1. **Поясните понятия REST и RPC.**

SOAP более применим в сложных архитектурах, где взаимодействие с объектами выходит за рамки теории CRUD, а вот в тех приложениях, которые не покидают рамки данной теории, вполне применимым может оказаться именно REST ввиду своей простоты и прозрачности. Действительно, если любым объектам вашего сервиса не нужны более сложные взаимоотношения, кроме: «Создать», «Прочитать», «Изменить», «Удалить» (как правило — в 99% случаев этого достаточно), возможно, именно REST станет правильным выбором. Кроме того, REST по сравнению с SOAP, может оказаться и более производительным, так как не требует затрат на разбор сложных XML команд на сервере (выполняются обычные HTTP запросы — PUT, GET, POST, DELETE). Хотя SOAP, в свою очередь, более надежен и безопасен.

REST - это парадигма организации API, которая подразумевает (помимо прочего) четкое разбиение на ресурсы и вызов операции с помощью конкретного HTTP-метода.

REST — это не стандарт и не спецификация, а архитектурный стиль, выстроенный на существующих, хорошо известных и контролируемых консорциумом W3C стандартах, таких, как HTTP, URI (Uniform Resource Identifier), XML и RDF (Resource Description Format). В REST-сервисах акцент сделан на доступ к ресурсам, а не на исполнение удаленных сервисов; в этом их кардинальное отличие от SOAP-сервисов.

И все же удаленный вызов процедур применим и в REST. Он использует методы PUT, GET, POST, DELETE HTTP протокола для манипуляции объектами. Кардинальное отличие его от SOAP в том, что REST остается HTTP-запросом.

1. **Поясните понятие «ASMX-сервис».**

ASMX предоставляет возможность создавать веб-службы, которые отправляют сообщения с помощью протокола SOAP. SOAP — это независимый от платформы и не зависящий от языка протокол для создания веб-служб и доступа к ним

1. **Какой протокол применяется при взаимодействии клиента с ASMX-сервисом?**

*SOAP*

1. **Что такое SOAP?**

*SOAP — это независимый от платформы и не зависящий от языка протокол для создания веб-служб и доступа к ним.*

Сообщение SOAP — это XML-документ, содержащий следующие элементы:

* Корневой элемент с именем конверт , ОПРЕДЕЛЯЮЩИЙ XML-документ как сообщение SOAP.
* Необязательный элемент заголовка , содержащий сведения, относящиеся к приложению, такие как данные проверки подлинности. Если элемент Header представлен, он должен быть первым дочерним элементом элемента конверта .
* Обязательный элемент Body, СОДЕРЖАЩИЙ сообщение SOAP, предназначенное для получателя.
* Необязательный элемент fault , который используется для указания сообщений об ошибках. Если элемент fault имеется, он должен быть дочерним элементом элемента Body .

SOAP может использовать несколько транспортных протоколов, включая HTTP, SMTP, TCP и UDP. Однако служба ASMX может взаимодействовать только по протоколу HTTP. Платформа Xamarin поддерживает стандартные реализации SOAP 1,1 по протоколу HTTP и включает поддержку многих стандартных конфигураций службы ASMX.

1. **Что такое Envelop?**

Элемент SOAP сообщения **ENVELOPE**определяет начало и конец сообщения. Именно благодаря этому элементу, клиент, принимающий сообщение знает, когда сообщение полностью получено. Другими словами, с помощью этого элемента мы можем узнать, когда сообщение получено и готово к обработке.

1. **Что такое WSDL?**

WSDL — это язык описания веб-сервиса, имеющий структуру XML. Основное назначение WSDL-файла — это интерфейс доступа к функциям сервиса, возвращаемым типам данных; путь к серверу, обрабатывающему запросы и т.д.

WSDL разбивает веб-службы на три определенных идентифицируемых элемента, которые могут быть объединены или использованы повторно после определения.

Три основных элемента WSDL, которые могут быть определены отдельно:

* Типы
* операции
* переплет

Документ WSDL имеет различные элементы, но они содержатся в этих трех основных элементах, которые можно разрабатывать как отдельные документы, а затем их можно объединять или повторно использовать для формирования полных файлов WSDL.

<!-- структура wsdl-документа -->

<definition>

<types>

<!-- оперделение типов, используемых web-сервисом -->

<!-- для пл.-независимого WSDL используется синтаксис XML Schema -->

</types>

<message>

<!-- сообщения, используемые web-сервисом -->

<!-- сообщений, может быть несколько -->

<!-- каждое сообщение может состоять из нескольких частей -->

</message>

<portType>

<!-- методы, предоставляемые web-сервисом -->

<!-- может быть несколько портов -->

<!-- определены операции web-сервиса и используемые сообщения -->

</portType>

<binding>

<!-- протоколы связи, используемые web-сервисом -->

<!-- форматы сообщений и детали протокола для каждого порта -->

</binding>

</definition>