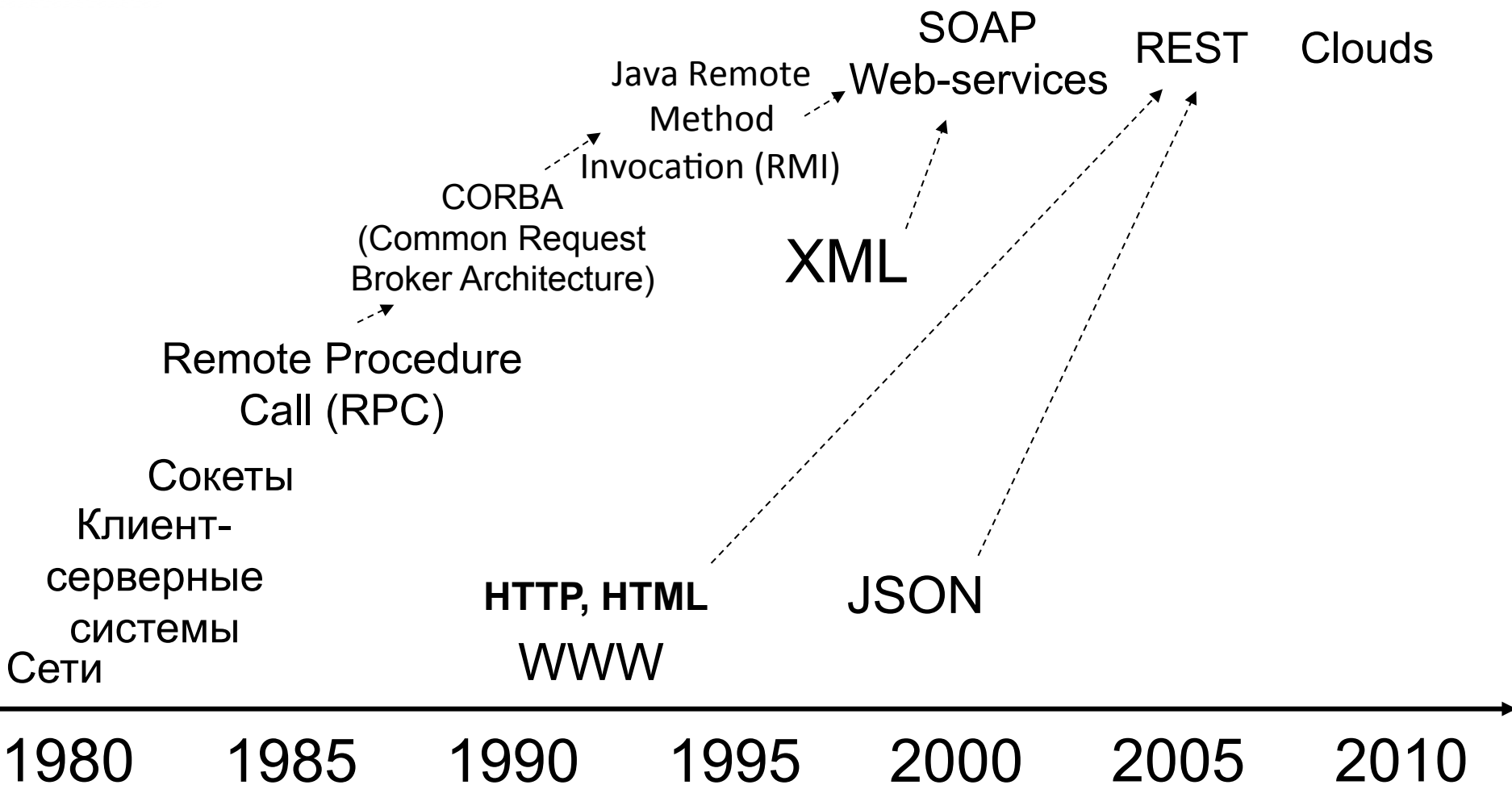


РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Сервис-ориентированная архитектура

СТАНДАРТЫ, ПРОТОКОЛЫ, АРХИТЕКТУРЫ РВС



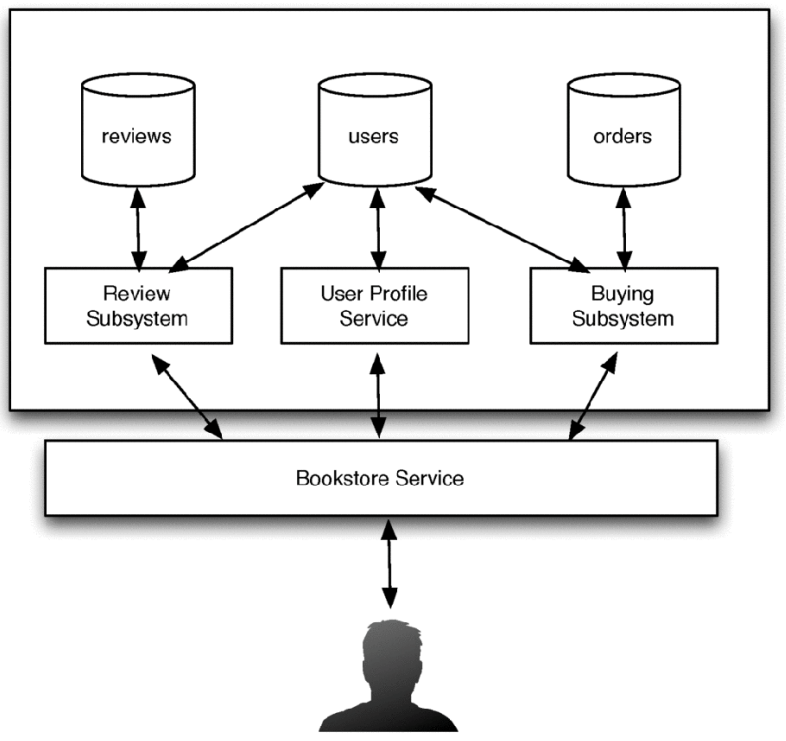
ВВЕДЕНИЕ: ПИСЬМО ДЖЕФФА БЕЗОСА (2002)

- ⦿ Все команды с настоящего момента обязаны предоставлять данные и функциональность через сервисные интерфейсы.
- ⦿ Команды должны взаимодействовать друг с другом посредством этих интерфейсов
- ⦿ Другие методы меж-процессной коммуникации запрещены: никакого прямого доступа к чужой оперативной памяти, никакого прямого доступа к чужим хранилищам данных и др. Всё взаимодействие – только через сервисные интерфейсы по сети.
- ⦿ Не важно, какую технологию вы используете (HTTP, Corba...)
- ⦿ Все интерфейсы сервисов должны быть спроектированы таким образом, чтобы быть расширяемыми. Команды должны разрабатывать интерфейсы так, чтобы сервисы можно было предоставить потребителям из внешнего мира. Без исключений.
- ⦿ Любой, кто не следует этим правилам будет уволен.
- ⦿ Спасибо, приятного дня!

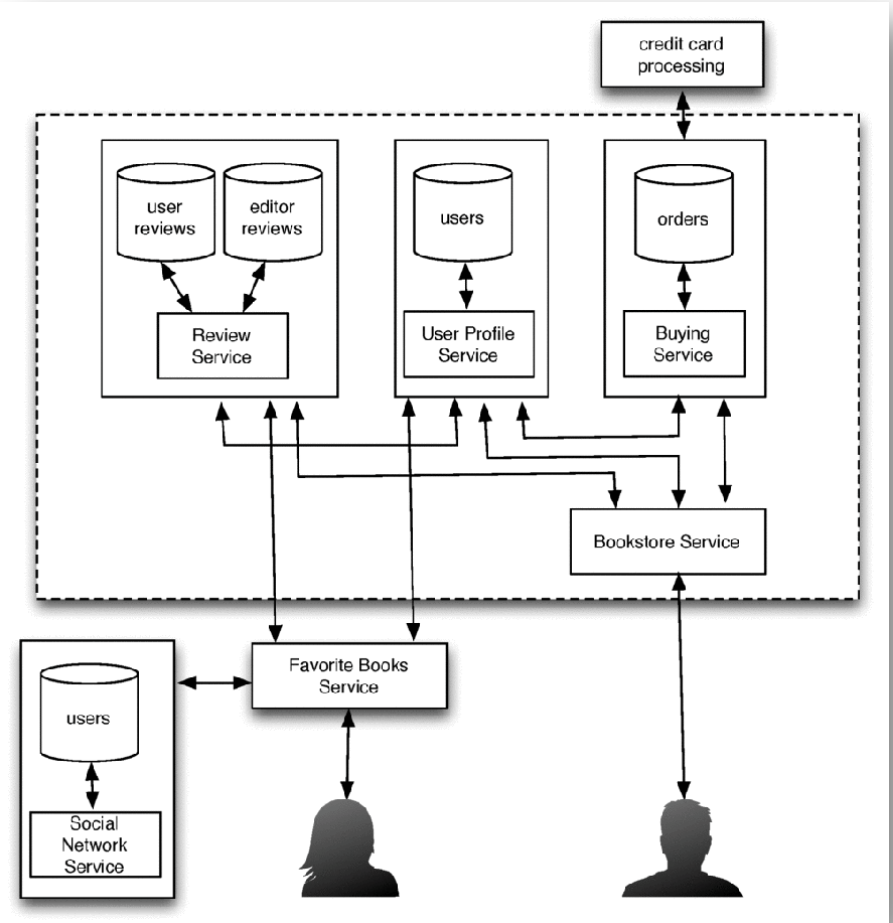
FACEBOOK (2007) И GOOGLE+

- ⊙ Аналогичный переход сделал Facebook в 2007 году
- ⊙ Была запущена Facebook Platform, позволив сторонним программистам разрабатывать приложения, которые могли получать доступ к социальному графу и другим данным пользователей в Facebook.
- ⊙ Google+ очень долго ругали за отсутствие API, но когда он был выпущен в сентябре 2011 года он оказался «монолитным» и абсолютно не «SOA-like», обеспечивая лишь простейший доступ к профилю пользователя и всему, что он видит.

ХРАНИЛИЩЕ VS SOA



«Хранилище»



SOA

СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА РАСПРЕДЕЛЕННЫХ СИСТЕМ

СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА (SOA)

- ◎ **Сервис ориентированная архитектура** – это парадигма организации и использования распределенных функциональных возможностей, которые могут предоставляться различными владельцами
 - ◎ Определение организации OASIS (Organization for the Advancement of Structured Information Standards)
- ◎ По существу, SOA означает, что *компоненты приложения* представляют собой взаимодействующие *сервисы*, которые могут быть использованы *независимо* друг от друга, или же вообще могут быть *отделены* для реализации других приложений.

ОБЛАСТИ ПРИМЕНЕНИЯ SOA

- ◎ Осуществление совместного использования информации путем объединения внутренних ресурсов отдельных компаний (B2B)
- ◎ Готовые модули для разработчиков
- ◎ Обеспечение дополнительных возможностей клиентских приложений
- ◎ Инструменты для обеспечения взаимодействия программ в рамках одной компании

Стили API ВЕБ-СЕРВИСОВ

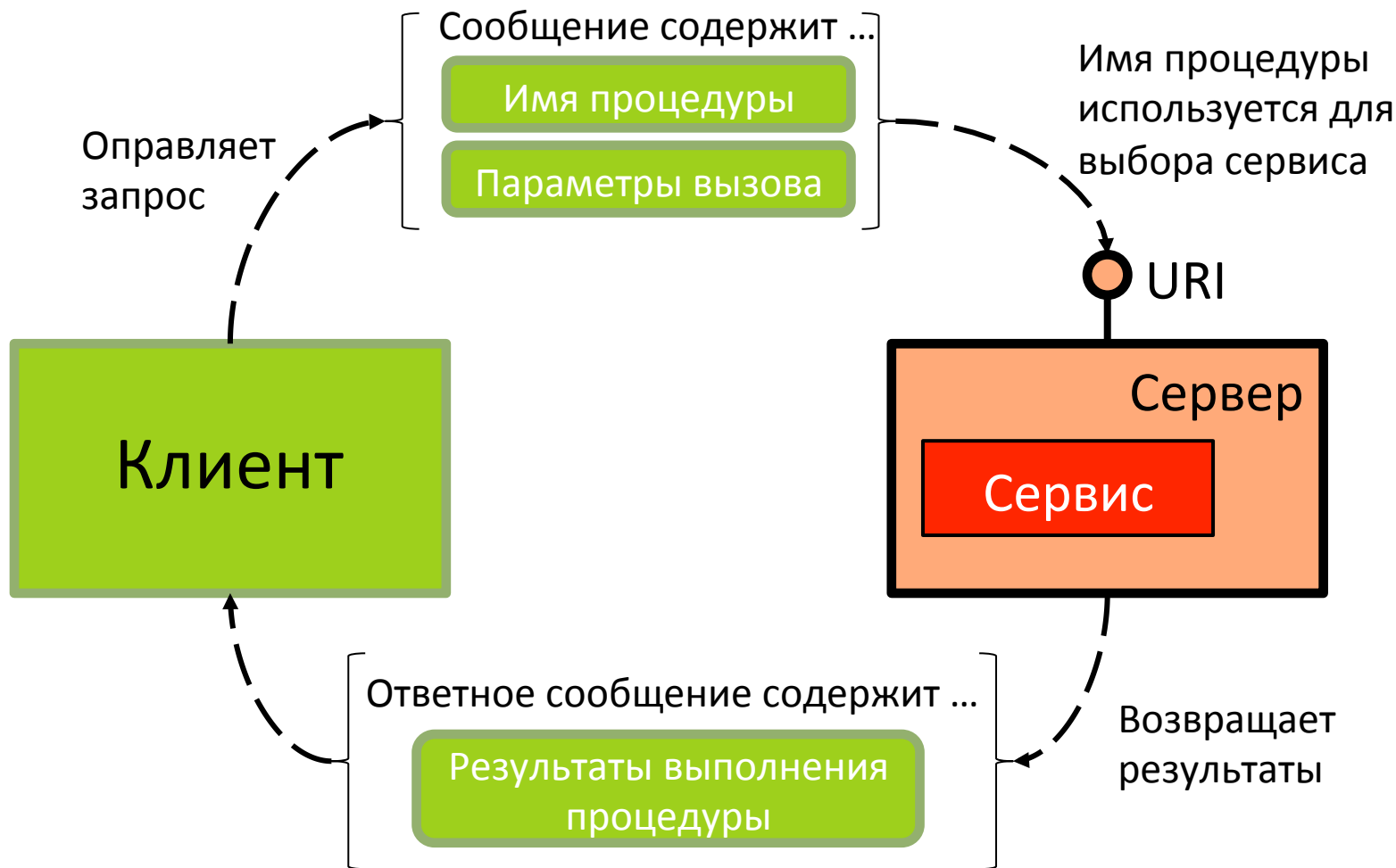
Стиль Веб-сервиса	Проблема
RPC API	Как клиенты могут выполнить удаленные процедуры посредством HTTP?
API сообщений	Как клиенты могут отправлять команды, нотификации или другую информацию удаленной системе по HTTP, но не привязывая себя к удаленным процедурам?
API ресурсов	Как клиент может манипулировать данными, предоставляемыми удаленной системой по HTTP, без связывания себя с удаленными процедурами, а также без необходимости создания специального проблемно-ориентированного API?

RPC API

Стили API ВЕБ-СЕРВИСОВ: RPC API

- ⊙ Необходимо определить сообщения, которые бы описывали:
 - ⊙ *удаленные процедуры,*
 - ⊙ *а также набор элементов, которые формируют набор параметров для удаленной процедуры.*
- ⊙ Клиент должен отправить сообщение с этой информацией по указанному URI для выполнения процедуры.

Стили API ВЕБ-СЕРВИСОВ: RPC API

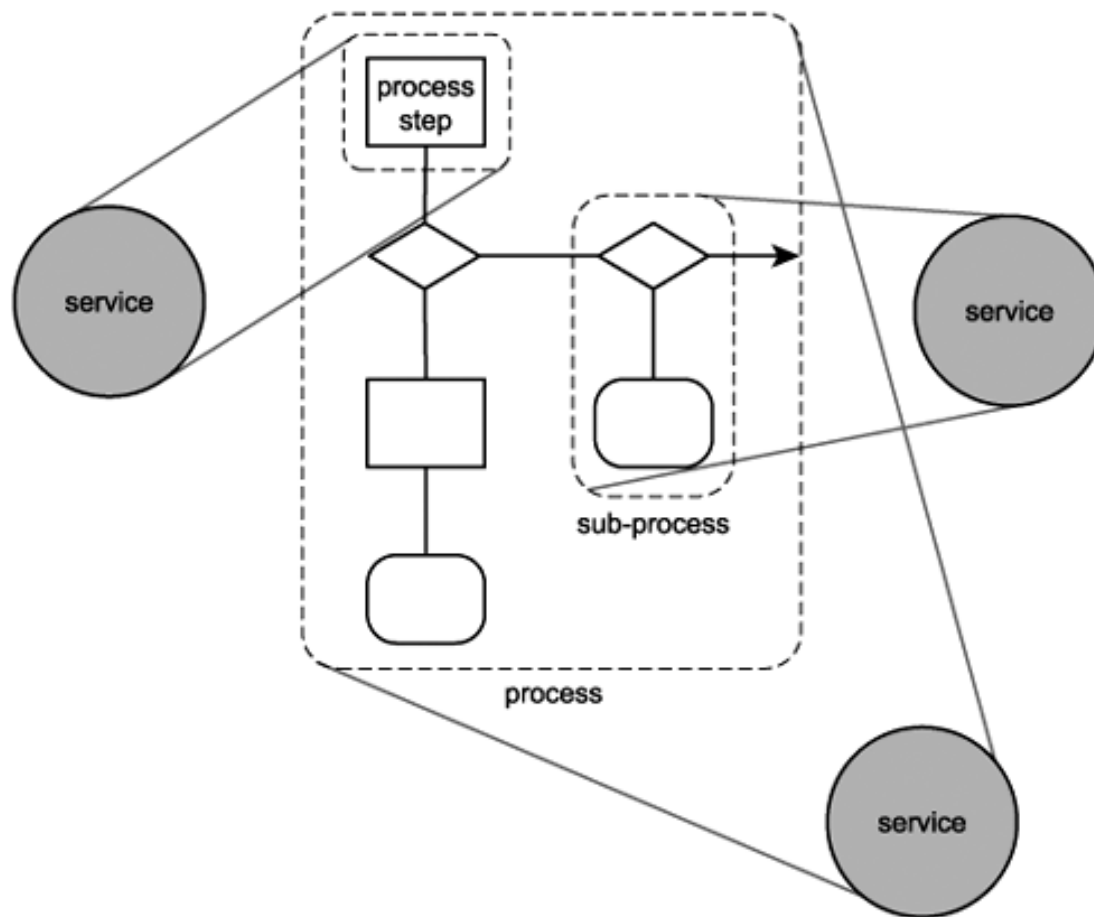


ОБМЕН СООБЩЕНИЯМИ

- ⊙ Использование низкоуровневого транспортного протокола для обмена сообщениями (BSD Socket API) приводило к необходимости сложной трансформации данных, чтобы типы данных, определенные у клиента можно было использовать на сервере.
 - ⊙ *На разных платформах могут использоваться различные методы кодировки данных (ASCII , EBCDIC , UTF-8, UTF-16, Endianness) для представления и хранения данных.*
- ⊙ Технологии CORBA и DCOM решили предложили решения для данной проблемы, посредством собственных абстракций над низкоуровневыми типами данных.
- ⊙ Но появились проблемы при связи между различными инфраструктурами (CORBA и .NET) + обмен сообщениями через Интернет (т.к. часто на брандмауэре закрыты все порты, кроме 80-го).

- ◎ Применение протокола HTTP решает многие из этих проблем, т.к. обеспечивает возможность взаимодействия клиентов и серверов, базирующихся на различных платформах через Интернет.
- ◎ Для организации RPC посредством HTTP существует множество реализаций на разных языках:
 - ◎ JAX-WS (Java)
 - ◎ WCF (Microsoft .NET)
- ◎ Они позволяют создавать простые веб-процедуры и клиентов к ним без необходимости понимания форматов данных (XML, JSON), методов кодирования (UTF-8 и др.) или описания структуры сервисов (WSDL).

RPC-СЕРВИСЫ ИНКАПСУЛИРУЮТ ДЕЙСТВИЕ



МЕЛКОМОДУЛЬНЫЙ СЕРВИС


- ◎ ***Мелкомодульный сервис***
предоставляет элементарный объем функциональной нагрузки и обеспечивает высокую степень повторного использования.
- ◎ Необходимо координировать работу нескольких сервисов для получения результата

Крупномодульный сервис

- ◎ **Крупномодульный сервис** обеспечивает высокую степень *инкапсуляции* функциональности.
- ◎ Но затрудняет повторное использование, в связи с узкой специализацией

Крупномодульный сервис РЕШЕНИЯ КВАДРАТНЫХ УРАВНЕНИЙ

$$a=5; b=10; c=3$$



Решить
уравнение
 $ax^2+bx+c=0$

$$x_1 = -0.3675$$
$$x_2 = -1.6325$$

ОРКЕСТРАЦИЯ МЕЛКОМОДУЛЬНЫХ СЕРВИСОВ ДЛЯ РЕШЕНИЯ КВАДРАТНЫХ УРАВНЕНИЙ

$a=5; b=10; c=3$

