

# SERVICE ORIENTED ARCHITECTURE

## 3.1 Service-Oriented Architecture

A service-oriented architecture is intended to define loosely coupled and interoperable services/applications, and to define a process for integrating these interoperable components.

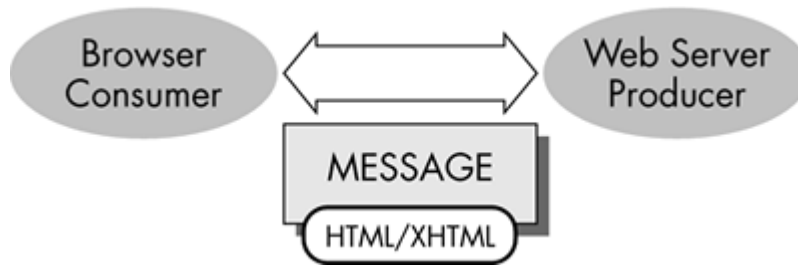
In SOA, the system is decomposed into a collection of network-connected components. Applications and resources within a service-oriented architecture should not be built as a tightly coupled monolithic model. Rather, these applications are composed dynamically from the deployed and available services in the network.

These services are dynamically discovered and bound together in a loosely coupled manner. This dynamic construction allows the system to quickly respond to new business requirements. Because these applications are not tightly coupled to their components, they provide increased stability and availability to users, reacting to component failure by finding and binding to equivalent components without requiring human intervention.

The most commonly understood SOA architectures are Web and Web services. The Web architecture has evolved even before we started talking about the SOA. On careful examination of the interaction model exhibited by the Web, it is inherent that it is a medium for message exchange between a consumer and producer using a common, well understandable, and interoperable data model, HTML/XHTML. There is no tight coupling that exists between the client and service.

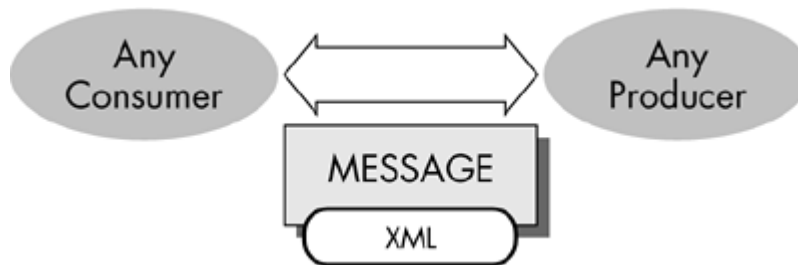
The above model (Figure 3.1) was adequate for the common user agent-based interaction. But when the interaction pattern becomes complex, such as business-to-business, the above Web architecture model needs more polished message exchange patterns to adapt to any user agent and/or applications of choice. There comes the concept of a more generalized solution, Web services. The Web service architecture extends the above interaction pattern further by adding the power and expressiveness of XML.

**Figure 3.1. Messaging interaction between the Web browser end user and the services producer.**



As we can see in Figure 3.2, the exchange messages centered on XML and the interaction pattern have evolved to an any-to-any scenario. This flexible interaction pattern using XML messages as the core data format increases the value of SOAs. This produces the interaction request “response patterns, asynchronously, for better interoperability between consumers and any of its producers. This is a very loosely coupled system.

**Figure 3.2 Messaging interactions between the end user and the producer.**



In order to better align with SOA principles, this architecture is polished with more emphasis on standard definitions of the service description, registration, discovery, and interactions. Figure 3.3 shows a commonly used interaction pattern.

**Figure 3.3. The SOA interaction pattern.**

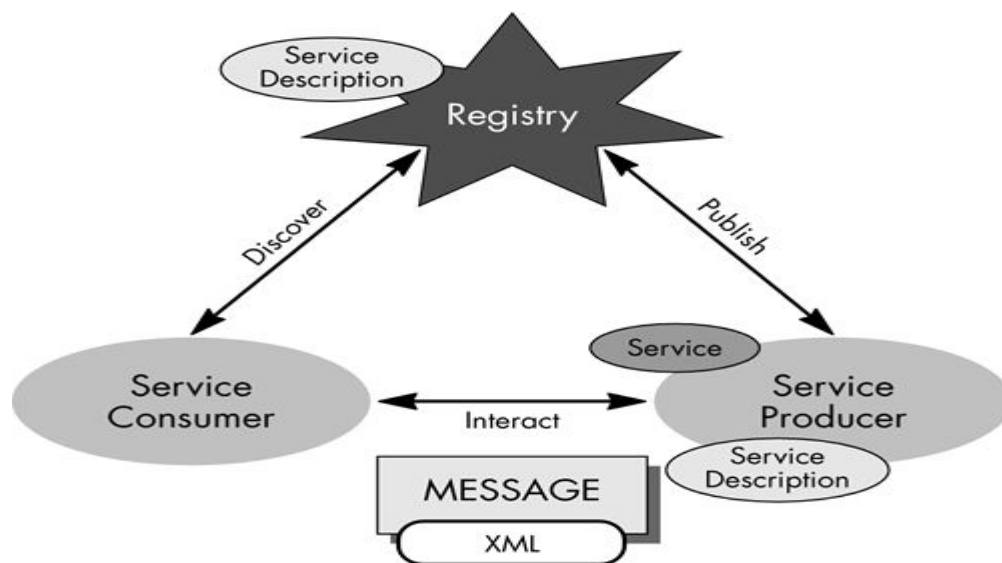


Figure 3.3 shows the following concepts:

- A service provider creates a service for interaction and exposes the service's description for the consumers with the necessary message format and transport bindings.
- The service provider may decide to register this service and its description with a registry of choice.
- The service consumer can discover a service from a registry or directly from the service provider and can start sending messages in a well-defined XML format that both the consumer and service can consume.

This is a simple yet powerful concept of providing message interactions. Each of these interactions can be further broken down into different layers. As an example, the interaction pattern between the consumer and the service is a combination of message, message packaging, and transport protocols. This architecture needs to further extend by adding QoS requirements, security, and management aspects. We will cover these details in the coming pages. Let us now open discussion on the Web services architecture as our example candidate to explore SOA.

Source : <http://elearningatria.files.wordpress.com/2013/10/ise-viii-grid-computing-06is845-notes.pdf>