## Стандарты доступа к сервисам.

Веб-сервисы основаны на интернет-стандартах, разработанных консорциумом World Wide Web Consortium (W3C) путем консенсуса между различными заинтересованными сторонами. Это позволяет обеспечить открытость стандартов и независимость от какой-либо отдельной коммерческой организации.

Ключевыми стандартами, определяющими веб-сервисы, являются:

* SOAP (Simple Object Access Protocol) – стандарт для обмена сообщениями между приложениями с использованием протоколов передачи HTTP и HTTPS.
* WSDL (Web Services Description Language) – язык описания интерфейсов веб-сервисов.
* UDDI (Universal Description, Discovery and Integration) – стандарт для индексации, поиска и обнаружения описаний веб-сервисов.

Веб-сервисы размещаются на серверах приложений и делаются доступными посредством HTTP GET, HTTP POST и HTTP SOAP. Интерфейсы веб-сервисов описываются с помощью языка WSDL и хранятся либо непосредственно на сервере, либо в специальных XML-хранилищах. Для описания типов данных в WSDL-документах могут использоваться ссылки на отдельные XSD-schema. Регистрация и поиск веб-сервисов осуществляется через бизнес-каталоги на основе стандарта UDDI.

## Стандарт SOAP, протокол SOAP

**SOAP** (Simple Object Access Protocol) — протокол простого доступа к объектам. SOAP основан на двух компонентах: HTTP и XML

Предшественник SOAP – это технология XML-RPC, представляющая собой простую спецификацию для вызова удалённых процедур через RPC с использованием XML для кодирования данных и HTTP в качестве транспортного протокола.

SOAP содержит заголовок, тело сообщения и может включать вложения.

SOAP позволяет веб-сервисам взаимодействовать друг с другом независимо от языков программирования и операционных систем, используя стандартизованный формат обмена данными.

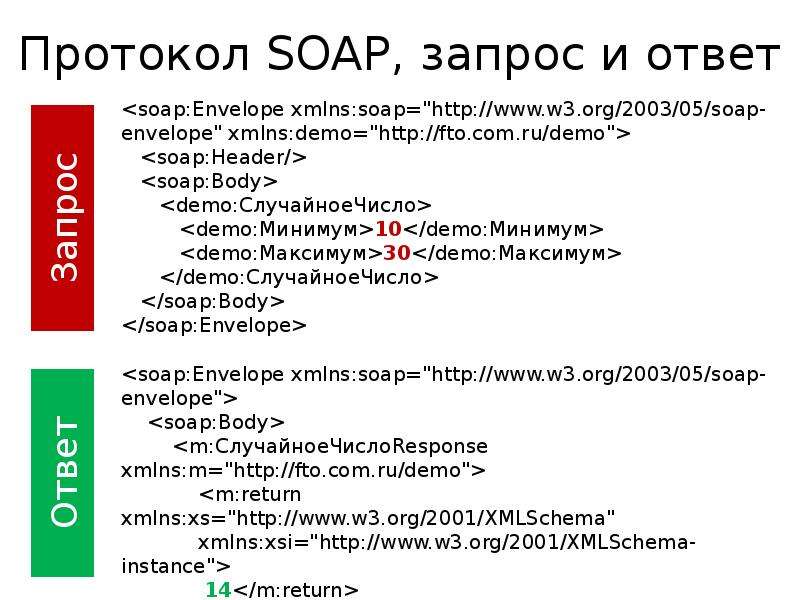


Рисунок 1.1 – Структура SOAP-сообщения

Как видно из рисунка, в запрос входят Envelope, Header, Body и Fault.

1. Envelope – корневой элемент, определяет сообщение и пространство имен, используемое в документе
2. Header – содержит атрибуты сообщения, например: информация о безопасности или о сетевой маршрутизации
3. Body – содержит сообщение, которым обмениваются приложения
4. Fault – необязательный элемент, который предоставляет информацию об ошибках, которые произошли при обработке сообщений

## Форматы XML и JSON

**XML** (eXtensible Markup Language) — язык разметки данных, позволяющий хранить и передавать структурированную информацию.

XML-документ содержит корневой элемент, который является родительским для всех остальных элементов. Элементы формируют иерархическую структуру в виде дерева, которое начинается с корневого элемента и разветвляется на нижележащие уровни. Каждый элемент может содержать атрибуты и вложенные элементы. XML позволяет авторам документов определять собственные теги.

**JSON** (JavaScript Object Notation) — формат обмена данными, основанный на объектной модели JavaScript. Обычно используется для передачи данных между веб-сервером и клиентом. Данные представляются в виде коллекций именованных свойств в формате, аналогичном объектам JavaScript. Это делает JSON компактным и удобным для передачи по сети.

XML более подходит для хранения данных, тогда как JSON лучше для передачи данных, так как имеет более компактный формат.

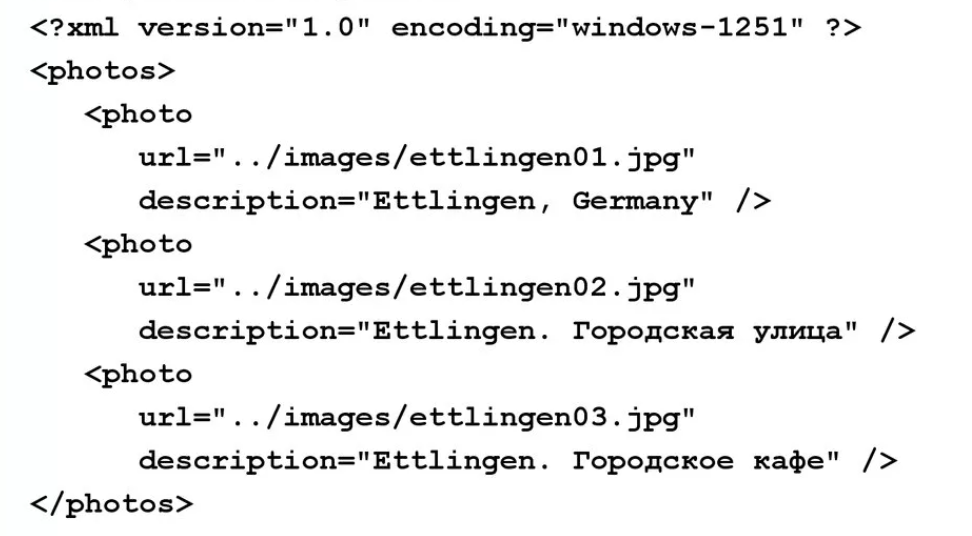


Рисунок 1.2 – XML-документ

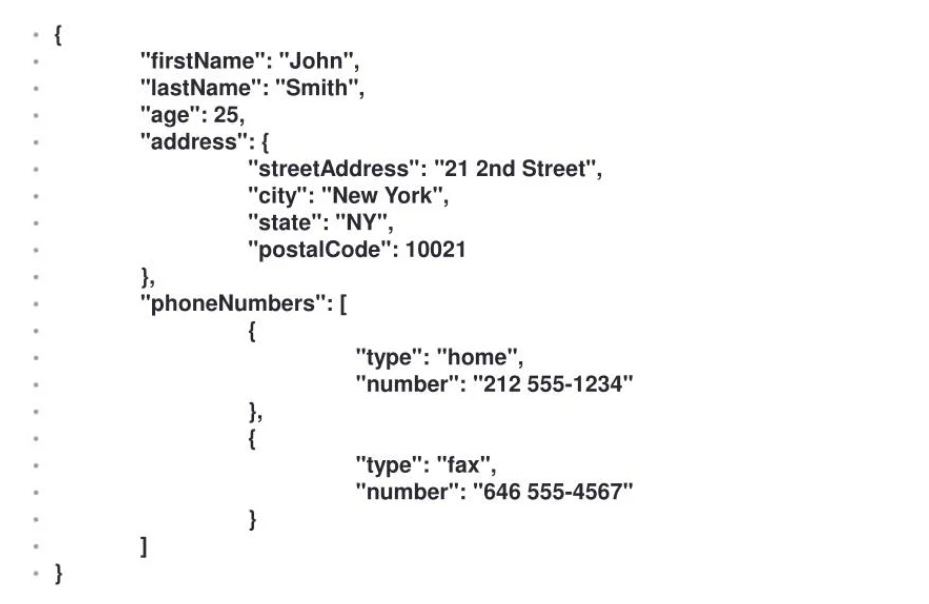


Рисунок 1.3 – JSON-документ

## Спецификация WDDX. Язык описания сервисов WSDL, REST и PRC

**WDDX** (Web Distributed Data eXchange) – технология обмена данными между приложениями в сети, независимая от языка программирования и платформы. Представляет собой спецификацию DTD XML 1.0, которая определяет формат представления данных, включая примитивные типы и более сложные структуры. Существуют библиотеки поддержки WDDX в основных языках программирования.

**WSDL** (Web Services Description Language) основан на XML. В 2007 году была утверждена версия WSDL 2.0 стандартом W3C, однако до сих пор широко используется WSDL 1.1 для описания SOAP-сервисов. WSDL 1.1 имеет сложную структуру, затрудняющую восприятие. На рисунке 1.4 представлена логическая схема WSDL, а на 1.5 – пример WSDL-описания.

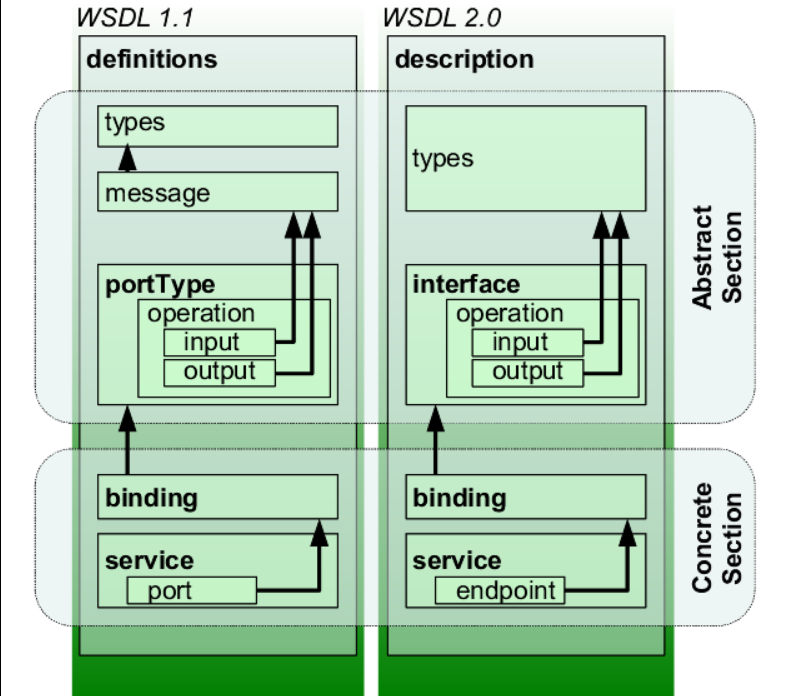


Рисунок 1.4 – Логическая схема WSDL

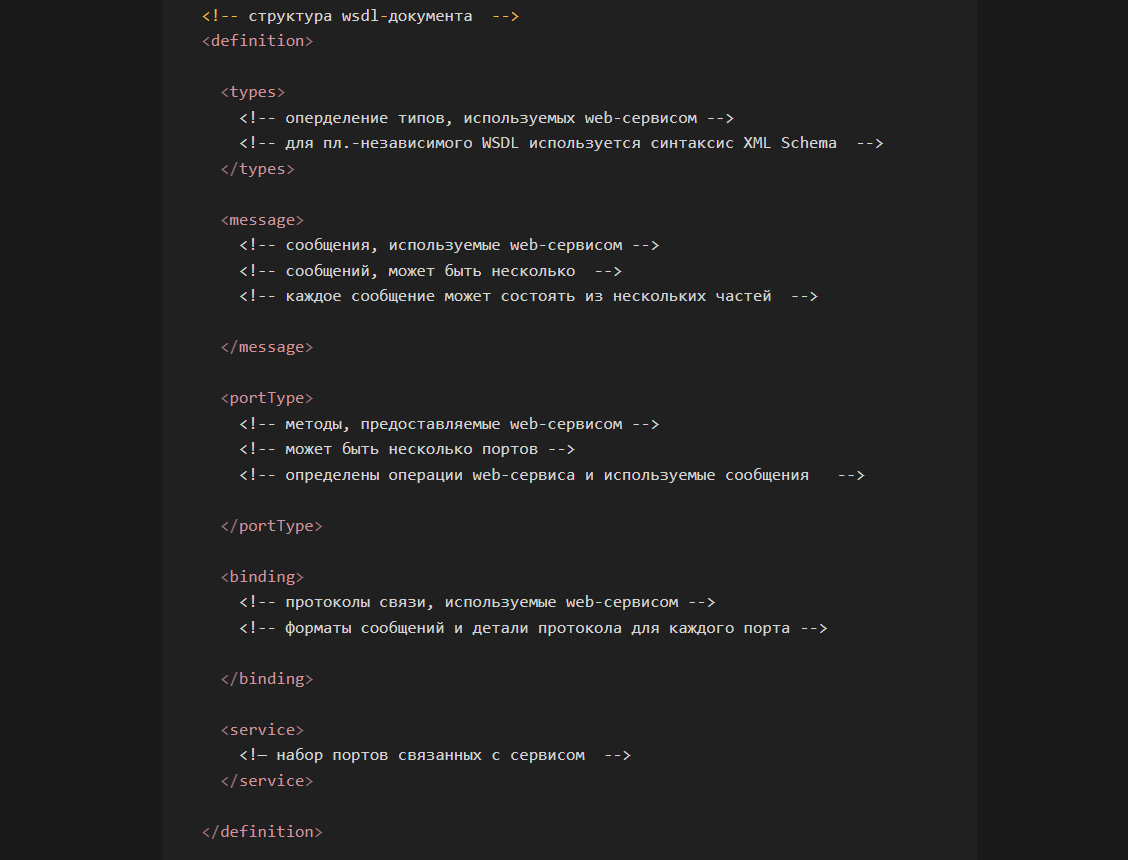


Рисунок 1.4 – Пример WSDL

**REST** (Representational State Transfer) – стиль архитектуры распределённых систем, таких как World Wide Web. RESTful-системы используют гибкий интерфейс управления информацией с помощью глобальных идентификаторов (URL). Для каждой единицы ресурса определены методы HTTP-запросов: POST, PUT, DELETE, GET.

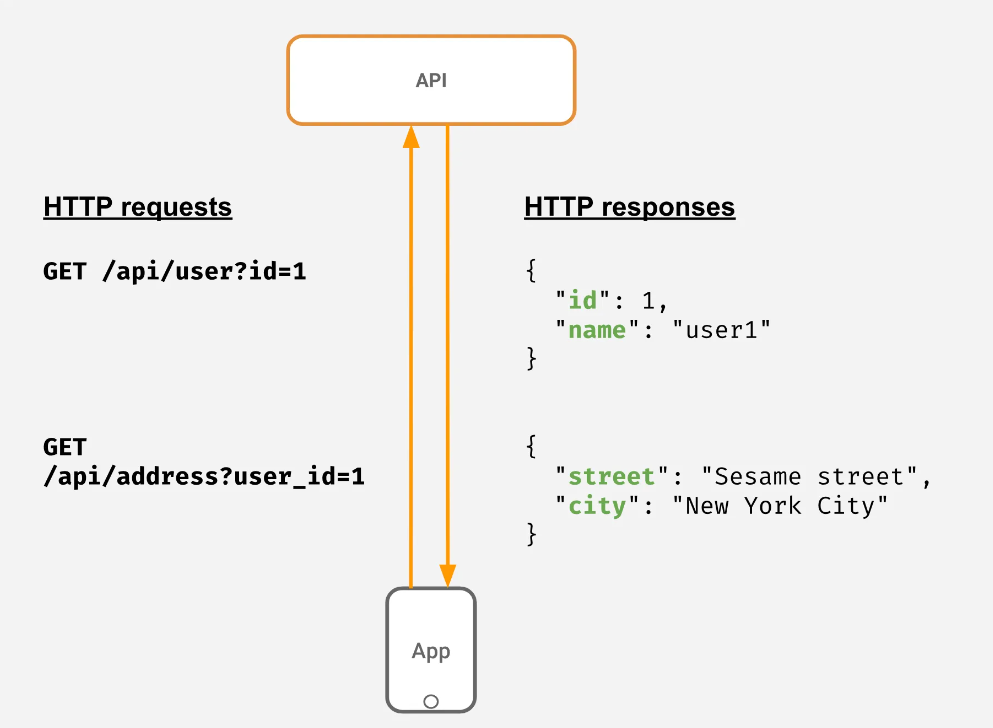


Рисунок 1.5 – Пример REST

У REST есть 6 ограничений:

* 1. Модель клиент-сервер
  2. Отсутствие состояния
  3. Кэширование
  4. Единообразие интерфейса
     + Идентификация ресурсов
     + Манипуляция ресурсами через представление
     + «Самоописываемые» сообщения
     + Гипермедиа как средство изменения состояния приложения (HATEOAS)
  5. Слои
  6. Код по требованию (необязательное ограничение)

**RPC** (Remote Procedure Call) — механизм вызова удалённых процедур, позволяющий вызывать процедуры, находящиеся на удалённом компьютере, так, как если бы они выполнялись локально.

RPC является одним из основных технических подходов для разработки распределённых вычислительных систем. Он позволяет объединять программы, написанные на разных языках программирования и работающие в различных операционных системах.

Обычно для реализации RPC используется клиент-серверная модель взаимодействия. Клиент формирует запрос на выполнение удалённой процедуры, включая имя процедуры и передаваемые аргументы. Запрос отправляется на сервер, где процедура выполняется. Результаты и возможные ошибки возвращаются клиенту.

В качестве транспортного протокола часто используется HTTP. Также для кодирования сообщений могут применяться форматы данных XML/JSON.

RPC является базовой технологией для построения распределённых вычислений и веб-сервисов в частности. В частности, он лёг в основу разработки протокола SOAP.

На рисунке 1.6. представлен пример RPC-запроса.

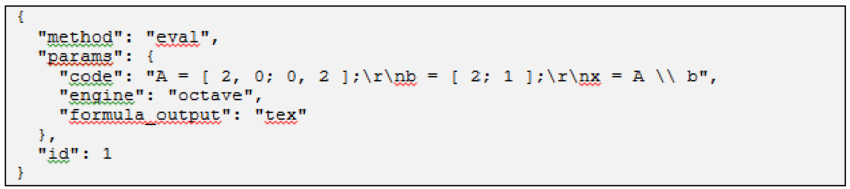


Рисунок 1.6 – Пример RPC

## Выводы

В ходе практической работы были изучены основные принципы и стандарты, лежащие в основе веб-сервисов. Были рассмотрены ключевые технологии, такие как SOAP, WSDL, UDDI, которые позволили определить набор стандартов, необходимых для разработки и использования веб-сервисов. Были проанализированы особенности форматов обмена данными XML и JSON, а также сопоставлены их возможности.

На основании изучения механизмов RPC и REST были сделаны выводы об основных подходах к разработке распределенных приложений и веб-сервисов. Путем анализа основных концепций, заложенных в стандарте WSDL, удалось получить представление о способах описания интерфейсов веб-сервисов.

В результате проведенного изучения была сформирована общая картина о структуре и принципах построения веб-сервисов на основе рассмотренных интернет-стандартов и технологий. Были сделаны выводы о роли каждой из них в обеспечении взаимодействия между распределенными компонентами.