**Questions/Answers for PM3**

1. **When should we use REST and when should we use SOAP?**

REST is used in scenarios when one wanted to use only HTTP as a communication protocol. REST is usually preferred for services that are exposed as public APIs. If you are writing code in a scripting language such as JavaScript, Python, Ruby or an older language such as VBA, or you are concerned about the size/format of data being sent, then REST is for you. REST is particularly useful for restricted-profile devices such as mobile and PDAs for which the overhead of additional parameters like headers and other SOAP elements are less.

SOAP can make use of any transport protocol not just HTTP. When there is a major security concern SOAP can be used. SOAP is preferred in enterprise environment where APIs are used locally and does not contain any public calls. If you are writing code in a compiled language (Java, C#) that has a strong SOAP library in place and you don't want to worry about how the data gets sent to/from, then SOAP may be the easiest option since all the SOAP functions become standard functions in the library being used. If you are having an application that involves transaction management, SOAP is always a better option, providing support for Atomic Transactions. One should use SOAP when publishing a complex API to the public.

1. **REST vs SOAP**

**REST**

1. REST is an architectural style.
2. In REST client has a specific list of functions exposed to them.
3. REST has better performance and scalability. REST reads can be cached.
4. REST is not a protocol but an architectural style.
5. REST uses URI to expose business logic.
6. REST does not define too much standards like SOAP.
7. RESTful web services inherits security measures from the underlying transport.

**SOAP**

1. SOAP is a protocol.
2. The biggest benefit of SOAP is that it provides a complete end-to-end RPC style interface with functions and types that are fully discoverable.
3. SOAP based reads cannot be cached.
4. SOAP is a XML based messaging protocol.
5. SOAP uses services interfaces to expose the business logic.
6. SOAP defines standards to be strictly followed.
7. SOAP defines its own security.

**REFERENCE**

<http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/>

<http://www.javatpoint.com/soap-vs-rest-web-services>

1. **Overview of SOA**
2. **How to implement web services using servlets?**