Service-Oriented Architecture

Sanjay Chaudhary
DA-IICT
References

1. Service-Oriented Architecture: Concepts, Technology, and Design” By Thomas Erl, Publisher: Prentice Hall PTR
3. Dr. Srinivas Padmanabhuni, SOA Centre of Excellence, SETLabs, Infosys Technologies Ltd, Bangalore
4. Telecom Research Innovation Centre (TRIC), IBM Research Lab, New Delhi
Introduction

• Service-oriented architecture (SOA) is an approach to distributed computing:
  – that is loosely coupled,
  – that is protocol independent,
  – that is standards-based,

• where software is accessed as a service, where software resources interact over a network according to a contract

• Contemporary SOA represents an open, extensible, federated, composable architecture that promotes service-orientation and is comprised of autonomous, QoS-capable, vendor diverse, interoperable, discoverable, and potentially reusable services, implemented as Web services.
Defining SOA

• Though accurate, this definition of contemporary SOA is quite detailed. For practical purposes, let's provide a supplementary definition that can be applied to both primitive and contemporary SOA.

• SOA is a form of technology architecture that adheres to the principles of service-orientation. When realized through the Web services technology platform, SOA establishes the potential to support and promote these principles throughout the business process and automation domains of an enterprise.
SOA Interaction Pattern
Definition of Web services

- World Wide Web consortium (W3C)
  “a software application identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artifacts. A web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols”
SOAP message formats

SOAP intermediaries
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1"
        xmlns:wsse="http://docs.oasis-open.org/oasis-wsswssecurity-secext-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>Karan</wsse:Username>
        <wsse:Password
          Type="http://docs.oasis-open.org/oasis-wssusername-token-profile-1.0#PasswordText">Karan</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>

  <soapenv:Body>
    <customerCheck xmlns="http://CustomerLibrary/CustomerInterface">
      <CustomerID xmlns=""">Karan</CustomerID>
      <Qty xmlns=""">200</Qty>
      <Crop xmlns=""">Rice</Crop>
    </customerCheck>
  </soapenv:Body>
</soapenv:Envelope>
Abstract Part

```xml
<wsdl:definitions targetNamespace="...">
  <wsdl:type>
    <schema elementFormDefault="qualified" targetNamespace="...">
      <element name="calculatePrice">
        <complexType>
          <sequence>
            <element name="order" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="calculatePriceResponse">
        <complexType>
          <sequence>
            <element name="calculatePriceReturn" type="xsd:float"/>
          </sequence>
        </complexType>
      </element>
    </schema>
  </wsdl:type>
  <wsdl:message name="calculatePriceResponse">
    <wsdl:part element="impl:calculatePriceResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:portType name="CalculatePrice">
    <wsdl:operation name="calculatePrice">
      <wsdl:input message="impl:calculatePriceRequest" name="calculatePriceRequest"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

Concrete Part

```xml
<wsdl:binding name="CalculatePriceSoapBinding" type="impl:CalculatePrice">
  <wsdl:operation name="calculatePrice">
    <wsdl:soap:operation soapAction=""/>
    <wsdl:input name="calculatePriceRequest"/>
    <wsdl:output name="calculatePriceResponse"/>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="CalculatePriceService">
  <wsdl:port binding="impl:CalculatePriceSoapBinding" name="CalculatePrice">
    <wsdl:soap:address location="http://localhost:8080/FirstWS/services/CalculatePrice"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```
<?xml version="1.0"?>
<definitions name="Procurement"
    targetNamespace="http://example.com/procurement/definitions"
    xmlns:tns="http://example.com/procurement/definitions"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/" >
    <message name="OrderMsg">
        <part name="productName" type="xsd:string"/>
        <part name="quantity" type="xsd:integer"/>
    </message>
    <portType name="procurementPortType">
        <operation name="orderGoods">
            <input message = "OrderMsg"/>
        </operation>
    </portType>
    <binding name="ProcurementSoapBinding" type="tns:procurementPortType">
        <soap:binding style='document'
            transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="orderGoods">
            <soap:operation soapAction="http://example.com/orderGoods"/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
    </binding>
    <service name="ProcurementService">
        <port name="ProcurementPort" binding="tns:ProcurementSoapBinding">
            <soap:address location="http://example.com/procurement"/>
        </port>
    </service>
</definitions>
Yellow, Green, and White Pages in UDDI
SOA – The next in architectural evolution

Coupling
- Tight
- Loose

Granularity
- Fine
- Coarse

Scope
- Program/Application
- Enterprise

Object Oriented
(for e.g. Java Objects)
- Objects implement fine grained functionality
- Objects are accessed in memory
- Base unit – Objects

Component based
(for e.g. EJB’s & .NET remoting)
- Defines well defined functionality from a set of objects
- Components are accessed via distributed middleware (RMI, .NET remoting)
- Base unit – Classes, jars, wars

Service Oriented
(for e.g. web services)
- Represents a higher-lever business function
- Predominantly asynchronous and stateless
- Services are accessed via the network (HTTP etc.)
- Base unit - schemas/interfaces
SOAP, WSDL and UDDI Interaction
## SOA Standards Stack

<table>
<thead>
<tr>
<th>Functional</th>
<th>Policy</th>
<th>Security</th>
<th>Transaction</th>
<th>Management</th>
<th>Reliable Messaging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPE4WS, BPML, XPDL, WSCI, BPSS, etc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSDL, IDL, XML, UDDI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service communication protocol</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOAP, Binary data, XML-RPC, XMLP, CORBA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP, HTTPS, JMS, IIOP, MQ, MS-MQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
WS-* standards and specifications stack
SOA with Web Services - standards stack

Management
- WS-Manageability

Presentation
- WSRP

Business Process Orchestration
- BPEL4WS, BPML

Security
- WS Security
  - XML Encryption
  - XML Signature
- XKMS, SAML, XACML, WS-Trust, WS-Federation

Reliable messaging
- WS-Reliability
  - WS-Reliable Messaging

Transaction
- WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity
  - WS-Context, WS-CoordinationFrmk, WS-TransactionMgmt

Provisioning
- SPML

Addressing
- WS-Addressing
- WS-MessageDelivery

Discovery and Registry
- UDDI, WS-Discovery

Description
- WS-Policy, WS-PolicyAttachment

Messaging
- WS-Policy, WS-PolicyAttachment

Transport
- SOAP, XML, XML Schema
  - HTTP, HTTP(s), JMS, SMTP...

Core standards
- MSFT/IBM
- OASIS
- Other
SOA: An EAI Perspective realized through ESB

Service Flow

Existing Applications

Data

New Service Logic

Enterprise Service Bus (ESB)

BPEL Workflow

Transformation Engine

XML Tools (Schema, Validators, XPath evaluators)

Routing engine

Messaging backbone

Portal

SOAP Service Request (e.g. J2EE, .NET)

B2B Interactions
Myths about SOA

- SOA = Web Services
- Wrapping Services over Applications yields services
- SOA is a technology issue
- Shared Semantics is trivial
- SOA needs fully standardized mechanisms
- SOA can be done at an application level
- SOA can be implemented just by bringing an ESB
How services encapsulate logic
How services relate
How services communicate
WSDL definitions enable loose coupling between services.
How services are designed

- How should services be designed?
- How should the relationship between services be defined?
- How should service descriptions be designed?
- How should messages be designed?
Key Principles of SOA

- Loose coupling
- Service contract
- Autonomy
- Abstraction
- Reusability
- Composability
- Statelessness
- Discoverability
WS-* specifications
Orchestration of Services
WS-Security specification

- A list of security specifications that may be used as part of SOA.
- WS-Security
- WS-SecurityPolicy
- WS-Trust
- WS-SecureConversation
- WS-Federation
- Extensible Access Control Markup Language (XACML)
- Extensible Rights Markup Language (XrML)
- XML Key Management (XKMS)
- XML-Signature
- XML-Encryption
- Security Assertion Markup Language (SAML)
- .NET Passport
- Secure Sockets Layer (SSL)
- WS-I Basic Security Profile
A digitally signed SOAP message containing encrypted data
Source: TRIC, IBMRL, India
### Best Practices for SOA

<table>
<thead>
<tr>
<th><strong>SOA requires</strong> <strong>strategic approach for maximum Reuse and flexibility</strong>, with models for sharing Costs, benefits across the organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority of implementations as we see today are <strong>stopping at Integration+</strong> owing to Governance issues</td>
</tr>
<tr>
<td><strong>SOA need not mean same thing</strong> for everyone. Depending on the application domain it can be for EAI, Or for B2B integration or for even Infrastructure Virtualization</td>
</tr>
<tr>
<td><strong>There is Strong need to develop both Processes and People dimension</strong> in addition to technology dimension</td>
</tr>
<tr>
<td>Implementing SOA is <strong>not just about implementing an ESB</strong> as product vendors may claim</td>
</tr>
<tr>
<td><strong>Architects and Developers need to retrained</strong> to follow contract first development approaches</td>
</tr>
<tr>
<td>Initial investment is high, with payoff in the longer run, hence a <strong>long term view is important</strong></td>
</tr>
</tbody>
</table>
Conclusions

• SOA enables a loosely coupled model of distributed computing
• SOA is best applied with strategic intent on the scale of an enterprise
• A structured approach to SOA planning is vital
• SOA is not the same as web services though web services are the most popular means of achieving SOA
• SOA needs to looked at from the domain of its application
• Best practices right through life cycle of SOA planning will yield optimal results
Thank You!

Answers?