



Архитектура банковских ПОИС.

Информационное обеспечение предметно-ориентированных ИС.



Требования к архитектуре IT финансового учреждения

- ▶ Сочетаемость и взаимодополняемость приложений, ориентированных на пользователей
- ▶ Многократное использование бизнес-сервисов
- ▶ Независимость от набора информационных технологий
- ▶ Автономность (реализуется через независимую эволюцию, масштабируемость, развертываемость).

Для успешного решения поставленных задач развиваются паттерны архитектуры на базе SOA.

SOA - это набор архитектурных принципов, не зависящих от технологий и продуктов.



Основные паттерны архитектуры SOA

- Общая архитектура брокера объектных запросов (CORBA).
- Веб-сервисы.
- Очередь сообщений.
- Сервисная шина предприятия (ESB).
- Микросервисы.



Общая архитектура брокера объектных запросов (CORBA)

Один из первых стандартов распределенных вычислений (разные технологии, разные компьютеры, разные ОС...). Стандарт обеспечивает:

- ▶ Не зависящие от платформы вызовы удалённых процедур (Remote Procedure Call).
- ▶ Транзакции (в том числе удалённые!).
- ▶ Безопасность.
- ▶ События.
- ▶ Независимость от выбора языка программирования.
- ▶ Независимость от выбора ОС.
- ▶ Независимость от выбора оборудования.
- ▶ Независимость от особенностей передачи данных/связи.
- ▶ Набор данных через язык описания интерфейсов (Interface Definition Language, IDL).



Недостатки CORBA, приведшие к эволюции паттерна

- ▶ **Независимость от местоположения:** клиентский код не имеет понятия, является ли вызов локальным или удалённым. Звучит неплохо, но длительность задержки и виды сбоев могут сильно варьироваться. Если мы не знаем, какой у нас вызов, то приложение не может выбрать подходящую стратегию обработки вызовов методов, а значит, и генерировать удалённые вызовы внутри цикла. В результате вся система работает медленнее.
- ▶ **Сложная, раздутая и неоднозначная спецификация:** её собрали из нескольких версий спецификаций разных вендоров, поэтому (на тот момент) она была раздутой, неоднозначной и трудной в реализации.
- ▶ **Заблокированные каналы связи (communication pipes):** используются специфические протоколы поверх TCP/IP, а также специфические порты (или даже случайные порты). Но правила корпоративной безопасности и фаерволы зачастую допускают HTTP-соединения только через 80-й порт, блокируя обмены данными CORBA.



Эволюция SOA в веб-сервисы

► **Web-сервис (служба)** – программа, которая организует взаимодействие между сайтами. Информация с одного портала передается на другой.

Для эффективной работы веб-сервиса необходима работа трех служб:

- Поставщик услуг (реализует услугу и делает ее доступной через HTTP)
- Провайдер услуг (любой потребитель веб-службы, открывает сетевое соединение и отправляет XML-запрос)
- Реестр служб (центральный каталог, где разработчики публикуют новые службы, или находят уже существующие, клиринговый центр).

Нуждики и решалки веб-сервисов

- **Нужен был надёжный канал связи, поэтому:**

- HTTP стал по умолчанию работать через порт 80.
- Для обмена сообщениями начали использовать платформенно-независимый язык (вроде XML или JSON).

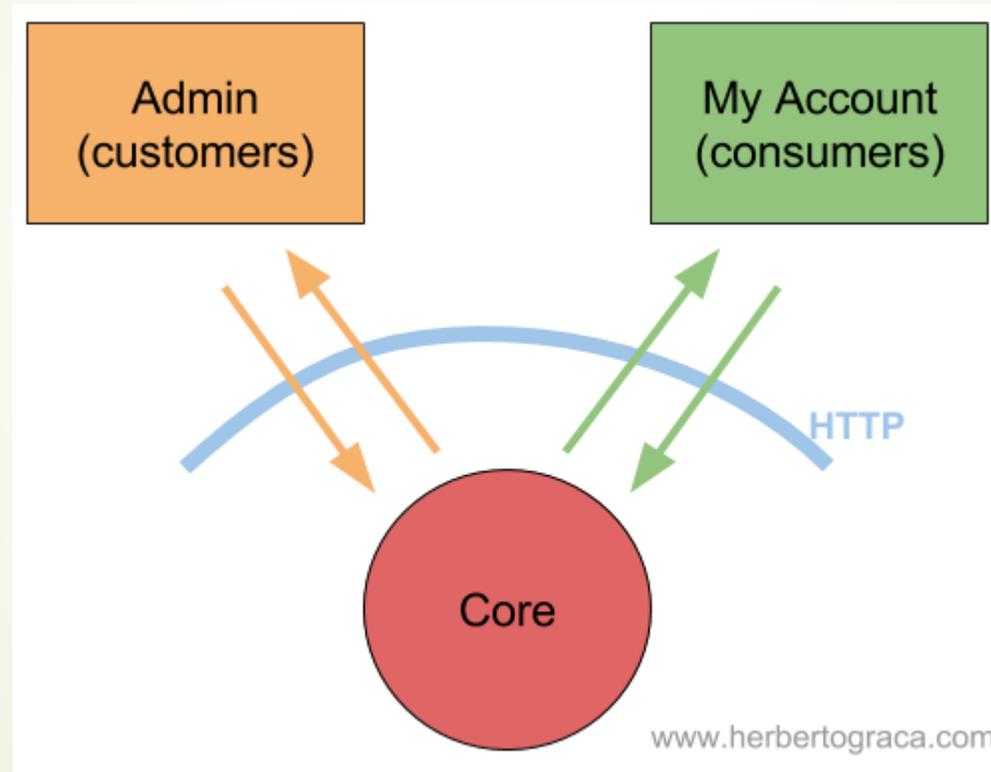
- **Нужно было уменьшить количество удалённых обращений, поэтому:**

- Удалённые соединения стали явными, так что теперь мы всегда знаем, когда выполняется удалённый вызов.
- Вместо многочисленных удалённых вызовов объектов мы обращаемся к удалённым сервисам, но гораздо реже.

- **Нужно было упростить спецификацию обмена сообщениями, поэтому:**

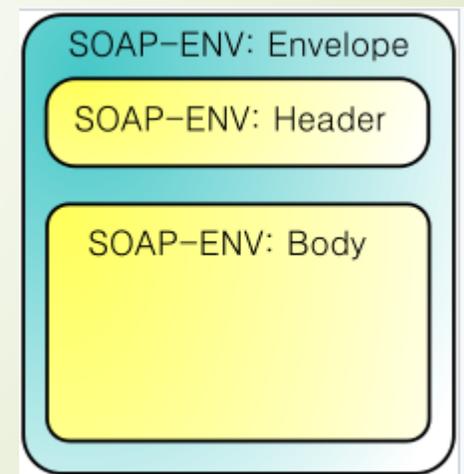
- Первый черновик SOAP появился в 1998-м, стал рекомендацией W3C в 2003-м, после чего превратился в стандарт. SOAP вобрал в себя некоторые идеи CORBA, вроде слоя для обработки обмена сообщениями и «документа», определяющего интерфейс с помощью языка описания веб-сервисов (*Web Services Description Language, WSDL*).
- Facebook разработал GraphQL в 2012-м, а публичный релиз выпустил в 2015-м. Это язык запросов для API, позволяющий клиенту строго определять, какие данные сервер должен ему отправить, не больше и не меньше.

Типовое информационное окружение веб-сервисов



SOAP (*Simple Object Access Protocol*)

- ▶ Базовый стандарт веб-сервисов.
- ▶ Протокол обмена структурированными сообщениями в распределённой вычислительной среде.
- ▶ SOAP может использоваться с любым протоколом прикладного уровня: SMTP, FTP, HTTP, HTTPS и др. Чаще всего SOAP используется поверх HTTP.



Пример SOAP запроса

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <getProductDetails xmlns="http://warehouse.example.com/ws">  
      <productID>12345</productID>  
    </getProductDetails>  
  </soap:Body>  
</soap:Envelope>
```

Версия XML

Упаковка

Тело

Внутренние теги с запросами

Закрытие тела

Запечатывание упаковки

Пример SOAP отклика

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productID>12345</productID>
        <productName>Стакан граненый</productName>
        <description>Стакан граненый. 250 мл.</description>
        <price>9.95</price>
        <currency>
          <code>840</code>
          <alpha3>USD</alpha3>
          <sign>$</sign>
          <name>US dollar</name>
          <accuracy>2</accuracy>
        </currency>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

Учебник по SOAP на русском: <https://www.w3.org/2002/07/soap-translation/russian/part0.html>