**SOA, ARCHITECTING WEB SERVICES, TECHNOLOGY STACK**

**The Service-Oriented Architecture (SOA)**

In order for Web Services to be able to work well together, they must participate in a set of shared organizing principles we call a service-oriented architecture (SOA). The term service-oriented means that the architecture is described and organized to support Web Services’ dynamic, automated description, publication, discovery, and use.

The SOA organizes Web Services into three basic roles: the service provider, the service requester, and the service registry. The relationships among these three roles are shown in Figure.

Architecting Web Services:

One established model for how architects visualize the systems before them is the 4+1 View Model of Software Architecture, popularized by Philippe Kruchten of Rational Software. Whereas the four blind men each touch the elephant in a different place and therefore come to different understandings of it, the architect has clear vision, seeing the elephant from all four views. As a result, the architect has a comprehensive picture of the elephant.

This is the same with the 4+1 View Model. This model describes four distinct ways of looking at the architecture for a system, plus a fifth view that overlaps the others, as shown in Figure.

---

**The Web Services Technology Stack:**

The base stack includes those technologies necessary to create and invoke Web Services. At the bottom is the network layer, which fundamentally allows Web Services to be available to service requesters. Although HTTP is the de facto standard network protocol, the architect may consider any of a number of other options, including SMTP (for email), FTP, IIOP, or messaging technologies such as MQ. Some of these choices are:

[Diagram of network stack and its components]
request/response based, whereas others are message based; furthermore, some are synchronous, whereas others are asynchronous. The architect may find that in a large system, a combination of different network protocols is appropriate.

On top of the SOAP layer comes three layers that together form the service description. WSDL is the de facto standard for service descriptions, with the addition of the still-tentative WSEL for endpoint descriptions. The service interface definition contains the binding, portType, message, and type elements, which form the portion of the service description that is reusable from one implementation to another.

The service implementation definition, however, contains those elements that are specific to each implementation: the service and port elements. A third party (say, a standards body) might specify the service interface definition for a particular type of Web Service, leaving the service implementation definition up to each implementation team.

Next comes the endpoint description, which introduces semantics to the service descriptions that apply to a particular implementation. Endpoint descriptions can contain security, QoS, and management attributes that help to define the policies for each of these vertical columns.

Once the architect has dealt with all the issues in the base stack, the Web Services are essentially fully constructed. Next, the development team uses UDDI to publish the services to a registry or another repository of information about available Web Services. Once Web Services are published, UDDI can then be used to discover them in the registries.