

## Testing Web Services based SOA

---

This Paper identifies the challenges in testing Service Oriented Architectures and defines the Test Approach and some of the Test Strategies that can be adopted in SOA environments built on Web Services.

# WHITE PAPERS

## Table of Contents

<b>1</b>	<b>INTRODUCTION TO SOA .....</b>	<b>3</b>
1.1	INDUSTRY DEFINITIONS OF SOA.....	3
1.2	COMPONENTS OF SOA .....	4
<b>2</b>	<b>CHALLENGES IN TESTING SOA / WEB SERVICES.....</b>	<b>4</b>
<b>3</b>	<b>SOA TEST APPROACH.....</b>	<b>5</b>
3.1	SERVICE/COMPONENT TEST PHASE .....	6
3.2	INTEGRATION TEST PHASE.....	7
3.3	SYSTEM TEST PHASE.....	7
3.4	SERVICE MAINTENANCE TEST PHASE .....	7
<b>4</b>	<b>SOA TEST TYPES.....</b>	<b>8</b>
4.1	FUNCTIONAL TESTING .....	8
4.2	INTEROPERABILITY TESTING.....	8
4.3	SECURITY TESTING .....	9
<b>5</b>	<b>CONCLUSION.....</b>	<b>9</b>

## 1 Introduction to SOA

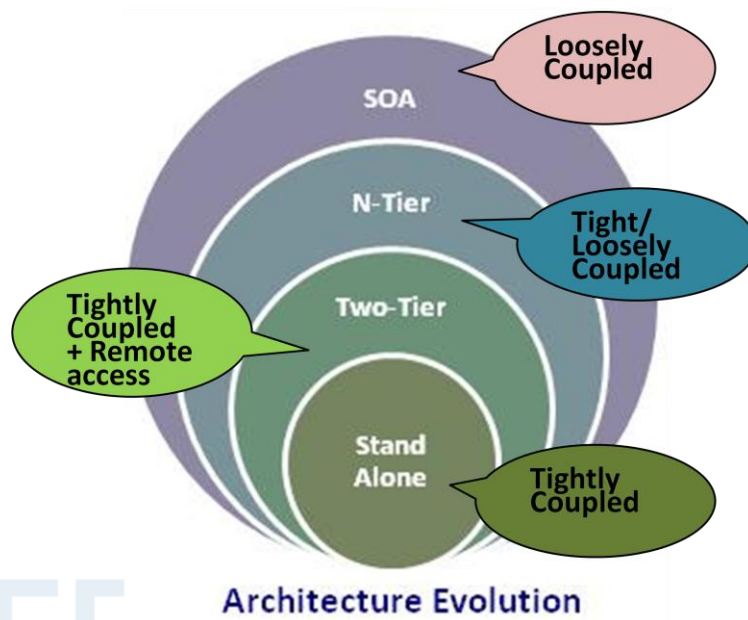
Most home grown and customized legacy systems which are the intrinsic part of many businesses are not flexible enough to quickly address changing market demands. By adopting service-oriented architecture, many organizations are successfully designing enterprise solutions in a more cost effective manner. Existing IT assets are being wrapped into services and reused across the enterprise.

### 1.1 Industry Definitions of SOA

SOA is an architectural style that guides all aspects of creating and using business processes, packaged as *services*, throughout their lifecycle, as well as defining and provisioning the IT infrastructure that allows different applications to exchange data and participate in business processes regardless of the operating systems or programming languages underlying those applications.

SOA refers to the design of applications via components (often referred to as "services") that expose interfaces that can be called by other client applications; multiple components will interact through request/reply mechanisms to create a larger "composite application".

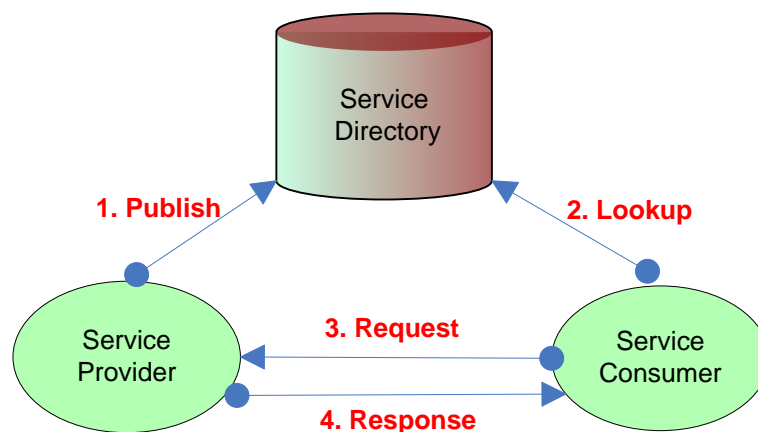
Though there are different ways to implement SOA, Web Services is emerging as a favorable alternative because of its seamless interoperable capability.



## 1.2 Components of SOA

Following are the components of a Service Oriented Architecture.

- ▶ Service – A component that is well-defined, self-contained, performs a specific function and does not depend on the context or state of other services.
- ▶ Service Provider – The provider of the service who publishes its service description in the service directory
- ▶ Service Consumer – The consumer of the service who accesses the service by looking up the directory
- ▶ Directory – The directory of services which contains the description of all the available Services



## 2 Challenges in Testing SOA / Web Services

### Tester's Skill set

Web Services testing requires an understanding of the architecture and the request/reply mechanism. The testing methodology is not a GUI based approach, but a request/reply mechanism based on the WSDL. It's mandatory for any tester to be able to understand the WSDL document. In addition to the above skills, the tester should possess a strong perception of the domain to design intelligent and effective test cases for the WSDL.

### Lack of User Interface

Ideally, Web Services do not expose any user interface. The tester/client to the service will be provided access to the WSDL for the service. For a traditional black box tester, this is a major shift in the Test Approach. Often, it may require development of customized client interfaces to test the functionality and inter-operability of the web services.

## Locating Errors

Traditional black box testing is focused on checking the functionality of the applications on the front end and verifying the data transfer across integrated systems. Whereas, testing in an SOA environment makes locating and analyzing errors an incomprehensible task for a typical tester. For example, a request for a particular transaction is initiated in system 'A', and the corresponding response has to be verified in system 'B'. But, system B does not exhibit the expected behavior.

The reason for failure could be

- The SOAP request message initiated through system 'A' might have an error.
- As a result the web service invoked might have delivered a bad response message
- Consequently, System 'B' which is integrated does not exhibit the expected positive behavior.

For a typical tester it is a big challenge to locate the actual error, because the error has occurred during the construction of the SOAP request message and it could be hidden in the hundreds of fields in the XML message.

With interlinked services, the situation is even worse. When a request is traversing through a bunch of chained services, being able to locate the error in a particular service imposes the biggest challenge.

## Diversified Development Locations

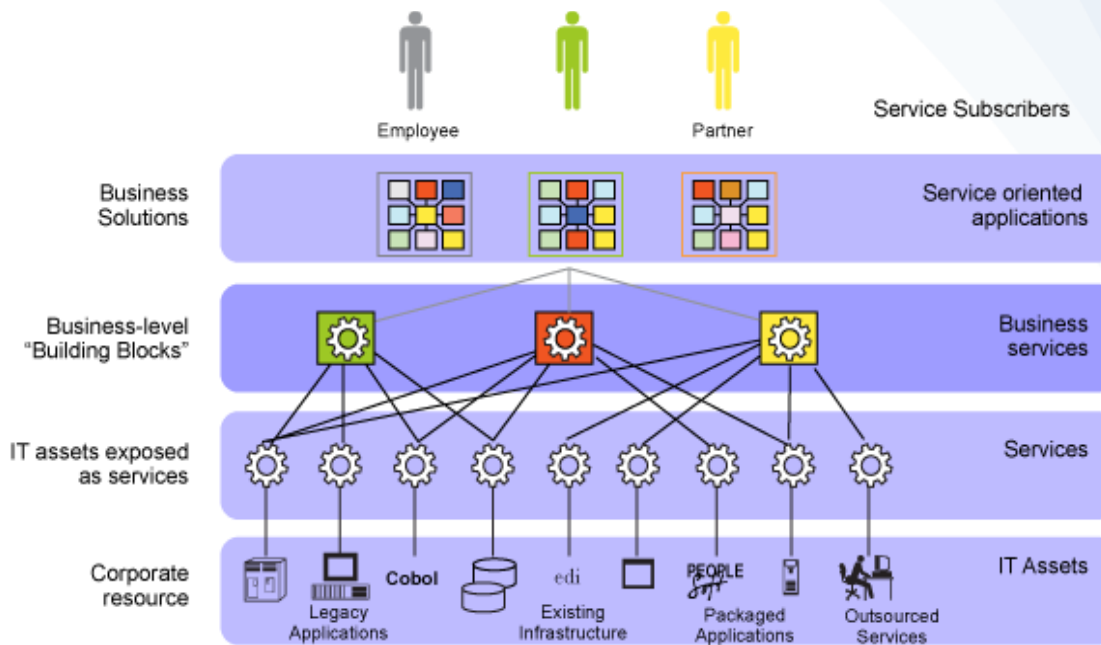
Typically, the deliverable unit of work in IT is an application. Instead of delivering conventional applications, SOA development teams deliver re-usable parts. A fundamental shift towards outsourcing and off-shoring complicates the testing process further. Limited accessibility to the components that are created by diverse SOA teams makes unit testing more complicated and unreliable.

## Defect Tracking and Regression Testing

In addition to the above challenges, tracking defects and maintaining test assets pose a major challenge in this area. If there is any modification or upgradation to a service, identifying the regression test suite and authoring new test cases for the added functionality leaves a knotty task on the tester. Being able to identify the regression tests and invoke the appropriate methods requires a fair understanding on the functionality of the methods of every service.

## 3 SOA Test Approach

The most effective way to test SOA environments is through a systematic end-to-end testing process that scrutinizes each layer of SOA as it's developed, run, and maintained. All the Services that are part of the architecture need to be tested independently and in integration with the other systems involved. With the advent of advanced Quality Assurance techniques like *Test Driven Development*, by laying the foundation for the preparation of tests from the start of the projects, the challenges in the SOA space can be understated.



## Phases of Service Integration and Infrastructure

The SOA Test Design should follow a Top down Approach and the Test Execution should follow a Bottom up Approach starting at the individual service level. Described below are the different phases in which SOA can be tested; at every phase of testing, all the different test types – Functional, Performance, Security, Interoperability, etc., should be conducted as applicable.

### 3.1 Service/Component Test Phase

This test phase ensures that each individual service in a whole universe of services deployed as a large package is functioning according to the standards and specifications.

WSDL, the Web Services Description Language is an XML document which contains the description of the service. This forms the basis for testing the service. Validation of WSDL can be considered as the first step in testing Web Services.

- ▶ Validation against the WSDL Schema
- ▶ Conformance check against WS-I's Basic Profile
- ▶ Regression testing of the WSDL

Unit, functional, performance, security testing should be performed on every service. The goal of this phase is to ensure that the service delivered from this phase is a completely tested independent component which does not induce problems into other components upon integration.

## 3.2 Integration Test Phase

The integration test phase focuses on service interfaces. This phase aims at determining if the functioning of the interfaces and the data that is shared between the services is according to the specifications. The tester has to ensure that all the services under this test phase comply with the specified interface definitions in terms of standards, format and data validation. Every time a SOAP request is initiated to a particular service, the behavior of the system or application in integration should also be tested.

Some of the challenges during integration testing would be to deal with missing components. There may be some third party web services which are not available during integration testing. So, the integration test strategy should involve simulation of those missing services.

## 3.3 System Test Phase

The system test phase in an SOA environment should verify whether the business requirements identified for the complete SOA solution have been properly met. The test team will need to identify certain key business scenarios for testing the entire system. This requires a very good understanding of the domain. Web Services orchestration testing plays a key role in this phase. Here the focus is to test for whether the data is being appropriately processed by different services to create a business process. In addition to this, all the non-functional requirements stated against the system like load, stress, etc, need to be validated. Also, SOA Governance strategies need to be clearly understood and appropriate verification check points have to be exercised in all phases to meet the governance criteria.

## 3.4 Service Maintenance Test Phase

Generally, organizations tend to upgrade services to grapple market needs, but would still need to maintain both the newer versioned service as well as the old one, because

- ▶ Not every client might be interested to migrate to new versions.
- ▶ Clients who are interested to migrate need to upgrade the existing client code and so demand some time.

As a result, service providers develop a version plan for maintaining the services. If the Web Services are maintained in parallel and multiple versions, then the test strategy should also involve testing each version of the service. If the versions are backward compatible, then the old client as well as the new client can access the services, but if there is a non backward compatible version, the old clients will fail to get the response from the server. Whenever changes are made to the services, unit and functional testing should be performed on the service. New and old versions should be tested in parallel for the timeframe that has been allocated in the versioning plan of the web services.

In addition to the above, testing the services for compliance to the Governance standards through out the entire life cycle should be taken care.

## 4 SOA Test Types

### 4.1 Functional Testing

The aim of Functional testing is to ensure that all the functional requirements stated against the system are met. Functional Testing of the services can be accomplished by identifying scenarios that represent any potential client behavior. The major challenge is to anticipate exactly what types of requests clients will send. Consumers of any service are definite to send different types of requests, negative inputs, and bad data to the Web service.

From a functionality standpoint, following are four main things to verify about the Web Services client (Service Requestor) and server (Service Provider).

- ▶ Whether the client is able to initiate a request to the service
- ▶ Whether the client exhibits the intended behavior on receiving the response
- ▶ Whether the server exhibits the intended behavior when it receives a valid request
- ▶ Whether the server generates an appropriate response for invalid requests

To achieve the goal of functional testing, the tester should be able to author intelligent, efficient and highly targeted test cases. The tester can use details from the WSDL such as the location of the Web Service, transport protocol, data types, etc., to identify a comprehensive list of tests. Following are some points to be considered while designing the functional test cases:

- ▶ Test cases should be designed based on the WSDL by analyzing all the data options available, including negative testing.
- ▶ Tests authored should cover all the methods available in the WSDL, including the various inputs with different data types and invalid inputs.
- ▶ Every valid SOAP request submitted to the server should result in a valid SOAP response.
- ▶ The SOAP response message should be validated for conformance to specifications.
- ▶ Testing of the transport medium like HTTP, JMS, SMTP, etc should be taken care.

There are many tools available in the market for automated testing of the web services and SOA. Custom Test harnesses can also be designed and built to automate the test process. Automated tools provide very high cost effectiveness for maintaining and executing regression test suites in SOA environments.

### 4.2 Interoperability Testing

The core technologies (SOAP, WSDL, and XML) using which Web services are designed are intended to ease interoperability and integration. Nevertheless, embracing these technologies does not guarantee seamless interoperability. Developers may unwittingly introduce platform-specific features during the course of development. For example, a developer may implement a Java web service without taking into account the effects on a .NET client.

There are two stages in which interoperability should be verified:

- ▶ Static design time verification against the WSI standards.
- ▶ Active run time validation against different client environments.

Statically, interoperability should be verified by ensuring that a service conforms to the universally implemented set of standards publicized by the Web Services Interoperability (WS-I) organization. Basic Profile 1.1 (BP 1.1) consists of implementation guidelines to develop interoperable Web services. Tests should be designed to ensure that the web services are being developed as per the WSI specifications. Code reviews play a vital part in verifying the conformance to specifications.

Dynamic testing of interoperability of Web Services requires creating specialized test beds for a web service. A test environment that can simulate various client environments like J2EE, .NET, AXIS, etc, should be built and tested against services. Different versions of a language framework like .NET WSE 2.0 and 3.0 should be considered in addition to considering various SOAP tool kits. The web services should exhibit the right behavior irrespective of the client that sends the request. Comprehensive design-time and run time Service interoperability testing ensures that the services can be integrated independent of platform, operating system, and programming language.

## 4.3 Security Testing

The key concept behind the invention of the Service Oriented Architecture is to arrive at modular, reusable, flexible services which can be distributed across the globe, thereby achieving high ROI. Anytime, one works to implement an Open and Standards based architecture, the target of attackers becomes increasingly high.

Security testing of Web Services should verify compliance with the WS-Security specification in addition to the following

- ▶ SOAP sender authentication
- ▶ SOAP receiver authentication
- ▶ Malware travel over the SOAP messages being exchanged
- ▶ Confidentiality and Integrity of Data wrapped inside SOAP messages
- ▶ Attackers conducting denial of service
- ▶ Replay of SOAP messages
- ▶ Transport Layer Confidentiality and Integrity

The SOAP responses returned by the services should be checked for any proprietary data leakage, governance violations or any other criteria that might cause a vulnerability risk.

## 5 Conclusion

Most organizations are still in the early stages of implementing SOA, and challenges are very likely to emerge. This paper attempts to highlight the challenges surrounding testing SOA packages and suggests the Test Approach and some of the Test Strategies that can be applied in such environments. Though this paper does not define the Performance Test Strategy of the Web Services, it throws some light on the functional, inter-operability and security testing of web services based SOA applications.