Service Oriented Architecture in the View of Web Services: A Review

Prof. P. A. Pawade[1]

[1]#Assistant Professor, CSE Dept., Jawaharlal Darda Institute of Engineering & Tech.,Yavatmal
payal_pawade@jdiet.ac.in

Corresponding Authors List # [1]Prof. P. A. Pawade

Email: payal_pawade@jdiet.ac.in;

ABSTRACT

Service-orientation describes an architecture that uses loosely coupled services to support the requirements of business processes and users. A traditional web service built on XML technology was developed; first to understand the technology behind web services and secondly to demonstrate the limitations of the original SOA framework. The main drivers for SOA adoption are that it links computational resources and promotes their reuse. The following guiding principles define the ground rules for development, maintenance, and usage of the SOA. Reuse, granularity, modularity, compos ability, componentization and interoperability compliance to standards (both common and industry-specific)Services identification and categorization, provisioning and delivery, and monitoring and tracking.

Keywords— SOA, Distributed Computing, and Web Services

1. INTRODUCTION

SOA operations rely on a mesh (Mesh consists of semi-permeable barrier made of connected strands of metal, fibre, Mesh is similar to web or net in that it has many attached or woven strands) of software services. Services comprise associated, loosely coupled units of functionality that have no calls to each other embedded in them.[1]

SOA developers associate individual SOA objects by using orchestration (Orchestration describes the automated arrangement, coordination, and authority of complex computer systems, middleware, and services.). In the process of orchestration the developer accomplice software functionality (the services) in a non-hierarchical arrangement using a software tool that contains a complete list of all available services, their component, and the means to build an application utilizing these sources. Analogously, the Web Services Description Language (WSDL) typically describes the services themselves, while the SOAP protocol describes the communications protocols. Whether these description languages are the best possible for the job, and whether they will become/remain the favorites in the future, remain open questions. [1]

SOA depends on data and services that are described by metadata that should meet the following two criteria:

1. The metadata should come in a form that software systems can use to configure dynamically by discovery to defined services, and also to maintain coherence and integrity.

2. The metadata should come in a form that system designers can understand and manage with a reasonable expenditure of cost and effort.

SOA aims to allow users to string together rather large chunks of functionality to form ad hoc applications that are built almost entirely from existing software services. Each interface brings with it some amount of processing overhead, so there is a performance consideration in choosing the granularity of services. [2]

2. WHAT IS SERVICE ORIENTED ARCHITECTURE

Service-Oriented Architecture (SOA) is applications built using

SOA style deliver functionality as services that can be reused when building applications or integrating within the trading partners. There are many definitions of SOA but none are universality accepted. What is central to all, however, is the notion of service.

It can Standardizes interplay of services become building blocks that form business flows services can be reused by other applications. Service users send requests to service providers.A service provider can also be a service user.A service user can dynamic discover service providers in a directory of services. An ESB can intermediate the interaction between service users and service providers.A service user does not need to have the service operation available at build time the service is located and bound at runtime. A service changes business data from one state to another. A service is the only way how data is accessed, if you can describe a component in WSDL, it is a service.A text of design, deployment, and management of both application and the software infrastructure where all software is standardized into business services that are network accessible and executable.Service interface rebased on public standards for interoperability. [1]

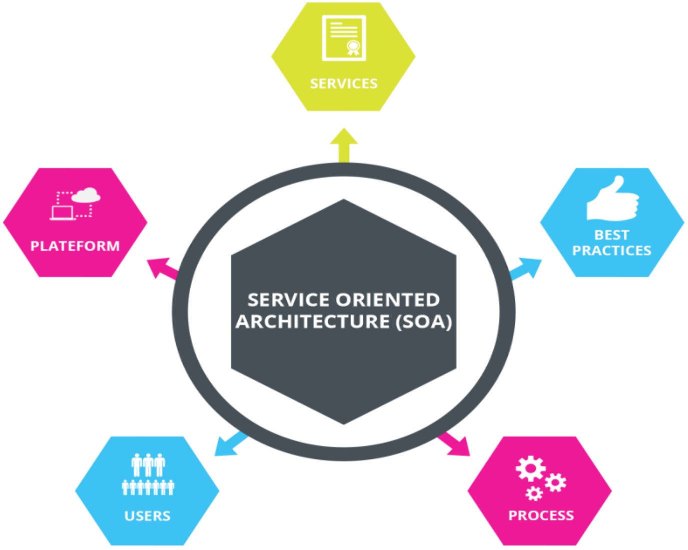

Fig.2.1 Service Oriented Architecture

3. WEB SERVICES

Web services can implement a service-oriented architecture. Web services make functional building-blocks accessible over standard Internet protocols independent of platforms and programming languages. These services can represent either new applications or just wrappers around existing legacy systems to make them network-enabled. [3]

Each SOA building block can play one or both of two roles:

3.1 Service Provider

The service provider creates a web service and possibly publishes its interface and access information to the service registry. Each provider must decide which services to expose, how to make trade-offs between security and easy availability, how to price the services, or (if no charges apply) how/whether to exploit them for other value. The provider also has to decide what category the service should be listed in for a given broker service and what sort of trading partner agreements are required to use the service. It registers what services and lists all the potential service recipients. The implementer of the broker then decides the scope of broker Public brokers are available through the Internet , while private brokers are only the accessible to a limited audience, for example, users of a company intranet. Some cover a broad landscape of services and others focus with in an industry. Some brokers catalog other brokers. Depending on the business model, brokers can attempt to the maximize look-up requests, number of listings or accuracy of the listings [3].

3.2 Service Consumer

The service consumer or web service client locates entries in the broker registry using various find operations and then binds to the service provider in order to invoke one of its web services. Whichever service the service-consumers need, they have to take it into the brokers, then bind it with respective service and then use it. They can access multiple services if the service provides multiple services. [3]

4. SOA AND WEB SERVICE PROTOCOL

Implementers commonly build SOAs using web services standards (for example, SOAP) that have gained broad industry acceptance. These standards (also referred to as web Service specifications) also provide greater interoperability and some protection from lock-in to proprietary vendor software. One can, however, implement SOA using any service-based technology, such as CORBA or REST. [2]

Architectures can operate independently of specific technologies. Designers can implement SOA using a wide range of technologies, including:

4.1 SOAP

4.2 REST

4.3 DCOM

4.4 CORBA

4.5 WCF

4.1 SOAP: (Simple Object Access Protocol)

SOAP is based on extensible markup language. It is an XML based message format that web service enabled the application use to communicate and interoperate with each other over the web. SOAP is a standard for encoding message in XML that invoke function in other application. [4]

4.2 REST: (Representational State Transfer)

REST is architecture style for building a scalable web services. It can approach to communication that is often used in the development of web service. The use of REST is often preferred over the more heavyweight SOAP (Simple Object Access Protocol) because REST does not leverage as much bandwidth, which makes it better fit for user over the internet. [3]

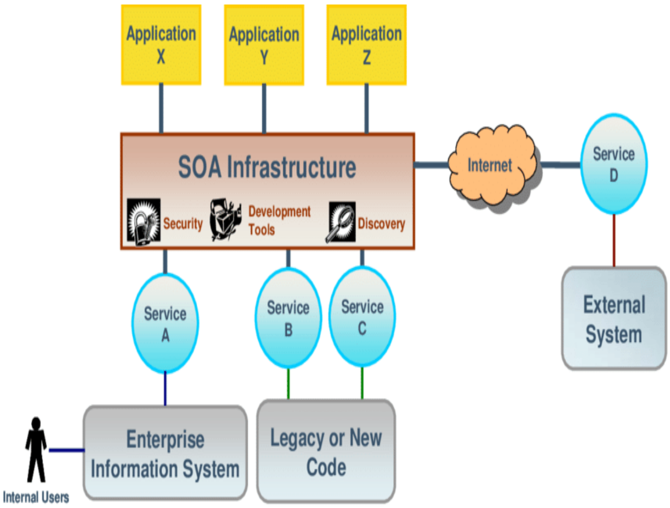

Fig 4.1 Components of SOA

4.3 DCOM (Distributed Component Object Model)

DCOM is an extension of component object model allows the component object model component to communicate across network boundaries. DECOM can also work on a network with in enterprise or on other network beside the public internet. It is generally equivalent to Common Object Request Broker Architecture (CORBA) in terms of providing a set of distributed service. [3]

4.4 CORBA (Common Object Request Broker Architecture)

CORBA is a standard development by the object management group to provide interoperability among distributed object. CORBA is a world's leading middleware solution enabling the exchange of information, independent of hardware platforms and programming language. [3]

4.5 WCF (Windows Communication Foundation)

WCF is a framework for building service oriented application. Using WCF, you can send data as asynchronous message from one service end point to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application.[1]

5. PRINCIPLE OF SOA

The following guiding principles define the ground rules for development, maintenance, and usage of the SOA:

- Reuse, granularity, modularity, composability, componentization and interoperability.
- Standards-compliance (both common and industry-specific).
- Services identification and categorization, provisioning and delivery, and monitoring and tracking.

The following specific architectural principles for design and service definition focus on specific themes that influence the intrinsic behaviour of a system and the style of its design:

- **Service encapsulation** – Many web services are consolidated for use under the SOA. Often such services were not planned to be under SOA.
- **Service loose coupling** – Services maintain a relationship that minimizes dependencies and only requires that they maintain an awareness of each other.
- **Service contract** – Services adhere to a communications agreement, as defined collectively by one or more service-description documents.

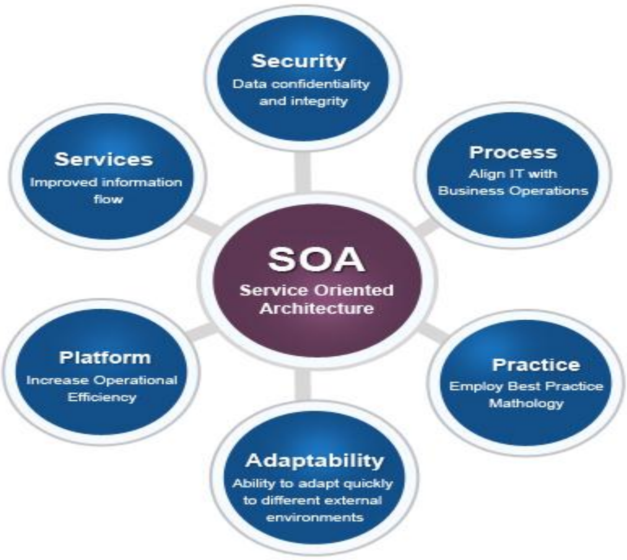
Security
Data confidentiality and integrity
Services
Improved information flow
Process
Align IT with Business Operations
SOA
Service Oriented Architecture
Platform
Increase Operational Efficiency
Practice
Employ Best Practice Mathology
Adaptability
Ability to adapt quickly to different external environments

Fig. 5.1 Applications of SOA

- **Service abstraction** – Beyond descriptions in the service contract, services hide logic from the outside world.
- **Service reusability** – Logic is divided into services with the intention of promoting reuse.
- **Service composability** – Collections of services can be coordinated and assembled to form composite services.
- **Service autonomy** – Services have control over the logic they encapsulate. [5]

6. THE ARCHITECTURE FRAMEWORK OF SOA

In real-time is achieved across these service models. In the SaaS service model, a specific methodology and tools have been developed, which allow application developers to engineer their application to deploy it within the SOI. The PaaS service model operates between applications and virtualized resources. As shown in the following figure, the core elements are Service Engineering and Service Management, which are described in more detailed in the subsequent sections. This layer aims to provide and manage the execution of real-time services inside the IaaS on request of the Application Layer, while conforming to the real-time constraints as determined in the Application- SLA. [4]

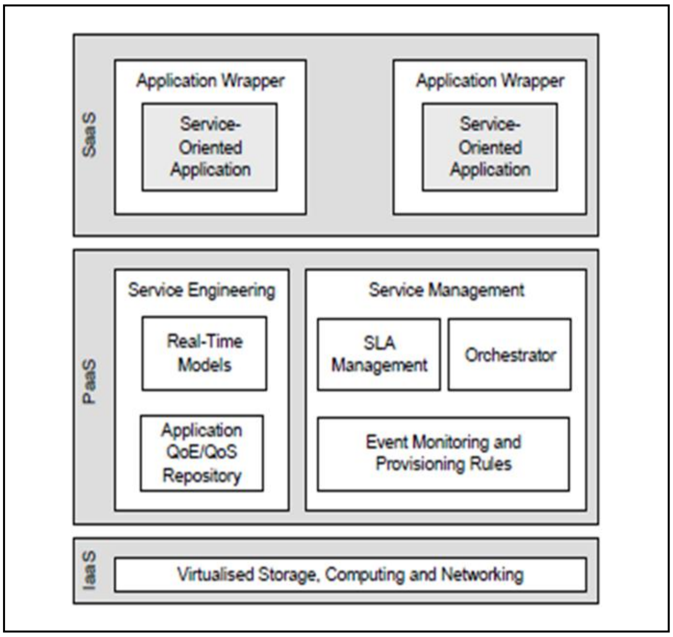
SaaS
Application Wrapper
Service-Oriented Application
Application Wrapper
Service-Oriented Application
PaaS
Service Engineering
Real-Time Models
Application QoE/QoS Repository
Service Management
SLA Management
Orchestrator
Event Monitoring and Provisioning Rules
IaaS
Virtualised Storage, Computing and Networking

Fig (6.1). Architecture of SOA

Apart from managing applications execution, the framework supports service engineering, fully automated SLA negotiation and re-negotiation, mapping high level performance parameters to low level resource parameters, discovery and reservation of the ISONI resources needed for the execution. [1]

7. REQUIREMENT OF SOA

In SOA interoperability between different systems and programming languages that provides the basis for integration between applications on different platforms through a communication protocol. One example of such communication depends on the concept of messages. Using messages across defined message channels decreases the complexity of the end application, thereby allowing the developer of the application to focus on true application functionality instead of the intricate needs of a communication protocol. [5]

Establish and maintain data flow to a Federated database system (A federated database system is a type of meta-database management system (DBMS) which transparently integrates multiple autonomous database systems into a single federated database). This allows new functionality developed to reference a common business format for each data element. Existing services can be used in different contexts. Reduced risk, cost and complexity for development – Clean architecture to reduced cost and risk – Increased developer productivity through reuse Service operations can be replaced as long as interfaces stay the same – Services can be relocated from one platform to another Service. [6]

8. CONCLUSION

Service Oriented Architecture is interaction occur with loosely coupled service that operate independently. Service oriented architecture is about creating assets that can be reused in many different contexts. The system designers must ensure the scalability of the functionality and regarding the technologies chosen. SOA Architecture can deliver functionality as a service that can be deliver functionality as service that can be used or reused. SOA uses open standard to integrate software assets as service.

REFERENCES

[1] Shikha Arya and Manisha Yadav (2014) “Servise Oriented Architecture: Concepts and Challenging Issues” ISSN:2277 128X K.

[2] Bell, Michael (2008). "Introduction to Service-Oriented Modeling". Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons. p. 3. ISBN 978-0-470-14111-3

[3] Bell_, Michael (2010). “SOA Modeling Patterns for Service-Oriented Discovery and Analysis”. Wiley & Sons.

P.390. ISBN 978-0-470-48197-4.

[4] Mike P. Papazoglou, Willem-Jan van den Heuvel "Service oriented architectures: approaches, technologies and research issues", Springer Verlag 2007

[5] Mike P. Papazoglou, Willem-Jan van den Heuvel "Service oriented architectures: approaches, technologies and research issues", Springer Verlag 2007

[6] Colan, M.: Service-Oriented Architecture expands the vision of Web services, Part 2. IBM Developer Works April 2004