

SOA

Service-Oriented Architecture is a component model whose goal is to achieve loose coupling^[2] between interacting software agents. This architecture consists of different functional units called “services”. A loosely coupled system has the benefit of being agile and able to survive noticeable changes in the service's internal structure and its implementation. This feature makes SOA very convenient for business dynamic environment and on-demand business, and thus for TrustCoM, where virtual organizations (discussed later) are to be dynamic, created on-fly, and can live for one transaction or more.

Achieving loose coupling in SOA is done through two architectural constraints:

1. A small set of simple and ubiquitous interfaces to all participating software agents, through defining only generic semantics at the interfaces to provide universality.
2. Descriptive messages constrained by an extensible schema, this allows no, or minimal system behavioral prescribing. The schema is for both limiting structure of the messages and allows defining new versions by using its extensibility property.

SOA also provides discovery services that aids to discover available services and their description in a system. Other constraints are also applied to SOA to provide better scalability, performance and reliability [SOA1]. for this, two types of services exist, stateless services, where all necessary information are included in the message, so the provider don't have to store state information. This improve scalability-- “services in mass production” -- where a request is seen as generic, and visibility, since monitoring is easier, one request only to deal with, and no intermediate results to track which makes recovery simpler.

The other type is stateful services, these are when a session is to be created for efficiency reasons. This necessitates storing some consumer-specific information, which could effect the scalability property.

SOA is enabled and defined by WSA (Web services Architecture). In WSA, services interacts using XML (Extensible Markup Language) messages. WSA is mainly characterized by: [SOA5]

1. Based on XML technologies (e.g. XML Information Model [SOA2], XML Base, and XML schema [SOA3])
2. Independency of underlying transport protocols(e.g. HTTP, SMTP)
3. Use XML messages for exchanges, these messages provides extensibility to adapt to different transaction requirements (reliability, security,...) These messages are called SOAP messages (Service-Oriented Architecture Protocol messages) [SOA4]
4. Capabilities of services are *published* using a description language like WSDL (Web Services Description Language)
5. Built on XML capabilities, WSA uses *service discovery*, transaction or state management, workflow choreography.

A Service model in Nutshell

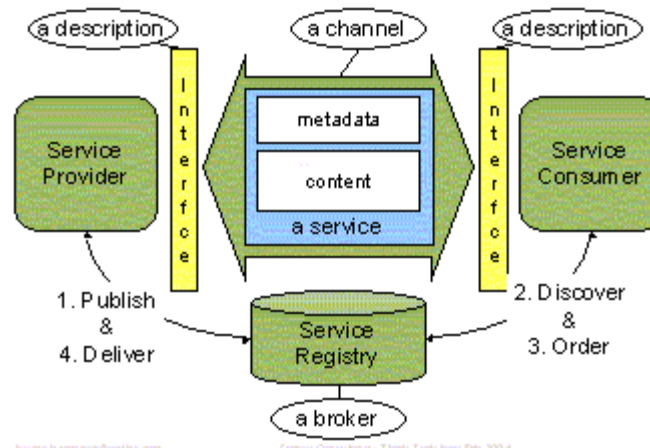


Figure 1

The above figure shows a service model, it consists of:

Service -- a software entity designed to support the characteristics of WSA mentioned above. Its a software entity that has an interface(s)¹. It uses SOAP messages (maybe asynchronously also) in its interactions. A service is an abstraction made concrete by developing a software agent.

Service Provider – the software entity that implements a service specification or description.

Service Consumer – Also called requestor, its the software entity client that calls a service provider to use a service. This can be a user application or another service.

¹An interface can be common to many services, and one service can have many interfaces (many-to-many relation)

Service Registry – A software entity that acts a service locator. It implements the discovery and order functions for the requestor for a specific service, and where new services are published and delivered.

Service Broker – A special service that can pass service requests to other service providers.

The service provider, consumer (requestor), and registry are the main roles in an SOA.

In addition, SOA implements three operations, these operations define the contracts between roles. [SOA6]

- *Publish*: Is an operation that acts as service registration or advertisement. Its operated between the service registry and service provider.
- *Find (Discover)*: Its the complementary operation of *Bind*, because services are published so they can be found. It's the contract between a service requestor and service registry. The find operation is executed on the registry according to a search criteria specified by the requestor. A search criteria might be the type of service, QoS (Quality of Service), etc...
- *Bind*: This operation binds both the service provider and requestor in a client-server like relationship. This relationship can be dynamic or static. Dynamic such as dynamic generation of client-side proxy. While static, the developer hand-codes the way of invoking the service to the client.

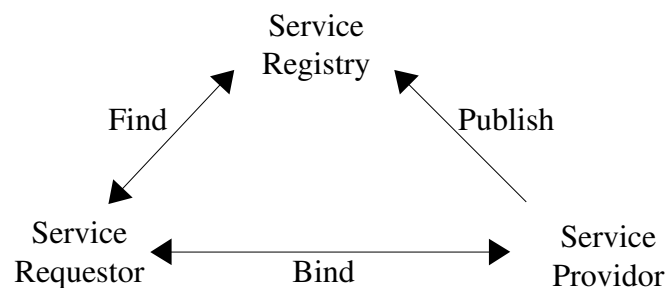


Figure 2 Service-Oriented Architecture -- Operations

2.1 Web Services

While they are related, but in concept, they are distinct. SOA and web services are commonly viewed together. SOA is an architectural concept that focuses on building loosely coupled set

(pool) of dynamic components. On the other hand, Web services are only a way to build SOA.

According to the W3C Working Group Note, Web Services Architecture, they define a web service as the following: [SOA7]

“[Definition: A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.]”

A web service is an abstraction that is to be implemented through a concrete hardware or software code that sends and receives messages, called an agent. A web service is a set of abstract functionalities that characterizes a resource. A web service can be implemented with one or more agents (e.g. different programming languages). The web services *semantics* (the “WHAT”) describes what is the expected behavior of a certain web service. Also, web service's *mechanics* (the “HOW”) is a description of how the web service is to achieve what is expected. This is usually described in WSDL.

References:

[SOA1] "<http://webservices.xml.com/lpt/a/ws/2003/09/30/soa.html>", "What is Service-Oriented Architecture", Hao He. Accessed on 9th November, 2005.

[SOA2] XML Information Set, WorldWide Web Consortium (February 2004), <http://www.w3.org/TR/xml-infoset/>.

[SOA3] XML Protocol Working Group, WorldWide Web Consortium (2004), <http://www.w3.org/2000/xp/Group/>.

[SOA4] XML Schema, WorldWide Web Consortium, <http://www.w3.org/XML/Schema>.

[SOA5] IBM Systems Journal, Grid Computing, Vol.43, No. 4, 2004. "Evolution of grid computing architecture and grid adoption models", J. Joseph, M. Ernest, and C. Fellenstein, "<http://www.research.ibm.com/journal/sj/434/joseph.html>" accessed on 10th November, 2005.

[SOA6] "Building Web Services with Java", 2nd Edition, Developer's Library, 2004, Chapter 1, pg13-16.

[SOA7] "<http://www.w3.org/TR/ws-arch/>", W3C Working Group, Web services Architecture, accessed on 11th November, 2005