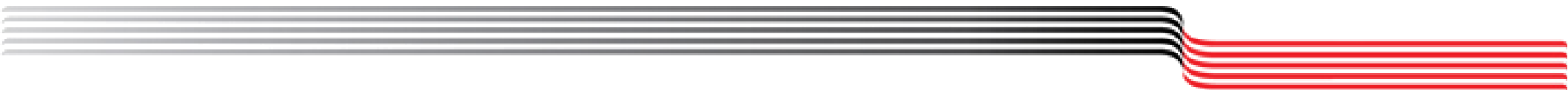


Архитектура web-приложений

Виталий Фирстов



Архитектура web-приложений

Что такое архитектура?

Что рассматриваем под веб-приложением?

Архитектура

Архитектура - это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

[IEEE 1471]

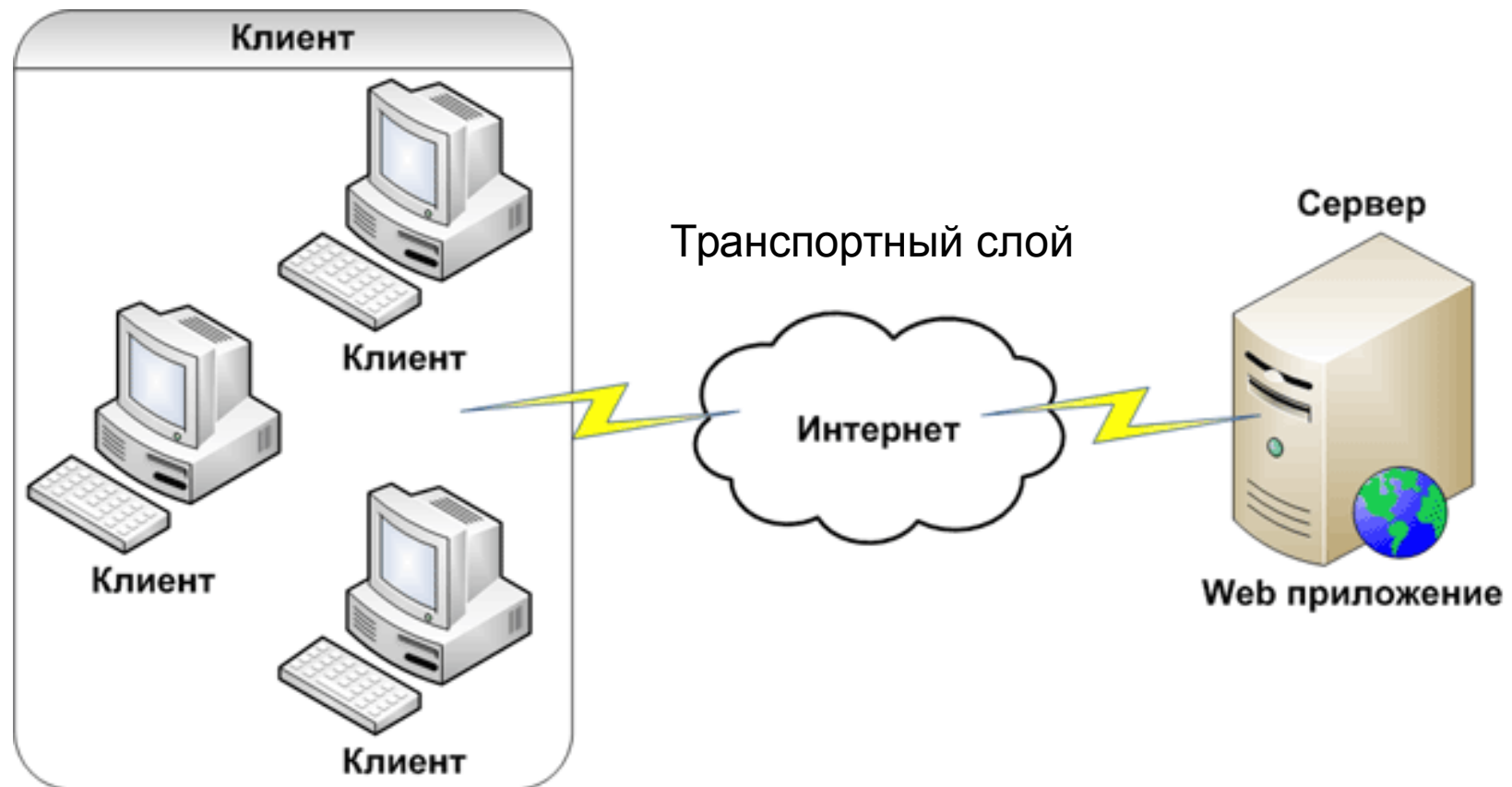
Архитектура

Архитектура - это **набор значимых решений** по поводу организации системы программного обеспечения, набор структурных элементов и их интерфейсов, при помощи которых компонуется система, вместе с их **поведением**, определяемым во взаимодействии между этими элементами, **компоновка** элементов в постепенно укрупняющиеся подсистемы, а также **стиль архитектуры** который направляет эту организацию - элементы и их интерфейсы, взаимодействия и компоновку.

Крачтен Kruchten

Веб-приложением — будем называть любое интерактивное приложение, которое выполняется на клиентском устройстве в браузере, получая информацию от серверной части. Обмен информацией происходит по сети, логика распределена между клиентским и серверным приложением, хранение данных осуществляется, преимущественно, на сервере.

WEB-Приложение



Особенности web-приложений

Много клиентов

Сервер

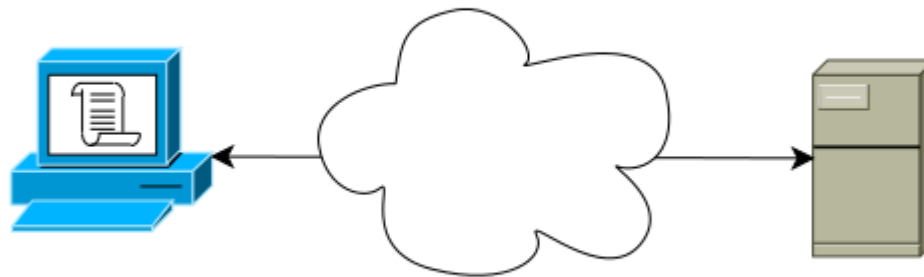
WWW

HTTP — протокол, как транспорт

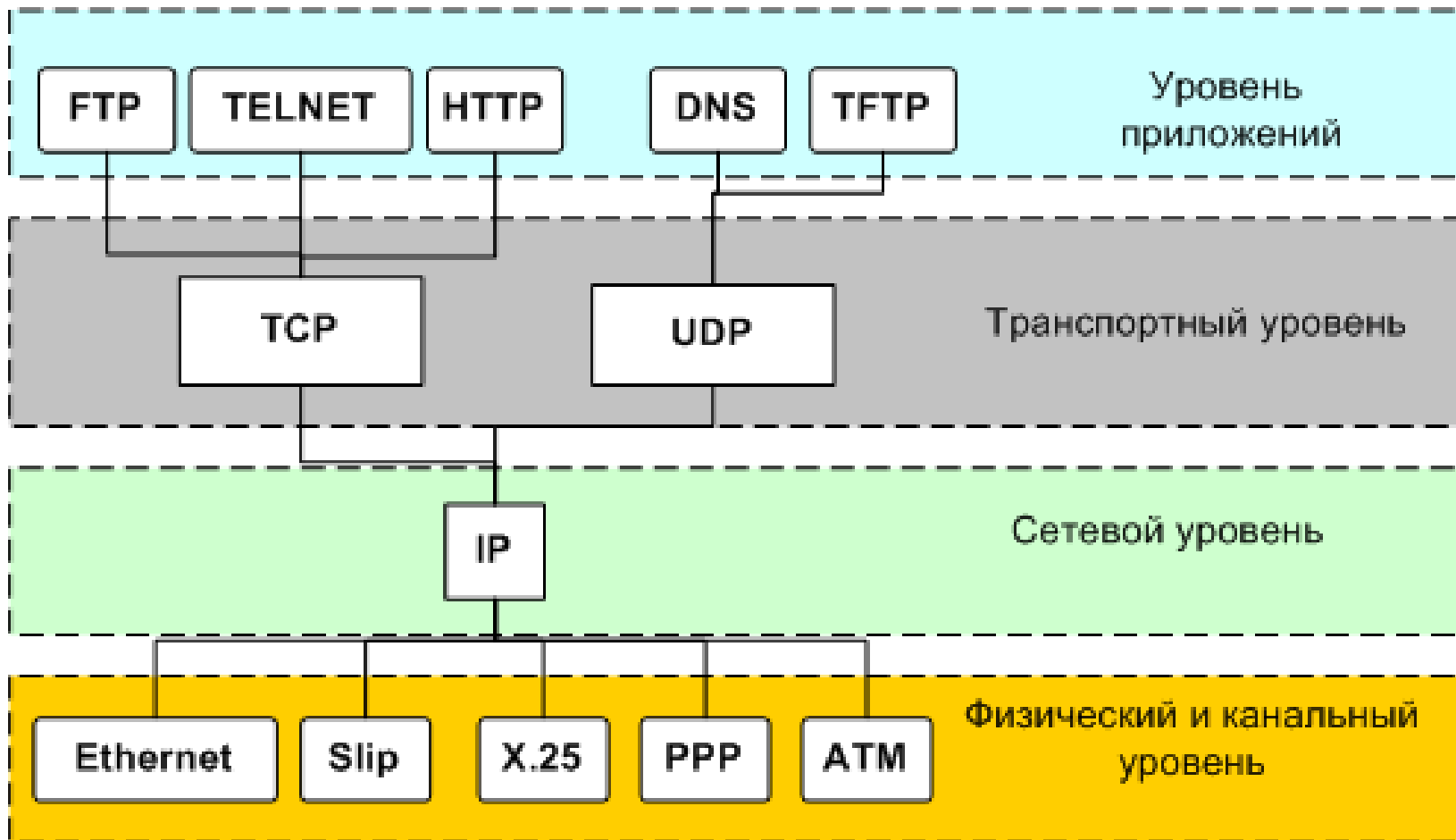
Первые шаги

Статические HTML страницы
Документо-ориентированная сеть

Common Gateway Interface (CGI)
Каждый запрос к серверу инициирует выполнение
CGI-программы в реальном времени



Стек протоколов



HTTP

- Текстовый, человекочитаемый
- Простой
- Не хранит состояния
- Нет шифрования

TCP-порт 80



- GET – запрос содержимого
- POST – передача параметров
- HEAD – только заголовки, без тела
- PUT – загрузить содержимое
- DELETE – удалить ресурс

HTTP, методы

GET / HTTP/1.1

Host: xbb.uz

User-Agent: Mozilla/5.0 ...

Accept: text/html,application ...

Accept-Language: ru-ru,ru ...

Accept-Encoding: gzip,de ...

...



Запрос

HTTP/1.0 200 OK

Server: nginx/0.7.67

Date: Tue, 08 Feb 2011 08: ...

Content-Type: text/html; c ...

X-Powered-By: PHP/5.2.12

Expires: Thu, 19 Nov 1981 ...

...



Ответ

HTTPS

Обычный HTTP

Работающий через шифрованные транспортные механизмы SSL и TLS

TCP-порт 443



RPC (Remote Procedure Call)

SOAP (Simple Object Access Protocol)

REST (Representational State Transfer)

WebSocket (протокол полнодуплексной связи поверх TCP соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени)

Двухзвенная архитектура

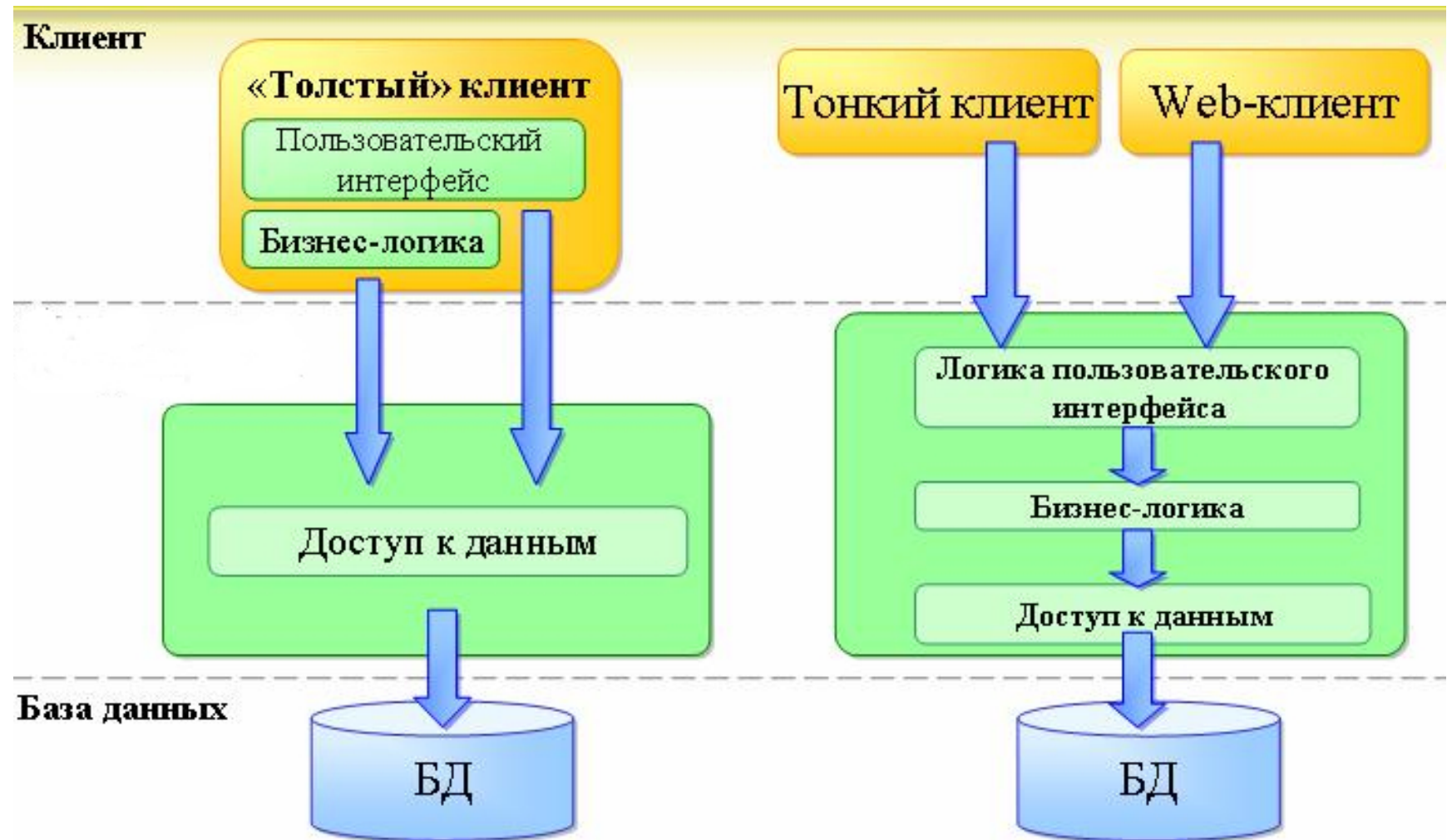
1. «Тонкий» клиент
2. «Толстый» клиент

Клиент:
Отображение

Сервер:
Данные

Что делать с бизнес-логикой?

Шаблоны архитектуры



Область применения

Этот шаблон больше всего подходит для Web-приложений или их окружения, когда клиент обладает минимальными вычислительными возможностями или не может управлять своей конфигурацией.

Примеры использования

Данный шаблон используется в большинстве Internet-приложений электронной коммерции, поскольку нет никакого смысла отказываться от покупателей только потому, что их клиентский компьютер не обладает достаточной вычислительной мощностью. Типичное приложение электронной коммерции предназначено для привлечения максимального количества покупателей.

Особенности

- высокие требования к производительности клиента
- работа в разных окружениях
- высокие требования к серверу (если он один), как следствие плохая масштабируемость

Выводы

Архитектура толстого и тонкого клиента
имеет свои недостатки,

что же делать?

Шаблоны архитектуры

для начала обсудим

Стратегии развития масштабируемых
веб-приложений

Так или иначе практически
все интернет-приложения стремятся к

- высокой доступности
- масштабируемости
- производительности
- управляемости
- низкой стоимости эксплуатации
- богатству функциональных возможностей
- генерации прибыли

Высокая доступность

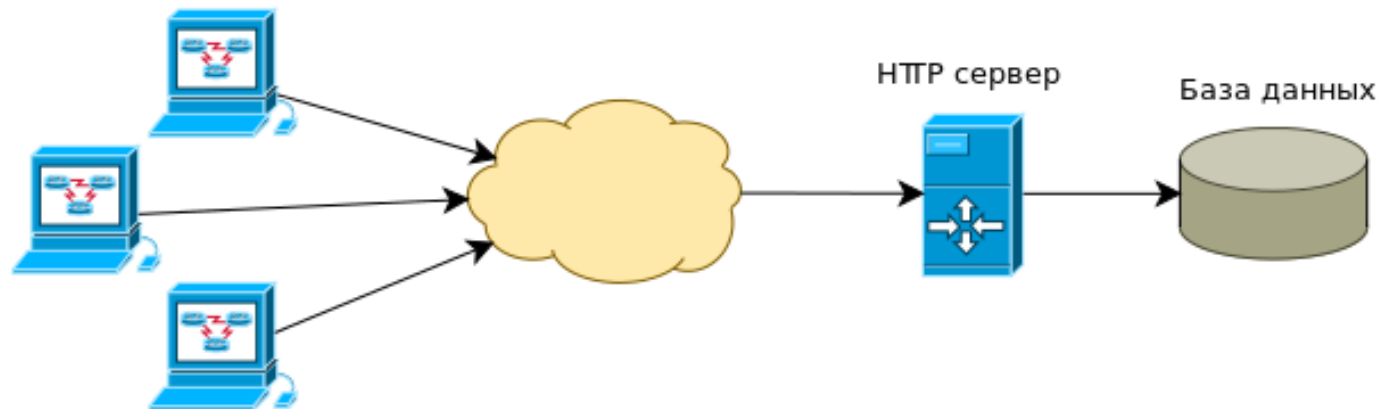
это результат проектирования и реализации,
который обеспечивает заданную степень
непрерывности функционирования

Масштабируемость

это желаемое свойство системы,
которое показывает ее способность
достойно справляться с возрастающей нагрузкой,
либо готовность к расширению
в случае необходимости

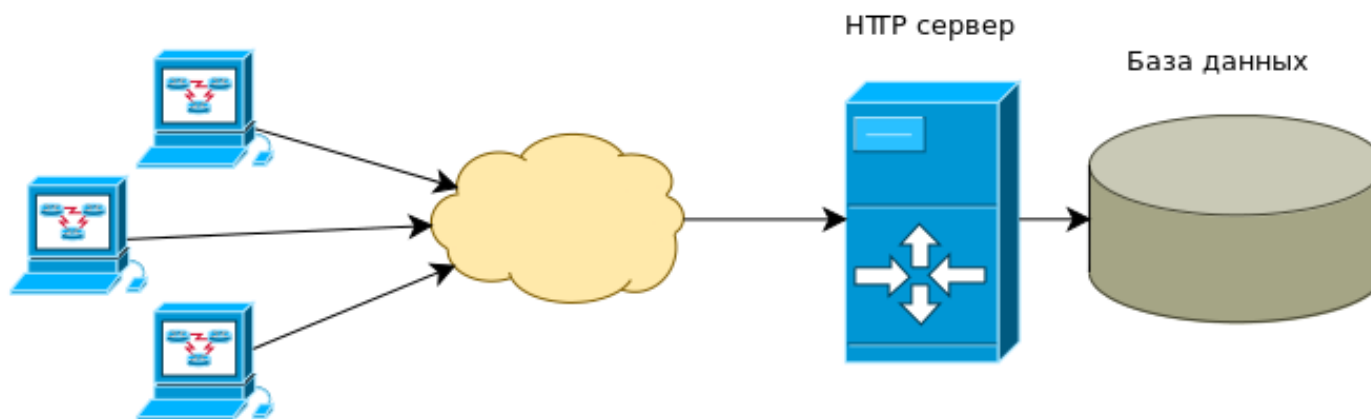
Шаблоны архитектуры

Рассмотрим типичную ситуацию



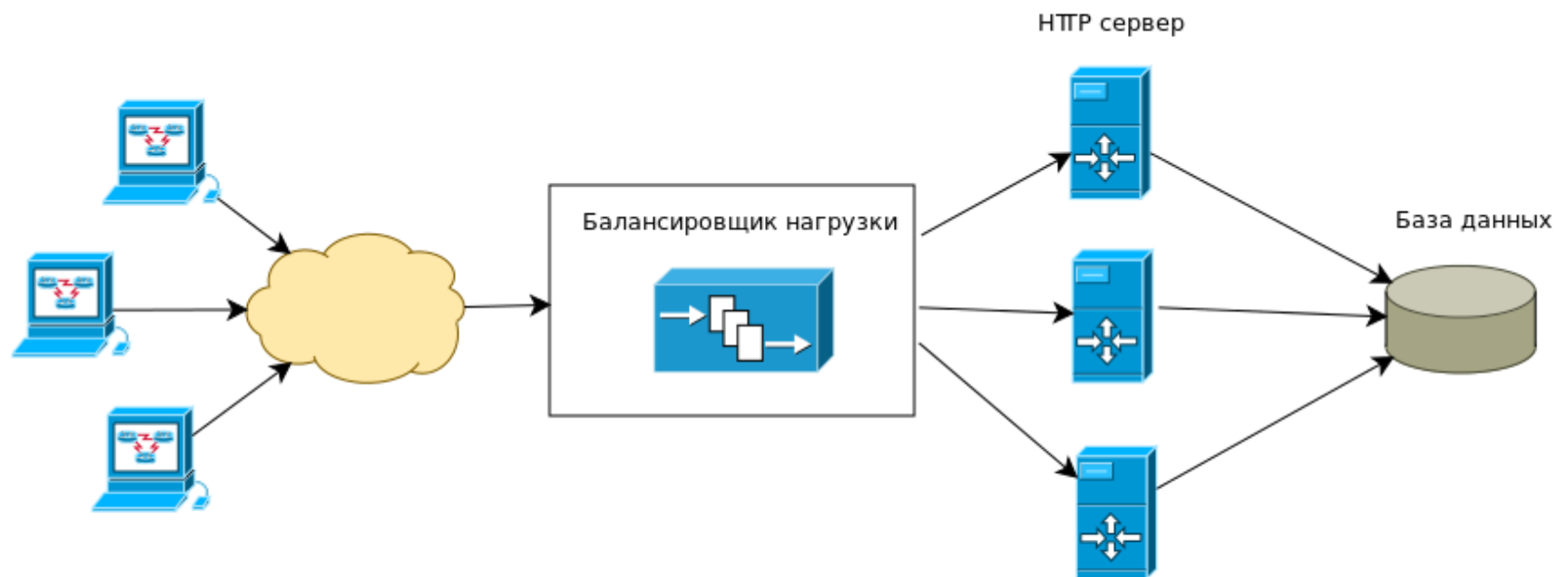
Шаблоны архитектуры

Добавили процессоров, памяти



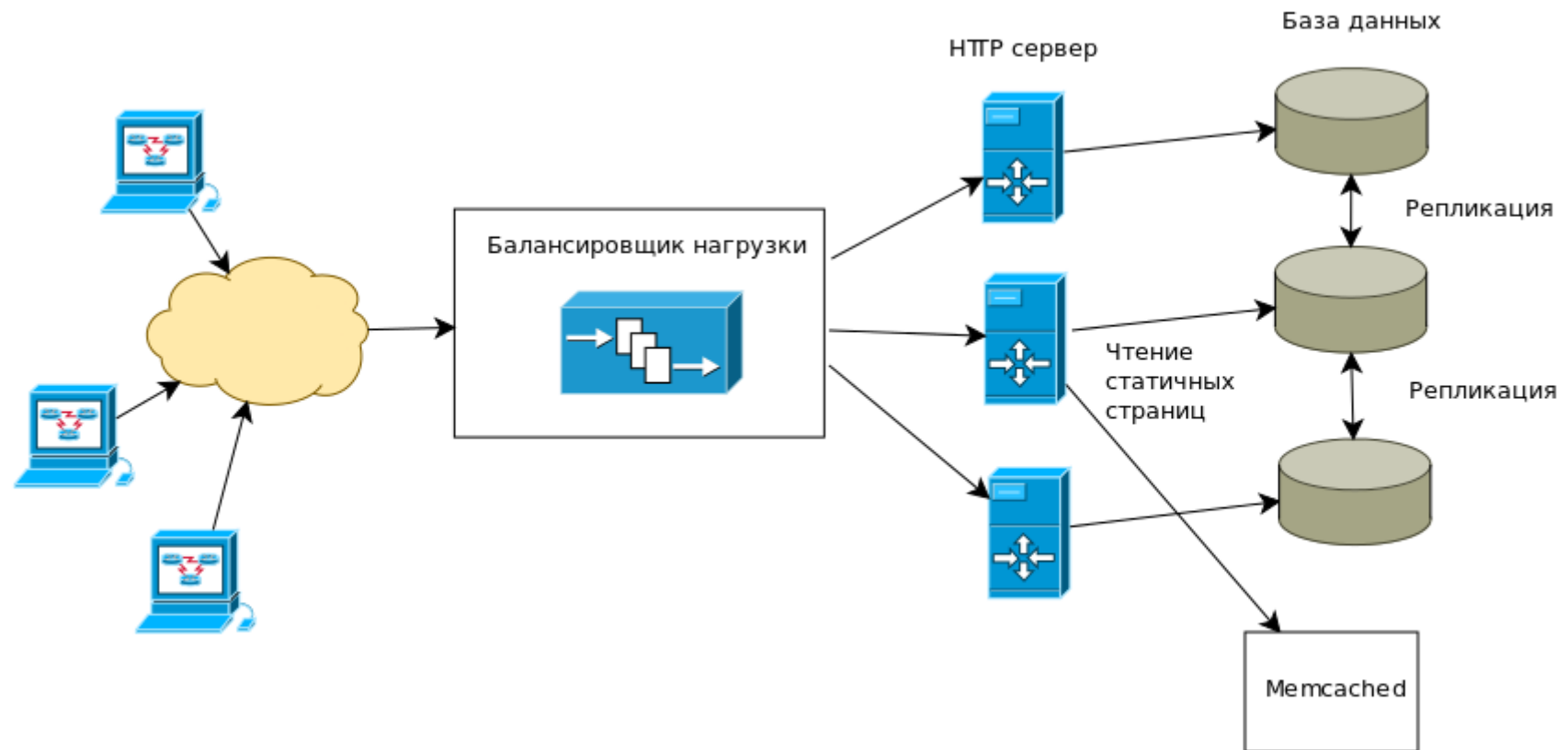
Шаблоны архитектуры

Добавили балансировщик и сервера



Шаблоны архитектуры

Добавили репликацию и кэширование



Шаблоны архитектуры

Секционирование (partitioning)

Есть поддержка из «коробки»

IBM DB2, Informix, Oracle Database, Teradata Database, Microsoft SQL Server, PostgreSQL, MySQL

Репликация

Шардинг

Шаблоны архитектуры

Проблемы становятся критическими

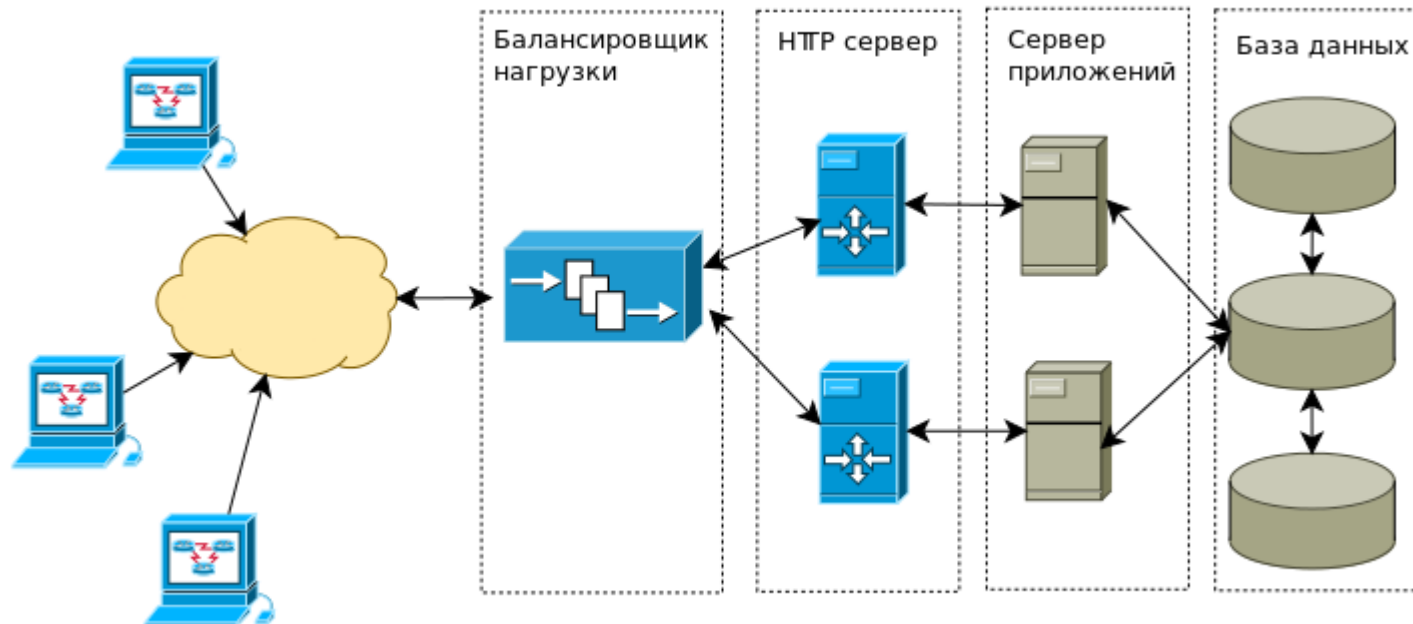
Возникает паника — «Разве мы не сделали это раньше?».
Нужно пересматривать бизнес-модель и переделывать
практически все приложение.

Неизменно возникает вопрос:

«Почему же мы сразу не проектировали
наше приложение
с учетом масштабируемости?»

Шаблоны архитектуры

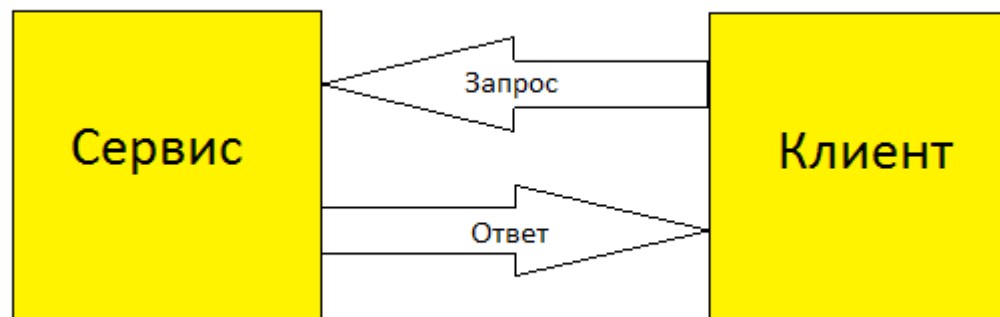
Многоуровневая архитектура



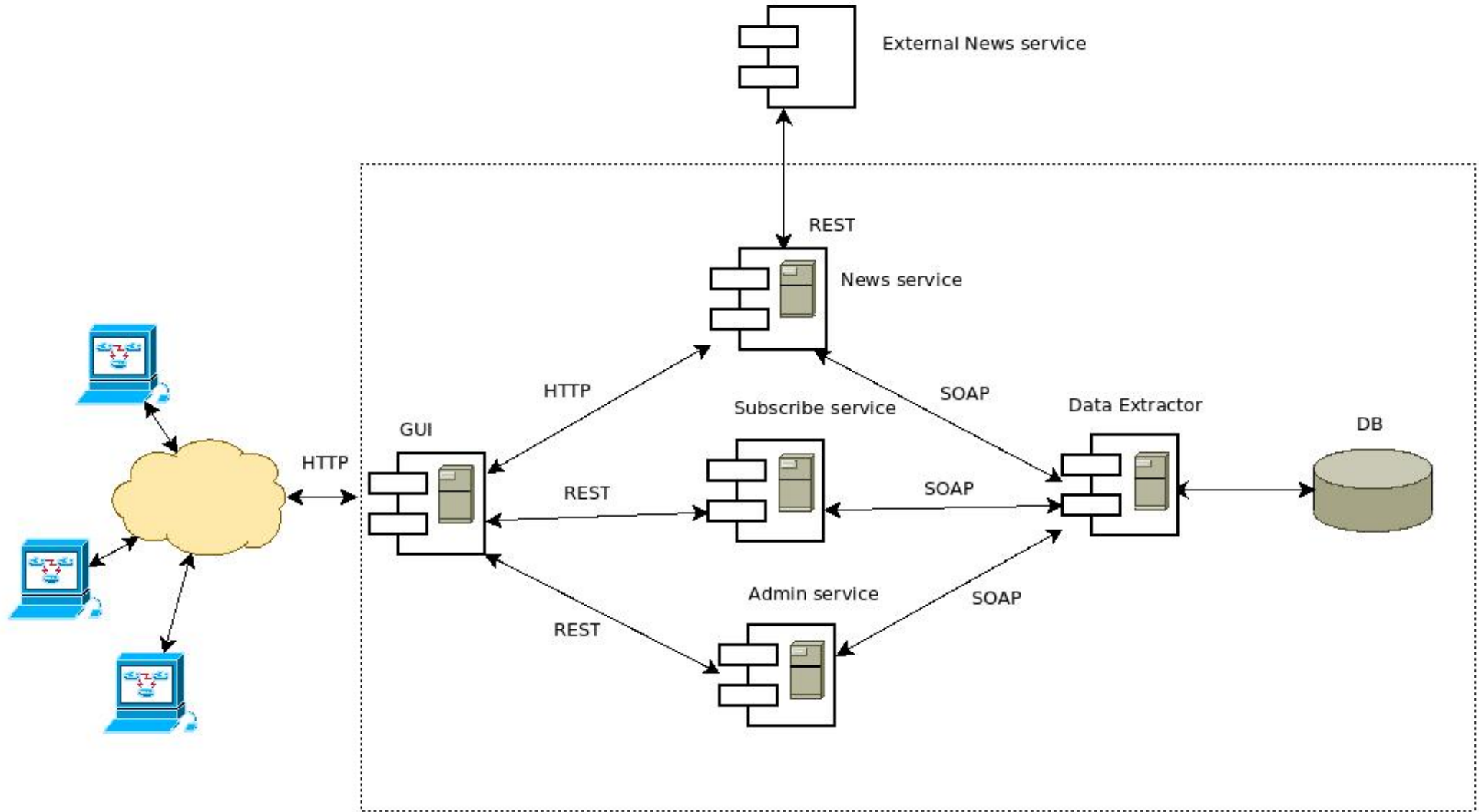
Шаблоны архитектуры

Сервис-ориентированная архитектура (SOA, service-oriented architecture)

модульный подход к разработке программного обеспечения, основанный на использовании сервисов со стандартизированными интерфейсами



Шаблоны архитектуры



Шаблоны архитектуры

- SOA архитектура не привязана к какой-то определенной технологии;
- независимость организации системы от используемой вычислительной платформы;
- независимость организации системы от применяемых языков программирования; использование сервисов, независимых от конкретных приложений, с единообразными интерфейсами доступа к ним;
- организация сервисов как **слабосвязанных** компонентов для построения систем

Вопросы?