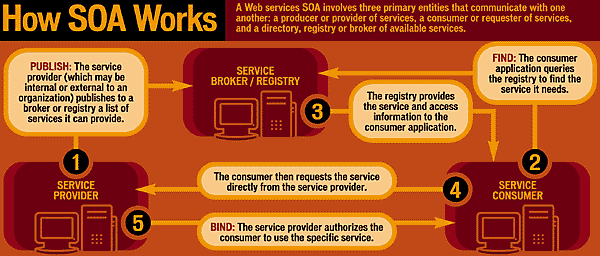
**SOA**

****

**Rameshkartik.RS**

**Table of Contents**

[Introduction 3](#_Toc386019853)

[What is SOA? 3](#_Toc386019854)

[SOA Overview 4](#_Toc386019855)

[SOAP Message 6](#_Toc386019856)

[Summary 9](#_Toc386019858)

**WHAT’s IN THIS ARTicle?**

* What is SOA?
* SOA Overview
* SOAP Message
* Summary

# Introduction

In the earlier days , connecting the systems across the system could be a challenging one until SOA is formed. Yes SOA is the perfect architectural approach for creating a OPEN, and interoperable computing environment. It should not seen just from a technology perspective, but the policies, practices and frameworks by which we ensure the right services are provided and consumed by clients across the platforms.

# What is SOA?

SOA – Service Oriented Architecture, Its an architectural approach in the software development where the application is organized as ‘Services’. Services are a group of methods that contain the business logic to connect DB or other services. Methods have a clearly defined and published so these methods are used by the clients as a black box. So what is a black box? It's nothing but a system or an object which can be viewed in terms of its input, output and transfer characteristics without knowledge of its internal workings. Across the platforms these methods can be accessed, No matter your client developing UI in C# or Java or any latest technology. It decouples the business services from the technical services, means the service methods having the business logic is not coupled with the particular programming language , Both will reacts independently.

# SOA Overview

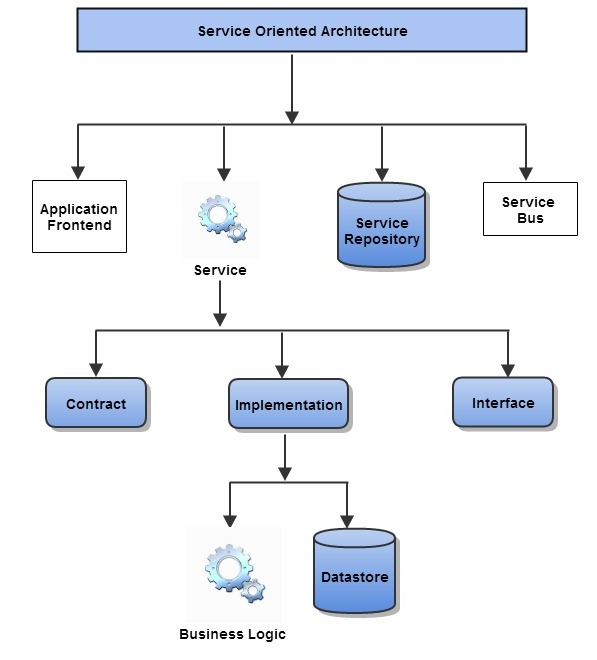
A Service oriented architecture is based on four key abstractions

* An Application Front End
* A Service
* A Service Repository
* A Service Bus

*Application Front End* – Application front end is decoupled from the services. Each service has a contract that defines what it will do, and one or more interfaces to implement the contract

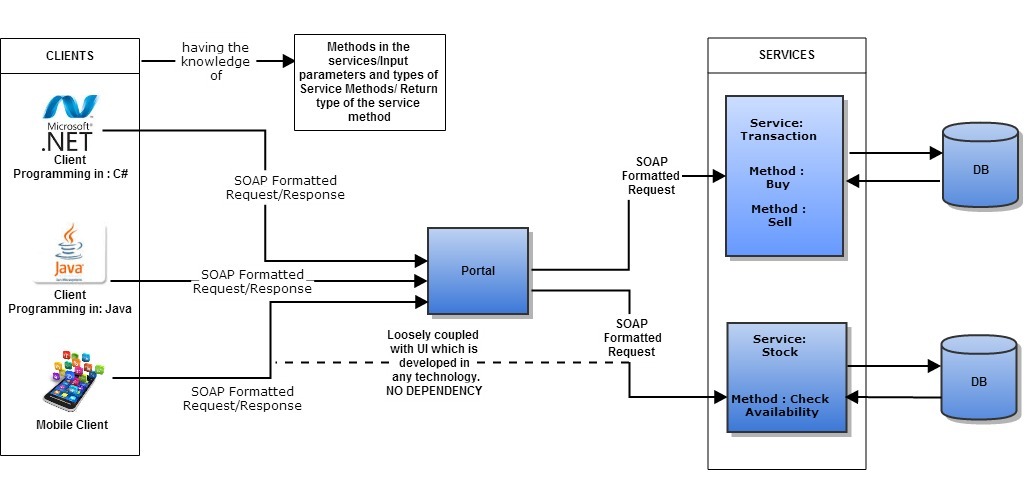
A Service - it has the methods with the defined contracts and the implementation of the business logic to connect DB or other service

*A Service Repository* -The service repository provides a home for the services, and the service bus provides an industry-standard mechanism for connecting to and interacting with the services



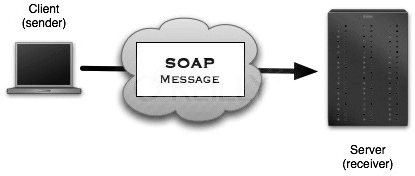
The above architecture, helping the businesses respond more quickly and cost effectively to the changing market conditions, Because all the services are decoupled from each another and from the application front end, SOA provides the desired level of interoperability in a nonproprietary open-systems environment.

endLet's talk about SOA in another detailed example

**SOA – Service Oriented architecture** 

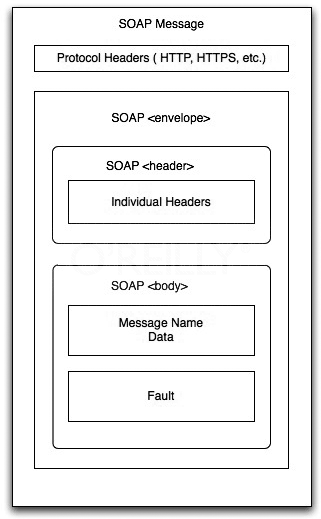
You could find two services named Transaction and Stock were hosted and it is consuming by the different clients whose UI been developed in different technologies like C#, Java, Android (Mobile App).. If the client has the knowledge of the input parameters, output parameters and return type of the method, across the platform the service will be consumed, This makes the system as a loosely coupled one, Infact the idea behind the SOA is to create the loosely coupled system. The clients will not be having a direct interaction with the server, instead the messages are formed out of the value of the input parameters, return parameters ,and the data are formatted by the SOAP. Every development platform has the SOAP stack, so working with service is supported in many environments. Implementing the application in this architecture will be the way to make the code and the functional behavior reusable in the future. Separation of concerns is an another advantage, When structuring the development team for a project, different sub teams can be assigned to the client and services, UI team can completely concentrate on user interaction stuffs without concern about the code dealing with business logic and the data access. Where as the services team can concentrate on forming the business logics in the service's methods without concern about the user interfaces. It clearly says that the developer is no longer responsible for the code end-to end,starting from the user interface ,business logic and the data access for the given requirement. So both of the teams can start their work at the same time. Another advantage here is UI related work can be outsourced and keeps the creation of business logic in house.

# SOAP Message



The above is an example how a client might format a soap message for requesting the stock information from the server

Structure of the message



Simple Object Access Protocol(SOAP) is a XML specification for exchanging data as structured information in messages across the platforms. SOAP standardizes how data are exchanged on the wire. SOAP envelope contains an optional header and a required body element. Header can contains information needed for the underlying technical infrastructure to support the communication not the business functionality. Whereas the body contains the functional data.

Each parameter for the service operation is present in the body in a serialized representation of data.A fault message will contains details about the exception (as appropriate) and one of the fault codes defined by the SOAP specification Client Request

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<getProductAvailablity xmlns="http://www.soaexample.org/ProductAvailable">

<product>Toy 0608</product>

</getProductAvailablity>

</soap:Body>

# </soap:Envelope>

Response from the server to the client

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<getProductAvailablityResponse xmlns="http://www.soaexample.org/ProductAvailable">

<getProductAvailablityResponseResult>

<productName>Toy 0608 </productName>

<productID>0608</productID>

<description>Barbie Doll</description>

<price currency="NIS">510.00</price>

<inStock>true</inStock>

<AvailablePieces>170</AvailablePieces>

</getProductAvailablityResponseResult>

</getProductAvailablityResponse>

</soap:Body>

</soap:Envelope>

# 

# Summary

SOA is an architectural approach to the distributed systems to integrate flexibly with interoperability, Newer technology systems can be easily adopted into it.