

## Summary of Chapter 18

What is a web service?

- An act or performance offered by one party to another. Although the process may be tied to a physical product, the performance is essentially intangible and does not normally result in ownership of any of the factors of production.
- Loosely coupled, reusable software component that encapsulates discrete functionality, which may be distributed and programmatically accessed.

How different from a component?

- Services are self-contained (perform predetermined tasks) and loosely coupled (for independence)
- Services should not rely on external components that may have different functional and non-functional behaviour

Service oriented systems

Service-oriented systems are a way of developing distributed systems where the system components are stand-alone services, executing on geographically distributed computers.

Benefits of Service Oriented Approach:

- Services can be offered by any service provider inside or outside of an organization.
- The service provider makes information about the service public so that any authorized user can use the service.
- Service users can pay for services according to their use rather than their provision.
- Applications can be made smaller for system that requires limited functionality

Example of a Service Oriented system:

## A real world example(cooks)

Let's assume you have 4 cooks. In SOA, you assume they hate each other, so you strive to let them have to talk to each other as little as possible.

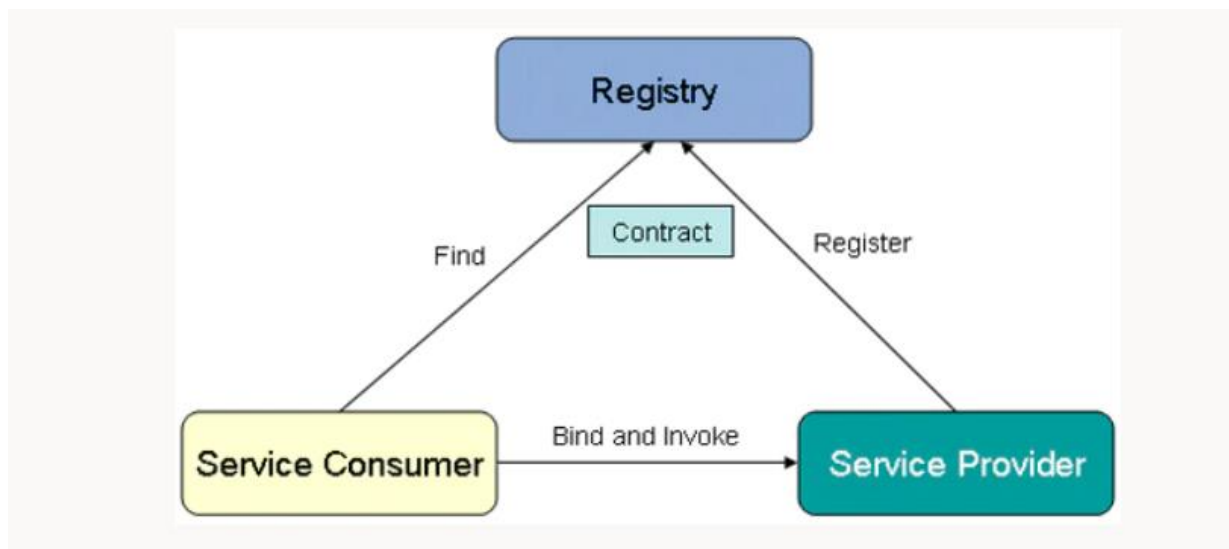
How do you do that? Well, you will first define the roles and interface -- cook 1 will make salad, cook 2 will make soup, cook 3 will make the steak, etc.. Then you will place the dishes well organized on the table (so these are the interfaces) and say, "Everybody please place your creation into your assigned dishes. Don't care about anybody else."

This way, the four cooks have to talk to each other as little as possible, which is very good in software development -- not necessarily because they hate each other, but for other reasons like physical location, efficiency in making decisions etc.

### Service Oriented Architecture :

- Service-oriented architecture (SOA) is an architectural style based on the idea that executable services can be included in application.
- Services have well-defined, published interfaces, and applications can choose whether or not these are appropriate.
- Same service may be available from different providers and that applications could make a runtime decision of which service provider to use

### HOW SOA works :

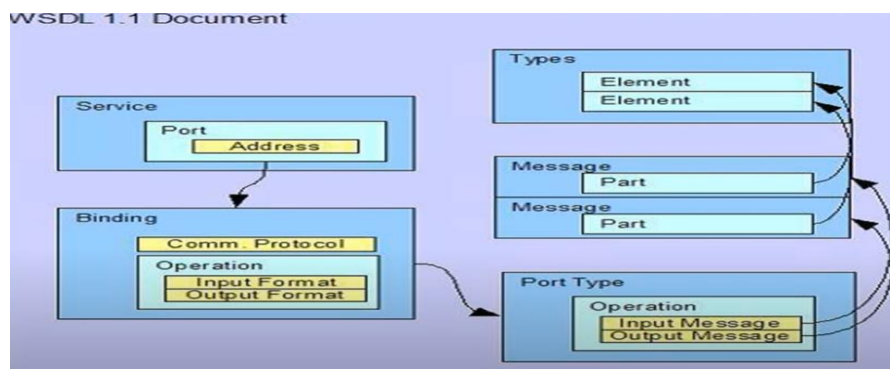


- The service provider creates a web service and possibly publishes its interface and access information to the service registry.
- The service broker, also known as *service registry*, is responsible for making the web service interface and implementation access information available to any potential service requestor. This is also called UDDI.
- 
- The service requestor or web service client locates entries in the broker registry using various find operations and then binds to the service provider to invoke one of its web services.
- Internationally agreed standards is the key to SOA to avoid any incompatibility. The standard is set by W3C.
- The Web Service Description Language (WSDL) is a standard for service interface definition. It sets out how the service operations (operation names, parameters, and their types) and service bindings should be defined.
- A webservice that complies to the SOAP webservice specification by W3C is a SOAP webservice.

The WSDL specification defines three aspects of a Web service:  
what the service does, how it communicates, and where to find it:

1. The “what” part of a WSDL document, called an interface, specifies what operations the service supports and defines the format of the messages sent and received by the service.
2. The “how” part of a WSDL document, called a binding, maps the abstract interface to a concrete set of protocols. The binding specifies the technical details of how to communicate with a Web service.
3. The “where” part of a WSDL document

- What are the things we can do with your application  
[ Addition ]
- What are inputs we should pass  
[ two integers ] → Input message  
Integer 1 → part  
Integer 2 → part
- What is the output we will get  
[ integer sum of two numbers ] → Output message  
Sum Integer → part
- Where should we send the request  
[ endpoint where request to be sent ] → Binding



```
<message name="getSumRequest">
  <part name="number1" type="xs:int"/>
  <part name="number2" type="xs:int"/>
</message>

<message name="getSumResponse">
  <part name="sum" type="xs:int"/>
</message>

<portType name="calculator">
  <operation name="getSum">
    <input message="getSumRequest"/>
    <output message="getSumResponse"/>
  </operation>
</portType>

<binding type="calculator" name="b1">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation soapAction="http://example.com/getSum"/>
    <input><soap:body use="literal"/></input>
    <output><soap:body use="literal"/></output>
  </operation>
</binding>
```

Input message

## RESTFUL API:

### Why Restful API:

- XML-based messages exchanged between services are good for complex services, dynamic service binding, and control over quality of service and service dependability.
- Services were developed, it emerged that most of these were single function services with relatively simple input and output interfaces. XML based applications are not lightweight. Implementing these standards requires a considerable amount of processing to create, transmit, and interpret the associated XML messages.
- This slows down communications between services, and, for high-throughput systems, additional hardware may be required to deliver the quality of service required

## **A SIMPLE LIGHTWEIGHT APPROACH was developed ----→ REST architectural style**

- REST stands for Representational State Transfer.
- It is not a PROTOCOL. It is a guideline
- Simpler method than SOAP/WSDL for implementing web service interfaces
- A fundamental element in a RESTful architecture is a resource. Everything is represented as resource.
- Resources may have multiple representation that is, they can exist in different formats.
- Resources have a unique identifier, which is their Endpoint URL.
- Used with HTTP protocol.
- Here content can be XML, JSON, String, Document.
- Supports Stateless, (no information from the past sessions)
- More Popular nowadays, due to mobile devices which require lightweight applications.
- Supports multiple MIME type

C	Create/POST
R	Retrieve/ Get
U	Update/ PUT
D	Delete/Delete

### **Example of GET operation.**

This would invoke the GET operation and return a list of maximum and minimum temperatures. To request the temperatures for a specific date,

a URL query can be used:

- <http://weather-infoexample.net/temperatures/edinburgh?date=20140226>
- <http://weather-info-example.net/temperatures/boston?date=20140226&country=USA&state=Mass>

Response:

```
{  
  "place": "Boston",  
  "country": "USA",  
  "state": "Mass",  
  "date": "26 Feb 2014",  
  "units": "Fahrenheit",  
  "max temp": 41,  
  "min temp": 29  
}
```

### **Issues of RestFul Approach**

- Complex Interfaces are difficult to design using Restful.
- You have to implement your own infrastructure for monitoring and managing the quality of service.
- No standards for RESTful interface description.

### **URL mapping of the below webservice:**

1. Change a supplier's phone number -----→ PUT

<http://www.example.com/SUPPLIER/12345/>

2. Create a purchase order for a specific supplier -----→ CREATE

<http://www.example.com/SUPPLIER/12345/orders>

3. Delete a supplier -----→ DELETE

<http://www.example.com/SUPPLIER/12345/>

4. Retrieve all data of a specific supplier -----→ GET

[http://weather-info-example.net/companytranscripts/record?supplier="ABC"](http://weather-info-example.net/companytranscripts/record?supplier=)

5. Retrieve all suppliers with a minimum turnover in a time period-----→ GET

[http://weather-info-example.net/suppliers?date=20140226&turnover="minturn"](http://weather-info-example.net/suppliers?date=20140226&turnover=)