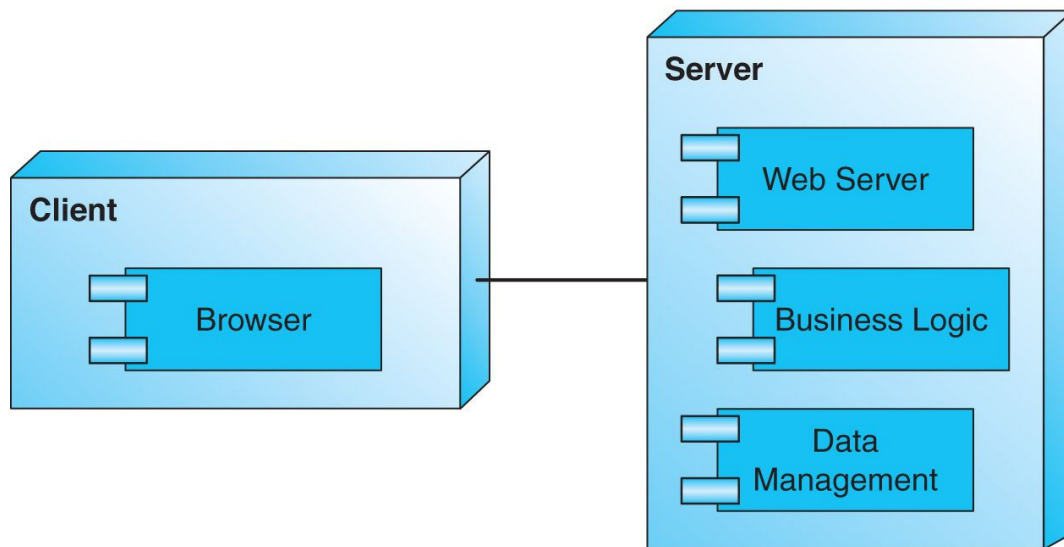
Обычно веб приложения создаются на 2-ух, 3-ех и N(много) уровнях

Каждый уровень - это платформа с определенными функциями

# 2-ух уровневая архитектура

Уровень 1 - Клиент, компьютер на котором запущен браузер

Уровень 2 - Сервер, платформа на которой запущены все компоненты



# Характеристики 2-ух уровневой архитектуры

Плюсы: цена

Минусы: зависимость компонентов, масштабирование

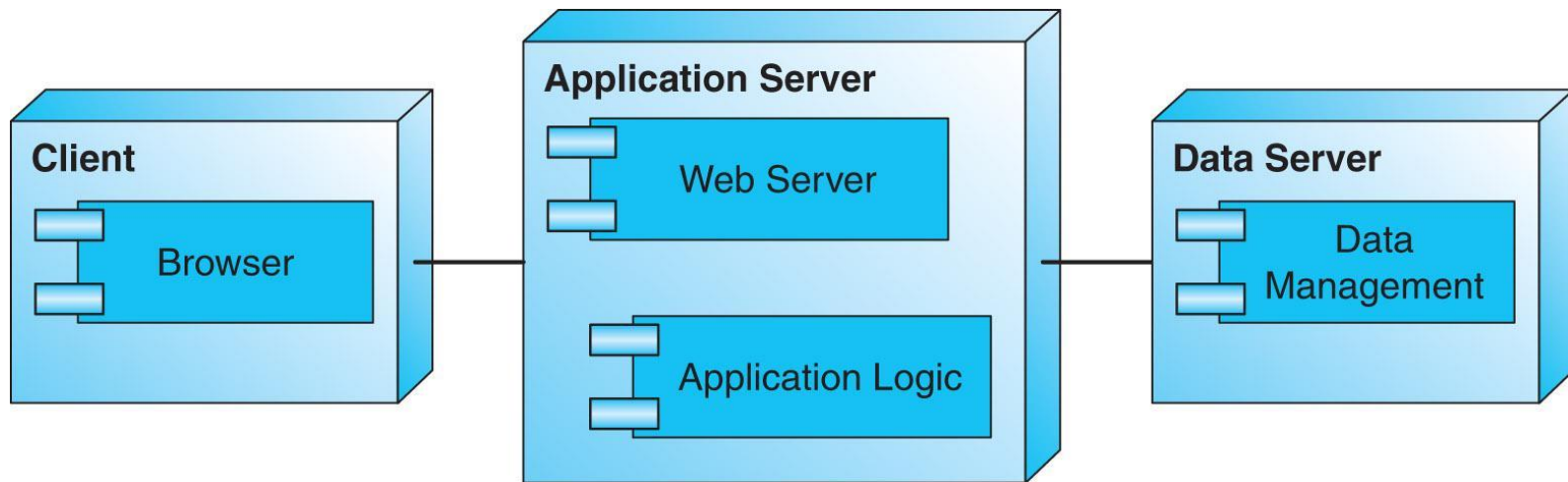
Обычно: маленькие компании, до 100 пользователей



# 3-ех уровневая архитектура

В прочему как 2-ух уровневая

Отличается появлением 3-его уровня отвечающего за данные



# Характеристика 3-ех уровневой архитектуры

Плюсы: производительность (специальная аппаратура), меньше зависимости между компонентами, масштабируемость

Минусы: цена

Обычно: 100-1000 пользователей, маленький бизнес

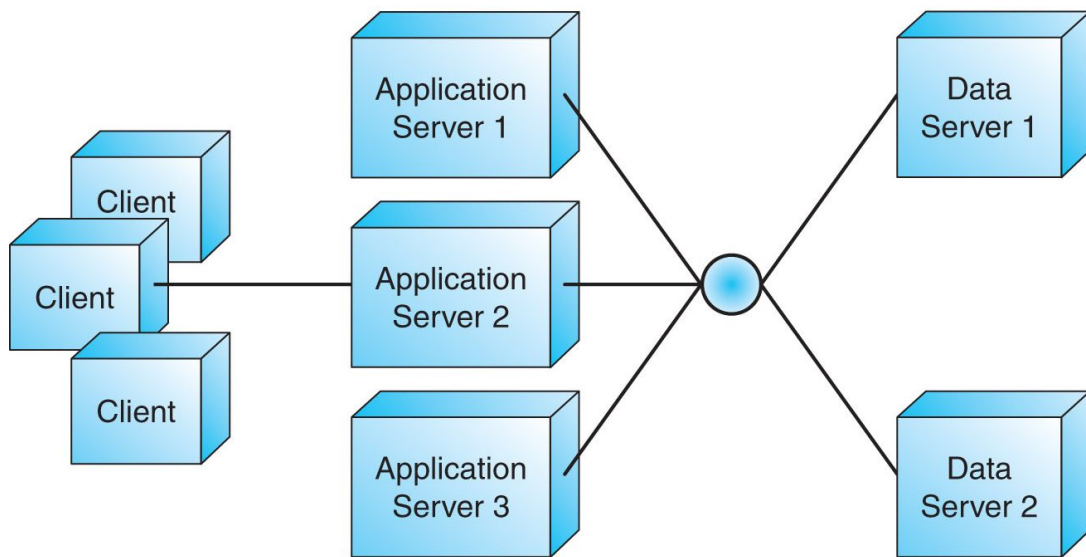
# Многоуровневая архитектура

Развитие тематики трех уровней одним из способов

- Репликация функций уровня
- Специализация функций уровня
- Утилитарные сервисы

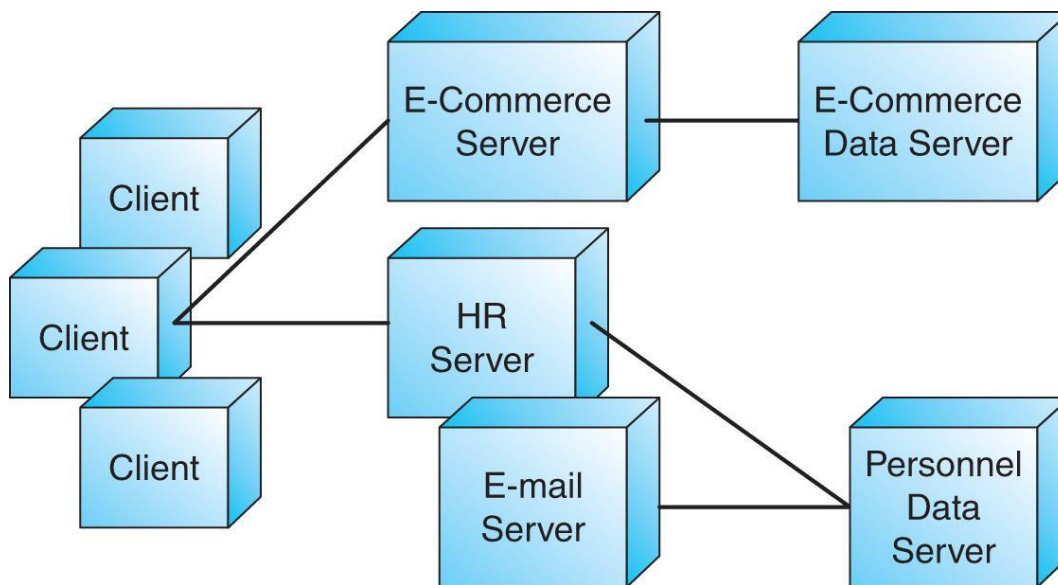
# Репликация

- Репликация серверов приложений и серверов данных
- Сервера делят нагрузку между собой



# Специализация

- Определенные сервера под определенные задачи
- Сервера отвечают за часть нагрузки - по задачам



# Утилитарные сервисы

- Обработывают входящий трафик и понижают нагрузку
- Балансеры, фаерволы и прочие зверьки

