

Comparison and Analysis of Service Oriented Architecture

*Thesis submitted in partial fulfilment of the requirements for the
award of degree of*

Master of Engineering

In

Computer Science and Engineering

Submitted By

Parneet Kaur

(801532040)

Under the supervision of:

Dr. Anil Kumar Verma

Associate Professor

Dr. Rajiv Kumar

Assistant Professor



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

JULY 2017

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Comparison and Analysis of Service Oriented Architecture*", in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Anil Kumar Verma, Dr. Rajiv Kumar and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Parneet Kaur
(Parneet Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Rajiv Kumar
(Dr. Rajiv Kumar)
Assistant Professor
Computer Science and Engineering
Department

Anil K. Verma
(Dr. Anil Kumar Verma)
Associate Professor
Computer Science and Engineering
Department

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds. With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide **Dr. Anil Kumar Verma**, Associate Professor, Computer Science and Engineering Department, Thapar University and coguide **Dr. Rajiv Kumar**, Assistant Professor, Computer Science and Engineering Department, Thapar University for their positive attitude, constant encouragement, keen interest, invaluable cooperation, generous attitude and above all their blessings. He has been a source of inspiration for me, I am grateful to **Dr. Maninder Singh**, Head of Department and **Dr. Ashutosh Mishra**, P.G. Coordinator, Computer Science and Engineering Department, Thapar University for the motivation and inspiration for the completion of this thesis. I will be failing in my duty if I do not express my gratitude to **Dr. S. S. Bhatia**, Senior Professor and Dean of Academics Affairs in the University for making provisions of infrastructure such as library facilities, computer labs equipped with internet facility, immensely useful for the learners to equip themselves with latest in the field.

Later but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted co-operation helped me in doing this thesis.

Parneet Kaur
(801532040)

ABSTRACT

In today's market, companies in the financial services have a common requirement to handle large amount of information with a high level of security and accuracy and provide on demand services. Innovative technology has been leveraged to endorse progressively complicated network of world-wide financial transactions which is managed by the financial services industry. Service Oriented Architecture (SOA) is becoming a popular access for integrating and designing enterprise applications. SOA enforce business services to deliver well defined services using a set of loosely couple services. Web services intended to be independent building blocks of Service Oriented Architecture. Web services are self contained. It provides a better alignment of IT with business vision, effective reuse of services, reduced cost of development and interoperability. SOA has its limitations and inherent issues. In this research, principles of SOA, benefits, limitations and challenges of SOA are discussed. We develop Currency exchange application based on Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). It is important to select appropriate interaction styles i.e., either SOAP or REST for building SOA services. Choosing service interaction style is an important architectural decision for designers and developers, as it influences the underlying requirements for implementing SOA service solutions. In this study, the performance of currency exchange application based on SOAP and REST is compared. Since SOA services operate over network response time are considered as a metrics parameter for evaluation.

Keywords: Service Oriented Architecture, Business Process Execution Language, Representational State Transfer, Simple Object Access Protocol, Web Service.

TABLE OF CONTENTS

CERTIFICATE.....	i
ACKNOWLEDGEMENT.....	ii
ABSTRACT.....	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES.....	vii
LIST OF TABLES.....	ix
LIST OF ABBREVIATIONS.....	x
1. INTRODUCTION	1
1.1 Service.....	2
1.2 Web Service.....	2
1.3 Enterprise Service Bus.....	3
1.4 Web Services Description Language.....	3
1.5 Simple Object Access Protocol.....	4
1.6 Representational State Transfer.....	4
1.7 Universal Description, Discovery and Integration.....	5
1.8 Business Process Execution Language.....	6
1.9 Client Server Architecture.....	6
1.10 Service Oriented Architecture.....	7
1.10.1 Principles of SOA.....	7
1.10.2 Benefits of SOA.....	10
1.10.3 SOA Limitations and Challenges.....	11
2. LITERATURE SURVEY	13
2.1 Service Oriented Architecture.....	13
2.2 SOA in Financial Sector.....	15
2.3 Response Time.....	21
3. RESEARCH PROBLEM	22
3.1 Problem Statement.....	22
3.2 Research Gaps.....	23
3.3 Research Objectives.....	23
3.4 Research Methodology.....	24

4. IMPLEMENTATION	25
4.1 Installation of Oracle SOA Suite 12c.....	25
4.2 Installation of Oracle XE 11g Database.....	26
4.3 Install Repository Creation Utility.....	27
4.4 WebLogic Domain Configuration.....	30
4.5 Starting the Servers.....	34
4.5.1 Starting the Node Manager.....	34
4.5.2 Starting the Administration WebLogic Server.....	35
4.5.3 Starting the Managed Server.....	36
4.6 Launching WebLogic Server in jdeveloper.....	38
4.7 Currency Exchange Database.....	39
4.8 Data Source Connection at Weblogic Server.....	40
4.9 Currency Exchange SOA Service Based on SOAP.....	42
4.9.1 XML Schema for SOAP based SOA Service.....	42
4.9.2 BPEL Process.....	43
4.9.3 Database Adapter.....	45
4.9.4 Integrate BPEL Process and Database Adapter.....	47
4.9.5 Deploy SOAP based SOA Service.....	49
4.10 Currency Exchange SOA Service Based on REST.....	50
4.10.1 XML Schema for REST based SOA Service.....	50
4.10.2 BPEL Process.....	50
4.10.3 Database Adapter.....	50
4.10.4 Create REST Binding.....	50
4.10.5 Adding a GET and POST Operation Binding.....	51
4.10.6 Adding a Mediator with Routing Rules.....	53
4.10.7 Deploy REST based SOA Service.....	57
5. RESULTS AND DISCUSSION	58
5.1 Currency Exchange SOA Service based on SOAP.....	58
5.2 Currency Exchange SOA Service based on REST.....	58
6. CONCLUSION AND FUTURE SCOPE	61
6.1 Conclusion.....	61
6.2 Future Scope.....	62
REFERENCES	63

LIST OF PUBLICATIONS

67

LINK TO VIDEO

68

LIST OF FIGURES

Figure No.	Description	Page No.
1.1	Service Oriented Bank Solution.....	1
1.2	Service Oriented Architecture.....	8
4.1	Oracle SOA Suite 12c Zipped File.....	25
4.2	Oracle XE 11g Database.....	26
4.3	Application Express Workspace.....	27
4.4	Database Connection Details.....	28
4.5	Select Components.....	29
4.6	Admin Server Configuration.....	31
4.7	Node Manager Configuration.....	32
4.8	Managed Servers Configurations.....	33
4.9	Configuration Success Screen.....	34
4.10	Node Manger Running Screen	35
4.11	Running WebLogic Server.....	36
4.12	Running Managed Server.....	37
4.13	Server states in WebLogic Administration Console.....	37
4.14	Configuration of WebLogic Server.....	38
4.15	Database of CurrencyDB1.....	39
4.16	Data Source Properties.....	40
4.17	Data Source List.....	41
4.18	Outbound Connection Pool.....	42
4.19	Design View of XML Schema.....	43
4.20	BPEL Process.....	44
4.21	Output Element of BPEL Process.....	45
4.22	Database Connection Details.....	46
4.23	Service Connection Detail.....	46

Figure No.	Description	Page No.
4.24	Routing Rules.....	47
4.25	Request Transformation Map.....	48
4.26	Reply Transformation Map.....	48
4.27	Deploy Currency Exchange SOA Service based on SOAP.....	49
4.28	REST Binding.....	51
4.29	Operation Binding for GET Method.....	52
4.30	Operation Binding for PUT Method.....	53
4.31	Target Services for Routing Rules.....	54
4.32	Mapper File for Request.....	54
4.33	Transformation Rules.....	55
4.34	Request Transformation Map.....	56
4.35	Design View of REST SOA Service.....	56
4.36	Deploy Currency Exchange SOA Service based on REST.....	57
5.1	Output of Currency Exchange SOA Service based on SOAP.....	58
5.2	Output of Currency Exchange SOA Service based on REST.....	59
5.3	Response Time for SOAP and REST.....	60

LIST OF TABLES

Table No.	Description	Page No.
1.1	HTTP Methods and CRUD Operations.....	5
5.1	Service Response Time for REST and SOAP Methods.....	59

LIST OF ABBREVIATIONS

Acronyms	Expansion
API	Application Programming Interfaces
AVD	Android Virtual Device
BMP	Bitmap Image
BPEL	Business Process Execution Language
CORBA	Common Object Request Broker Architecture
CRUD	Create Read Update Delete
DBMS	Database Management System
EAI	Enterprise Application Integration
ESB	Enterprise Service Bus
GUI	Graphical User Interface
IT	Information Technology
HTTP	Hyper Text Transfer Protocol
JDBC	Java Database Connectivity
JDK	Java Development Kit
JNDI	Java Naming and Directory Interface
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
OASIS	Organization for the Advancement of Structure Information Standards
PC	Personal Computer
PE	Processing Elements
RAM	Random Access Memory
RCU	Repository Creation Utility
REST	Representational State Transfer
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing

Acronyms	Expansion
STB	Service Table
SQL	Standard Query Language
UDDI	Universal Description, Discovery and Integration
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locators
WADL	Web Application Description Language
WSDL	Web Services Description Language
W3C	Worldwide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformations

In 1990s, banks started to think about business processes as items of software solutions and electronic information in the form of objects. To access user information, banks used telephone support system. To gather the useful data about the user, banks had to go through various steps. The terminals interact directly to the data storage computers and acquired a complicated set of commands for extracting the data, so this process was slow and prone to error. The banks created a common software interface for each individual account, using this new technology named as Common Object Request Broker Architecture (CORBA), to handle the issues. These CORBA objects related to distinguish user interfaces. All the information related to the customers of banks could be collected from a single screen. It provides a user interface separated from the actual data by providing a new abstraction layer. This was the commencement of three layer application architecture. This architecture has three independent layers named presentation logic layer, business logic layer and data information layer. At presentation layer, mobile banking services, internet banking, services related to customers are provided as shown in Fig. 1.1.

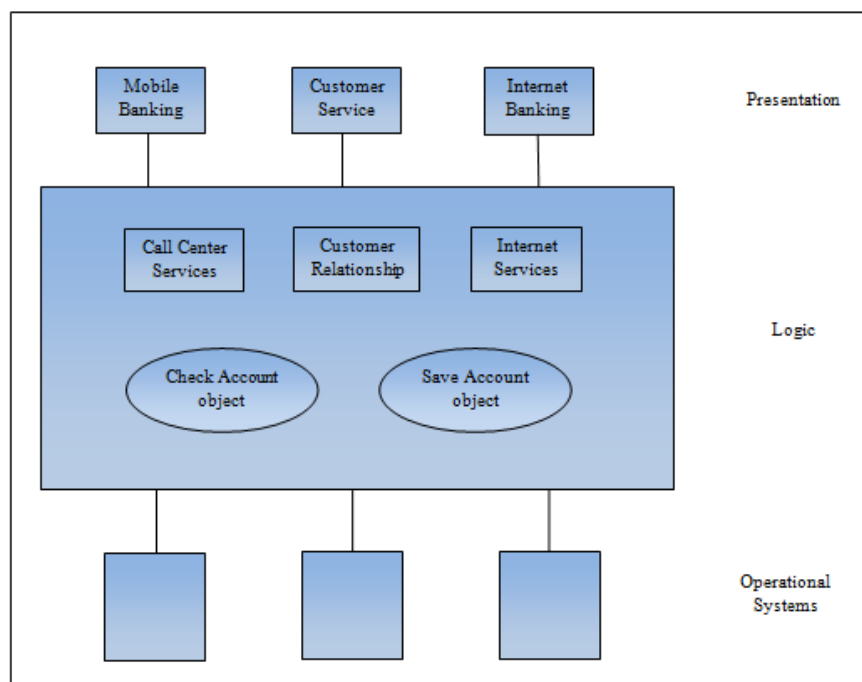


Fig. 1.1: Service Oriented Bank Solution

The example of the banks depicts the beginning of SOA. At service layer each service represents a unique business problem, like saving account savings or checking account savings. Without a standardized protocol, services had a problem to communicate with each other. With the increasing demand of these services, organizations realized the demand for a new construct to create, control and manage these services. Thus, Service Oriented Architecture came into existence because of its potential for solving these problems and specifies a design for e-service development [1].

1.1 Service

Service is the core key component of SOA foundation. Services are self-contained and standard part of business functionality with a specified outcome. A service should be an independent that doesn't depend on the state or context of other services. Service act as a black box for its consumers because only the purpose and outcomes of services are shown to its consumers and the implementation technique of the service is hidden from its consumers by encapsulating the functionality. Service interface is independent of the implementation. Services are reusable, having abstract behaviour, stateless and discoverable component. Service interacts with other services or an application in a loosely coupled style that can either be asynchronous or synchronous. Services are location transparent and support interoperability. The key benefit of developing services in SOA is that they give a usual medium of interaction through which services can co-operate with the diverse platform.

1.2 Web Service

Web Service is a distributed application component or a software component that can be platform independent and developed in any language and easily accessed over internet through Uniform Resource Locators (URL). Worldwide Web Consortium (W3C) designs the Web Services to enhance the communication between programs. The models for procedure calling protocol named as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) are already designed by W3C as official Web Service standards [2].

Web service is a prototype to provide services on web. Web services can communicate using open protocols. Various standards are there to support the functionalities of web services like SOAP, WSDL, UDDI and XML. Web service can work as an individual or combined with different services to form a group of services that deal with transaction that are complex in nature. Message exchange mechanism is specified by Simple object access protocol (SOAP) and it is based on XML language. It describes how we packed the data into a well defined manner and in structured form. Basically SOAP is a communication protocol that transmits the information over the web.

1.3 Enterprise Service Bus

Enterprise Service Bus (ESB) acts as a channel among the services that communicates with each other. To interact with the services, service needs the metadata of another service. ESB provides the metadata to the service [4]. ESB furnish an implementation layer. Services are exposed as a web service or as any another protocol and distributed over the network through an implementation layer. So that we can easily create and modify the existing services and new feature can easily added to the services. Managing services, logging and reliability are the basic functions that are provided by the ESB. ESB enables the services to be discoverable.

1.4 Web Services Description Language

Web Services Description Language (WSDL) provides mechanism to depict the service in terms of an interface. WSDL is based on XML standard. We required certain information related to the service like message passing transport protocol, where the services are located, what parameters are used to call the service, before calling the service. WSDL provides all these required information. The key benefit of WSDL is that it is supported by all the vendors. It gave the information that how we access the services over REST and other protocols and what services can do. Physical address and endpoint interface are described by the WSDL. Message format, network protocol and port type elements are described in the endpoint interface description.

1.5 Simple Object Access Protocol

Message exchange mechanism is specified by Simple object access protocol (SOAP) and it is based on XML language. It describes how we packed the data into a well defined manner and in a structured form. Basically SOAP is a communication protocol that transmits the information over the web. SOAP offers lightweight, simple mechanism to communicate information across various platforms and over various transport protocol. One way asynchronous messaging interaction patterns and two way synchronous messaging interaction pattern are provided by SOAP. SOAP document consist of interaction pattern i.e. document style or RPC style. Message format is described in SOA envelope. SOAP based web services have three entities: service registry, service requestor and service provider. XML and SOAP are used to carry operations among these three entities.

1.6 Representational State Transfer

Representational State Transfer (REST) is an architectural style that uses HTTP protocol for data communication. REST is an independent protocol but mostly it uses HTTP as an underlying protocol. Web Application Description Language (WADL) is utilized to depict REST resources. Service URI, data that service requires and serves, request that can be addressed to the service can be described in a WADL files. Statelessness, uniform interface and addressability are the main principles on which REST relies. As a resource, REST models the dataset for addressability and URI is used to indentify an each resource. Any kind of information like search result, any row of database and document that can be named and important to be referenced are categorised as a resource. Uniform interface and HTTP standards are used to access REST resources. POST, GET, PUT and DELETE methods are used to perform CRUD operations such as creating, reading, updating and removing the tasks as shown in Table 1.1. Regarding statelessness, information related to the previous request is not required by the server to answer the new request and when the server wants to perform the requested action then all the required information is provided by the REST request to the server.

Table 1.1: HTTP Methods and CRUD Operations

HTTP Method	CRUD Operation
POST	Create a resource
GET	Retrieve a resource
PUT	Modify a resource
DELETE	Delete a resource

1.7 Universal Description Discovery and Integration

Universal Description, Discovery and Integration (UDDI) define a framework for publishing and finding services. Services can be searched either programmatically or manually. UDDI renders general plan for business to build a global registry, register services and find the available services. It provides graphical user interface (GUI) to explore and query the registry. It provides the sequence of services that are already existed in the registry. To access a particular service, we need pointer and information related to the service. This information is provided by the UDDI. The key benefits of UDDI are composition of services, dynamic selection of services, runtime binding and reuse of services. UUDI registry can be public or private.

UDDI comprises four type of description in its architecture as following:

- 1. Business Entity:** It contains basic information about the service, information about the entity.
- 2. Business Services:** The business services offered by the business entity which has a unique identifier.
- 3. Binding Templates:** It provides binding information about the technical details for accessing the entry point, URL and service entry point for a client to invoke a particular service.
- 4. T Models:** The t Model (technical model) points to the external specification of the service.

UDDI comprises six sets of Application programming interfaces (API): Publisher API, Subscription API, Replication API, Inquiry API, Security API, Custody and Ownership API for users to register services, explore services and to exchange the information.

1.8 Business Process Execution Language

Business Process Execution Language (BPEL) is used for composition and execution of a business process. BPEL is based on XML language. IBM, Microsoft and BEA developed BPEL and for standardization it is submitted to OASIS (Organization for the Advancement of Structure Information Standards). BPEL offers a business process extension because it is constructed on an existing web services. BPEL comprises of two files, BPEL files and WSDL files. Process, input variables, output variables, fault, receive and reply, partner link, operation and events etc. tags are defined by BPEL. Composite web service represents as a single point interface by using BPEL. BPEL model actual behaviour of business process controls the flow between the processes, invocation of services and sequencing of process activities. WSDL files and BPEL files are deployed on BPEL execution engine. BPEL engine deals with the process execution and interaction between the services. Web services in service oriented architecture can interact and share data using BPEL [3].

1.9 Client Server Architecture

The Client Server Architecture differentiates the connecting networks and client and server. Client sends a request to the server, waiting for the reply from the server and process the reply. The server, at first authorized the client and then performs the process that is essential to generate the desired result. Server relies on various protocols and data formats to deliver the information to the client. Basically server represents the stored procedures as application logic of the database. In an architecture in which multiple clients can directly access the server is known as two tier architectural style. In client server architecture, a service is given by the server as a set of services to the client. The client can use these set of services. Client does not aware about the existence of the other clients. Clients are only aware about the availability of the servers. Client and server are two distinct processes that are distributed over the network. It is not necessary that there is a 1:1 mapping between the processor and processes because multiple server processes can run on a single server processor.

Following are the variations of client server architecture:

- 1. Application Servers:** In this architecture, client can access the services through special application client installed software and through a browser. Server can executes and host the services and applications.
- 2. Peer to Peer (P2P) Applications:** In order to distribute data and files across multiple clients, this architecture permits the server and client to interchange their roles.
- 3. Client Queue Client Systems:** In this architecture, the client can interact with other clients and exchange data with one another through a server based queue. In this server act as a data queue for the servers. This is known as passive queue architecture.

1.10 Service Oriented Architecture

Service Oriented Architecture (SOA) is an architectural model that depends on an open standard platform, an independent design and language independent technologies and focuses on service based solutions as a way to improve the efficiency and performance of an enterprise. SOA is a conceptual architecture design approach to develop, design, deploy and manage business information system. Services are the building blocks of SOA archetype. Services are autonomous, interoperable software components. Services in SOA share a communication model with well defined API [4].

Consumer can access SOA service to create new business process from any location by using any device, using any operating system in any language. SOA is essentially a pool of services that pass messages from one service to another service using service interfaces. SOA provides application functionality as a service to other applications. SOA functionality does not depending on any vendor, product and application.

1.10.1 Principles of Service Oriented Architecture

SOA is a special kind of software architectures which is designed to create a dynamically organized environment of services that are capable of being partitioned and inter-operable [5].

Principles of service oriented architecture are summarized below:

- 1. Standardized Service Contract:** Services follow service description documents within a given set of services. Standardization helps to maintain the consistency of service description. It is useful for the developers or users to interpret the capabilities of a given web service. It specifies the guidelines for service providers that how the services are provided. It also determines guidelines for service requestors that through what method services are invoked.

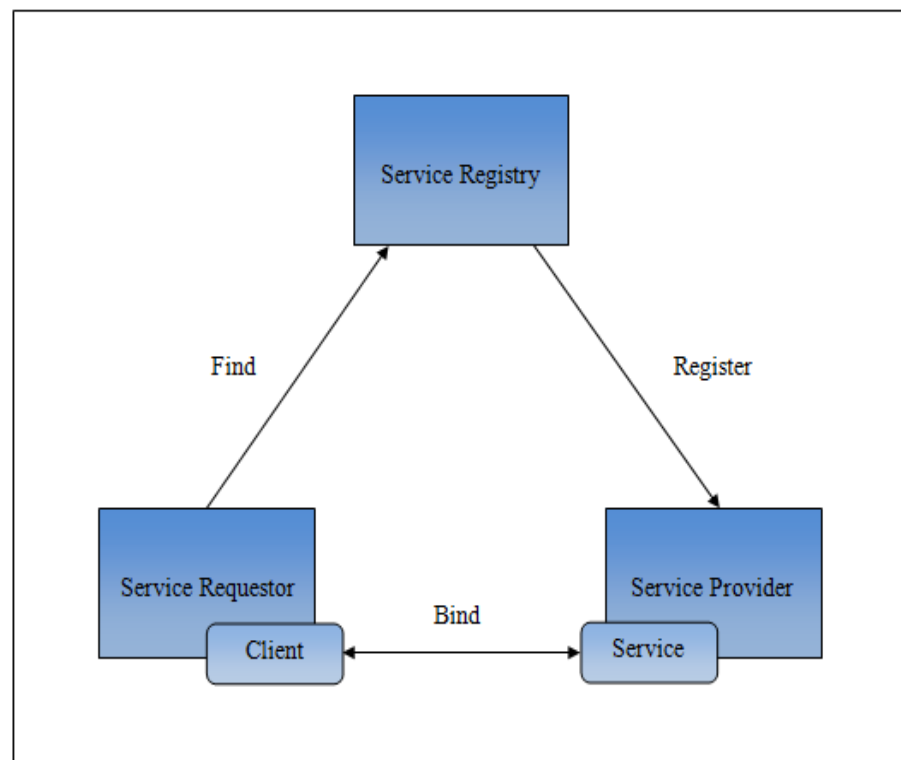


Fig. 1.2: Service Oriented Architecture

The service provider creates and publishes their types and description of service using Universal Discovery, Description and Integration (UDDI) directory as a common measure for services to be discoverable by its requestors over the network as shown in Fig. 1.2. The service registry deals with these received services. The service requester or service consumer invokes or bind the service using Simple Object Access Protocol (SOAP) messages. The service requester finds a service description stored in a UDDI by a service provider. When service requester or consumer invokes a service then the service consumer is bind to the service provider and selective data of the provider is shared with the service consumer. This selective data includes

the communication address and the basic information about service provider. The communication between the service registry and the service provider is the publication of service information. Similarly, the relation between the consumer and the registry is finding the service information. Contact between the service provider and the requester is basically a bidirectional interaction, indicating the request response between them.

2. **Loose Coupling:** Services are construct on the principle of loosely coupled means the relationship between the services should be minimized. Services are scattered over the internet, but services can be called from anywhere because services are independent of the location where it is present. Loosely coupled services can work independently and it has everything to provide functionality. Loosely coupled services create a flexible environment in which services are dependent on themselves or having a minimal dependency on other services. To achieve loose coupling, web service and its service requestor are independent of one another such that any requestor whatsoever can invoke the web service.
3. **Service Reusability:** Services are reusable. Services have their own resources (database), so services are independent from one another. Services should be coarse grained to reduce dependencies and to promote reuse. Services provide reusable business functionality via well defined interfaces [6]. Services must be designed in such a way that its internal logic supports reusability. For example: 1000 of debit cards are used at same time. So services can be used again and again instead of rewriting them.
4. **Service Discoverability:** Services can be discovered, accessed and reused, so services should be described by the service contract. Services should describe its purpose and its whole functionality. At design time and run time, services should be discovered, retrieved and interpreted. WSDL is used to describe the service to be as self discoverable as possible. It makes convenient for service requestor to locate a service and consume them.
5. **Service Abstraction:** Service provider can hide the implementation logic, design details and the technology used to build the service from the service consumer. Services act as a black box for its consumers. Except service contract, no other document should be available to the service consumer.

System is able to receive and respond SOAP messages that request for its functionality with the help of service layer without concerning the internal details and working of web services.

6. **Service Statelessness:** Service does not maintain any state information. Services do not retain any data. It takes data, performs actions on it and returns the requested value. It cannot provide sessions between service provider and consumer.
7. **Platform independence:** SOA services are independent in terms of hardware platform of the system, the programming languages and the operating system in which the services are implemented.

1.10.2 Benefits of Service Oriented Architecture

SOA offers numerous benefits as follow:

1. Due to loosely coupled nature of SOA, it allows software applications to be reusable. This saves time and allows for flexibility in applications development. It lowers the software development and management cost.
2. Services are loosely coupled, so it becomes easy to upgrade system products.
3. **Financial Benefits:** Due to quickly response to the services, revenue of business enhanced and there is drop off in the cost. Overall management expenses of services get saved. New products on the basis on new services are created and there is decrease in time to marketplace these products [7].
4. SOA has ability to create new process and develop new functions quickly from the already created services. Current business world required fast response to business services, so there is a requirement to make fresh products by using composite services. Services or composition of other services referred as Composite Service.
5. In today's business world, where agility and flexibility is required in businesses, with the help of SOA companies can respond quickly to customer needs. IT infrastructure can easily adjust to changing business requirements using SOA.
6. SOA services provide parallel and an independent development.

1.10.3 Service Oriented Architecture Limitations and Challenges

Limitations and challenges of SOA can be summarized as follows:

- 1. Security:** SOA is built using open standards. These standards do not have security of their own. This makes services and system vulnerable to attacks. Thus security becomes an issue. Lack of encryption and denial of service attack are the security flaws identified by SOA.
- 2. Lack of Logging Feature:** SOA does not have a logging feature. So it becomes difficult to track whoever has used the service and from where the request came in case there are breaches in security [8].
- 3. Web Service Standards:** Web services standards are open and amorphous. SOA is built using open standards SOAP, WSDL, UDDI and XML. High level services and security have not been standardized at all.
- 4. Governance:** SOA governance shows the establishment policies, controls and enforcement mechanism that contribute to the success of implementing SOA. The most common failure of SOA projects is due to the lack of working governance.
- 5. Interoperability:** SOA web services encapsulate XML data to exchange messages over the web. When we integrate these services in heterogeneous environments then a problem may be come across.
- 6. Application Ownership:** In SOA boundaries of application ownership is not clear. So it becomes an issue that who owns the application.
- 7. Quality Assurance:** Services are distributed over network and ownership is not clear, so SOA quality assurance becomes difficult [9].
- 8. Metadata Management Issues:** Services can interact with one another using message exchanging to perform tasks. Single application can exchange millions of message to communicate with other services to provide functionality. So managing services metadata becomes a big issue in SOA.
- 9. Organizational Issues:** Organizational issues include flexibility adoption especially when the technology is new and evolving. They include organizational structure and culture, enterprise connectivity, organizational standards maturity, goal settings, industry leadership, and governance. People

related issues are related to factors like expertise, skills, SOA teams, communication, stakeholder's participation, common language and vendor support [10]. It takes time to understand and use new technologies.

2.1 Service Oriented Architecture

The service oriented architecture is an application system that takes regular business applications and separates them into the individual business capacities and procedures, called services. A SOA allows you to create, deploy and integrate services. Services are independent of applications and the platforms on which services are run [11]. Service oriented architecture refers to a way of building reliable distributed systems that provide functionality as services, as well as extra emphasis on loose coupling between interacting services. Using Business Process Execution Language (BPEL), services are dynamically or statically consolidated. These services are from various sellers and publishers [12]. More than one service exists from various service providers to meet the requirements of enterprise customers or users to execute the similar task of reconfiguration. A service composer can choose a service that is in line with the requirement of competitive service providers [13]. As the rise of service oriented computing and service oriented architecture, using distributed object oriented development, and software development is changing. All major companies have been adopted and support SOA and SOC [14].

Different SOA frameworks have been created for web services, enterprise architecture [15], software engineering, operating systems, electronic productions [16], mobile computing system [17] and frameworks of the scientific workflow structure [18]. From these methodologies, we can outline the significant advantage of SOA so that various computing resources can be enclosed and packaged into an integrated service access interface. Services are characterized as various functionalities that are open to clients. All services are bundled in the function library and brought by standard function calls. Great adaptability and compatibility are provided by traditional SOA at lower cost by providing reusable modules. In addition, SOA can reduce the complexity of application development and integration to a great extent by providing well defined bundled interfaces. With all these advantages, the SOA idea has been generally applied in web services, software services and operating system designs. It is able to organize various processing elements (PEs) through well defined interfaces

and without concerning the software engineer with the usage of hardware platforms, working frameworks, programming languages and operating system. Therefore, the SOA-based configuration is an effective method for rapidly building prototyping frameworks [19].

Rajni and Bhuvaneshwari [20] proposed architecture for healthcare environment. This architecture enhances the authentication technique based on service oriented technology, to find any material from the web service using a general or knowledge based search. A new algorithm named as minute matching algorithm has been proposed using labelling technique for authentication technology. In order to assess the efficiency of the proposed algorithm, the researcher has used two parameters: false acceptance rate and false rejection rate. Both parameters and total execution time were contrasted with existing solutions. The new minutiae matching algorithm proposed here can be used to match the compressed image and it also supports different types of images such as JPEG, BMP, etc. Additionally the researcher has proposed the modified repetition elimination minhash clustering algorithm, which was used to minimize the search time of clustering the patient's data to the name of the disease. In order to minimize the search time, clustering algorithm was used. Two measurements were presented in order to assess the caching efficiency of the web service: provider file hit ratio and provider byte hit ratio for each page and throughput. The higher caching efficiency for the web service is shown by the provider byte ratio and lower value for provider file hit ratio.

Clement et al. [21] proposed a Service Oriented Reference Architecture for Smart Cities. In this architecture, each connected system in the city renders micro-services and services benefits in the SOA sense, can become progressively complex frameworks that use the existing infrastructure. As a result of the business model around the use of services provided by both private and public enterprise, the service economy can be considered as the foundation of advance city's development [22]. Many services can be used to work, possibly provided by various systems, which can be executed simultaneously in a process. As a result of these actions, complex behaviours may arise, such as reactions, to control or control the data, and enable the framework to grow as demand increases. One such case of a workflow may be that it

is a collection of simulations that model the conduct of some framework and give decision support or what-if analysis [23].

A process is categorized into two categories on the basis of services it provides- basic activities and structured activities as generalized in BPEL process. Basic activities consists of message receiving, provide a reply to message, wait, while, switching between messages and ordering that is the sequence of messages. Basic activities find out what happens in a process. The sequence arrangement of the procedures and structure of the process according to the needs are determined by the structured activities [24].

1.2 SOA in Financial Sector

With the excessive market competition, it becomes difficult to survive without taking an action towards improving the business services. Today's business world is uncertain and swiftly changes. It is difficult to estimate the future demands of a business. Innovative and fresh products can construct in business using SOA capability and it reacts promptly and suitably to the outside insistence [25].

Service-Oriented Architecture (SOA) allows organisations to run their IT business with their IT scheme, and makes an innovation system that is lined up with that of the business. Organizations demand to continually reassess the position of the organisation and rapidly adjust to modification in their business condition [26] [27] [28]. This is particularly valid for the financial industry. In this industry, organizations require to alter their exchanging procedures rapidly to take advantage of new business openings in a quickly changing market. This form of productivity is important for pricing and making new calculations for estimating and exchanging, and freeing up more capital via appropriate, convenient risk management. Apart from this, the request made by administrative as well as lawful consistence in the financial sector is also getting tough. That is why IT needs to create more consistent and more adaptable solutions, while the business value is also to be distributed. SOA is deployed by numerous financial foundations keeping in mind the end goal to make streamlined, easily managed and simplified infrastructures.

Markey and Clynch [29] consider two scenarios for the benchmarking session, which are done to enable a comparison to be drawn. First of all, service calls are utilized to request an object from each web service. An object test is the record of a country created from the database on the server. This situation is utilized to display the weight and request of each demand and response and it shows that the wrapping protocol may have an effect. The second situation involves two sections. The primary section considers the same format as the first scenario, each service call will request an object from its service and complete the service call using the database. The request for this scenario is for an accumulation of 249 country items. In this test, the call is repeated at a set time limit of one minute, each call is kept continuously. It will not only show the weight of the call and its corresponding wrapping protocol, however the amount of traffic that is repeated by each service can be made with the call. In the second part, only the use of GET headers in web API is appropriate to RESTful service calls. In this segment the test caching capabilities have been re-enabled. It will again demonstrate the weight of the traffic that is transmitted and it also demonstrates the number of calls which can be effectively set inside the given time limit. Also, considering the requests and responses of the services, it will indicate what influence enabling caching can have. In the primary test, a method was applied to both services, which communicated with the database through the unit framework and retrieved an item. This item was then bundled up and the customer was taken back to different formats. This item was then packaged up and transported back to the client in the various formats. This was used to demonstrate the differences in size of the sending and returning single payloads. It was used to show the difference in sending and returning the size of the single payload. Second test were included in two sections. The first type of service was used to call a method which would be a collection of items from the database (repeatedly 249 items per day). Again, each call will be kept continuously, followed by a second trial for the RESTful service, this time caching is enabled.

Johal and Singh [30] have provided a performance analysis of web services. Performance analysis is done for the two most significant methods that are utilized for the implementation and building web services for android-based mobile devices such that REST and SOAP based web services. Execute SOAP and REST web service and host them on Apache server. The server keeps running on windows 8 platform, with

IBM compatible PC 3.2 GHz processor and 2 GB RAM. An android application will be created and tested by an emulator without utilizing a physical device. An android emulator mirrors the majority of the equipment and software features of an average mobile phone, with the exception of that it can't hold real mobile calls. To evaluate an application effortlessly, the emulator uses android virtual device (AVD) designs. After an application runs on an emulator, it utilize android platform services to invite different applications, access the network, play sound and video, store and recover data, inform the client and present graphical transitions and themes. The experimental results demonstrate that REST web services beat SOAP web services.

Kumari and Rath [31] have provided the performance of the web services application for enterprise application integration (EAI) in view of SOAP and REST has been contrasted. Because web services work on network throughput and response time is considered to be a matrix parameter for assessment. They have considered a laonbroker system. In loanbroker system, customer requests best loan quotes between diverse banks. It clearly shows a complete ESB architecture. OpenESB tool is utilized for the construction and integration of loan broker system. Loanbroker application is the consequence of consolidation of various elementary services according to SOA standards. Utilizing BPEL engine of openESB tool, these services are integrated in SOAP interface. With the assistance of JSP, the production of a service is consolidated into each other as input in case of REST. In SOAP, BPEL module is accessible for direct support of service integration and it prepares services with the help of WSDL address. In REST, web application description language (WADL) is created. WADL is not supported by BPEL engine. BPEL supports WSDL address only.

Ruan et al. [32] proposed SOA architecture for the development of shopping sites. This architecture design can illuminate service consolidation and extension issues due to business transformation and advancement.

Gomathy and Rajalakshmi [33] proposed the SOA Reference Architecture. This architecture guarantees that that service oriented architecture has been united together in engagements in the best processes around the world and customers have been exposed to high performance with posts and more than ever before. It incorporates three sections: SOA platform, SOA delivery architecture and SOA patterns. These

provide extensive incentive by assisting associations to conquer the most difficult restriction in SOA development and deployment. SOA delivery architecture comprises definitions and descriptions that include technical architecture, application framework and services in all application styles involving SOA concepts.

Brandner et al. [34] utilized web services to break technical contrasts between existing customer segments, to give a brought together interface and coordinate inter-organizational banking system. The concepts of SOA deployment include all present ideas and thoughts, for example, use of an integrated point to guarantee consistent access to the framework, or technology neutral interface services to take into account the framework to coordinate with other hierarchical frameworks. These ideas and standards give a basis on which one can create. These solutions can be obtained through web services, how SOA can be accomplished. For many organisations, it is the easiest way to deal with loose coupled architecture. So web services are the most well-known usage. Web service does not consider as the only means to implement SOA [35]. Various issues have to be determined in the creation of high performance financial transaction systems utilizing the current technologies. Financial systems technologies are basically in view of formal protocols whereas web technologies, for example web services concentrate on rendering the interoperability utilizing web based correspondence protocols like SOAP. The issues related to this are inefficiency results from the interpretation of messages to SOAP and the other way around. The proposed SOA framework was intended for the working of financial business applications because of the raised issues.

From a business point of view, SOA will have to offer some benefits to the organisation outside just tackling with the issues of technical integration. IT efforts should be lined up with the business strategy to solve the issues of technical integration. Ross [36] points out that enterprise architecture is utilized to keep the technology standards with regards to the business needs. Jones [37] describes it as a way toward understanding the business, vendor strategies and to accomplish its business goals and objectives, organizations align these methodologies to characterizing the practices for the organization. The presented SOA framework is a theoretical framework. It is a basic framework that is abstract with the usage and execution of the design and gives a valuable beginning stage on which the more

elaborated framework work. In order to maximize both interoperability and efficiency, the researchers distinguish among two services: internal services and interface services. Business functions are performed by internal services. Internal services imply that these services only meet with interface services or other internal services. To maximize performance, internal services interact using the predefined original message format. Mask for internal services is furnished by the interface services. The interface services are wrapped up by various endpoints that handle the interoperability. External client applications can interact with each other using interface services. These services are responsible to make an interpretation of the messages into the original format utilized by the internal services. This research contributes to the target on the issues of technical implementation of the presented SOA architecture and in this manner it runs to adopt the technology approach [38]. By understanding the flow of information between business context and processes, diverse activities were utilized to distinguish the functional components and how they identify with each other, and not how they will be executed.

In order to acquire competitive advantage, the financial sector relies intensely on such measures. Currently, such works are done onsite in the distributed environment. SOA and grid demonstrate an option innovation to give common access to heterogeneous assets in a protected, dependable, versatile way. Financial applications are ideal for grid based execution. Numerous applications include process and data compact computation. High accessibility of assets is requested by finance applications. The unavailability of assets implies that there is an absence in the market's business, thus, can prompt missed opportunities. Security assurance is of central significance. Moreover, administrative issues are present in the Geographical Courts, compelling institutions to put confinements on the access to spatial data over their distributed enterprises. Businesses are progressively dependent on their IT infrastructure. An organization's IT framework frequently endorses both approaches to communicate with clients and support essential day to day business functions.

Donachy et al. [39] et al consider the use case ABS that requires regularly terminate a portfolio of foreign derivatives in line with administrative prerequisites along with interior market risk control. These derivatives are extremely complicated, which require a computational intensive numerical system for the cost. In the ABS, in-house

abilities are constrained and due to the periodic nature of the necessity, they are not ready to invest in the development of solutions of their own home. A regulatory condition is that this information cannot be transferred outside the UK or circulated. Various financial service providers built up an online portfolio revaluation service that gives the outcomes and ABS has contracted to give these estimating pricing features. In assessing the use case, little or no work has been done to determine the spatial prerequisites for such SOA based infrastructures inside a high assurance environment. SOA based Grid infrastructure has raised as a suitable business option to an expensive devoted high end IT frameworks, particularly inside the finance sector.

The idea of a SOA gives a structural style which is specifically aimed at facilitating business and the between operation of various components of that business. By organizing the capacity as significant, granular services rather than obscure, enterprises can rapidly recognize their practical capacities and abstain from copying comparable abilities in various areas of the association. Limits the effect of modification in standardization of services and interoperability of services and expects a series of possibilities of enterprises. An enterprise upgrades its involvement within its partners and its third party service providers by implementing a service orientation [40]. To furnish service oriented architecture user management, access management, access control, public key infrastructure, encryption, secure storage capabilities, role management, audit, secure communication capacities and audit depends on the built in platform. Like every other technique in the web stack, web services are tested by various security issues. The assaults introduced by this approach made it clear that the insufficient secure web service servers may be influenced by a single or few messages. Despite the fact that vulnerabilities are consequence of implementation issues, a broad majority leverages web services. To overcome these issues, SOA developers and architects should be behind these weaknesses and potential effects. The broad and dynamic hazard modelling approach can make us better for confront and reduce these assaults and arranged the service-oriented architecture as safe as would be. We recommend that implementation of services and service oriented designs outline the dangers and exploit the tools to enhance their understanding to the attacks and weaknesses there [41].

2.3 Response Time

Response Time is defined as the total amount of time it takes to serve a request for service. At the point when the customer application requests to serve with proper input, then server continues to deal with information and experiences a progression of services in light of the fact that the currency exchange application is consolidated utilizing each of these services. After processing on demands, the resulting output is sent to the customer as an answer. At any given moment the concurrent request is transfer to the server. For distinctive number of the service requests and for various files size, response time is calculated.

$$R_t = t_1 - t_2 \quad (1)$$

Where R_t = Response Time

t_1 = Time on which the request was sent

t_2 = Time on which the response is received

To compare, developed SOAP based REST based web services. The communication between the customer application and employee service was utilized as the reason for analyzing the performance of SOAP and REST.

3.1 Problem Statement

With the excessive market competition, it becomes difficult to survive without taking an action towards improving the business services. Today's business world is uncertain and swiftly changes. It is difficult to estimate the future demands of a business. Innovative and fresh products can construct in business using SOA capability and it reacts promptly and suitably to the outside insistence [42].

SOA offers an extended functional region together with the service oriented combination and process flow. It provides functionality to develop single application logic or advancement of a consolidated application logic which may incorporate different applications and services within the enterprise and expects to take care of a correct business issue area and serve generally engaged joined functionalities. Service based application advancement has more challenges and has particular prerequisites than the improvement of a traditional application.

SOA in an effective way is expected to depict what it is, examine which data must be incorporated into the structural model on service oriented architecture, and investigate possible and conceivable execution situations. Assortment of services like business process services, user interface (UI) services and technical services are included in service based enterprise application design. Without the necessity of the placement on a solitary PC, these services can be organized autonomously. Such application framework clearly contrasts from a traditional architecture. It requires special plan or design considerations which enable the application to utilize networked computational assets with offering a few key points of interest. Services are utilized to incorporate dissimilar applications and software systems inside various communicating partners and processing environment. Forthcoming potential applications that are not known at the development period of services are also supported by the design of services. Clarification of the problem domain and the meaning of the functional and non functional prerequisites essential for the entire framework are included in the service oriented analysis. When we developed services in the analysis and design phase of the application advancement, it is important to consider some methodologies and

configuration issues because service orientation requires particular decisions on the structure and modules of the application architecture. It is important to consider that how we can implement service oriented architecture in the most efficient way.

3.2 Research Gaps

- A lot of search is carried out in the field of service oriented architecture. But there is a lack of research in making the decision that how we implement service oriented architecture in the most efficient way.
- A very few research is done in the field of comparison of web services in service oriented architecture.
- There is a lack of research to analyze the performance comparative analysis by varying the number of clients and varying payload.
- In development of healthcare environment using SOA, a matching algorithm for authentication and identification of the user either by face, eye etc and two or more fields can be considered for clustering method.
- To distinguish the web service caching efficiency, a comparison with any existing one can be undertaken.

3.3 Research Objectives

In the light of above discussed research gaps following objectives have been formulated.

- A performance comparative analysis would be done among different SOA web services.
- To make a decision regarding how we can implement service oriented architecture in the most effective way.
- It is required to show the fundamental principles and concepts of service based application development.
- To get a handful experience on Oracle SOA suite 12c tool.

The core objective of the research work is to analyze the importance and usefulness of Service Oriented Architecture (SOA) in financial sector through system architecture design and implementation, which is developed in Oracle SOA Suite 12c using an external web services. Performance comparison of two SOA services will help us to make a decision regarding how we can implement Service Oriented Architecture in the most effective way. This analysis will also help us in deciding either to opt for REST based SOA services or SOAP based SOA services.

3.4 Research Methodology

In our research work we will use Oracle SOA Suite 12c and Oracle XE 11g database. Oracle SOA suite 12c is an integrated tool that is used for building and integrating SOA services. It is easily consumable by the modern web applications and it is service enabled. SOA Suite 12c provides an analytics and greater visibility controls to impose how processes and services are deployed, changed and reused. User installed the supported Java Development Kit (JDK) for the operating system and set the JAVA_HOME path.

The key objective of this thesis is to do performance comparison of SOA services using the two web service building technologies REST and SOAP. HTTP is used as a layer of transport by both the technologies. XML is used as medium of communication to transfer the messages from server to client for SOAP. REST depends upon four HTTP functions named as POST (to send data at the server side), GET (for retrieving data from the server), DELETE (to delete the data) and PUT (to update the data. SOAP also uses these four methods same as REST. Services may use combinations or variants of these methods. Each web service uses these four functions. We developed REST based and SOAP based SOA services that uses PUT, GET, POST and DELETE functionalities. We compared the performance of SOAP and REST SOA services.

4.1 Installation of Oracle SOA Suite 12c

Oracle SOA Suite 12c is an integrated tool for building and integrating SOA services. It is a standard platform that allows enterprise to easily and quickly accommodate the changes in the IT environments and in the business. It is easily consumable by the modern web applications and it is service enabled. During the entire lifecycle of services, Oracle SOA Suite 12c provides an analytics and greater visibility controls to impose how processes and services are deployed, changed and reused [43].

We installed the supported Java Development Kit (JDK) for the operating system and set the JAVA_HOME path under environment variables. Oracle SOA suite 12c includes JDeveloper, WebLogic Server, Oracle Service Bus, RCU, SOA BPEL and JDev SOA extension.

We can download the Oracle SOA Suite quick start installer zip file from <http://www.oracle.com/technetwork/middleware/soasuite/downloads/soa1213qsdownload-2769533.html>. The downloaded zip file has two jar files as shown in Fig. 4.1.

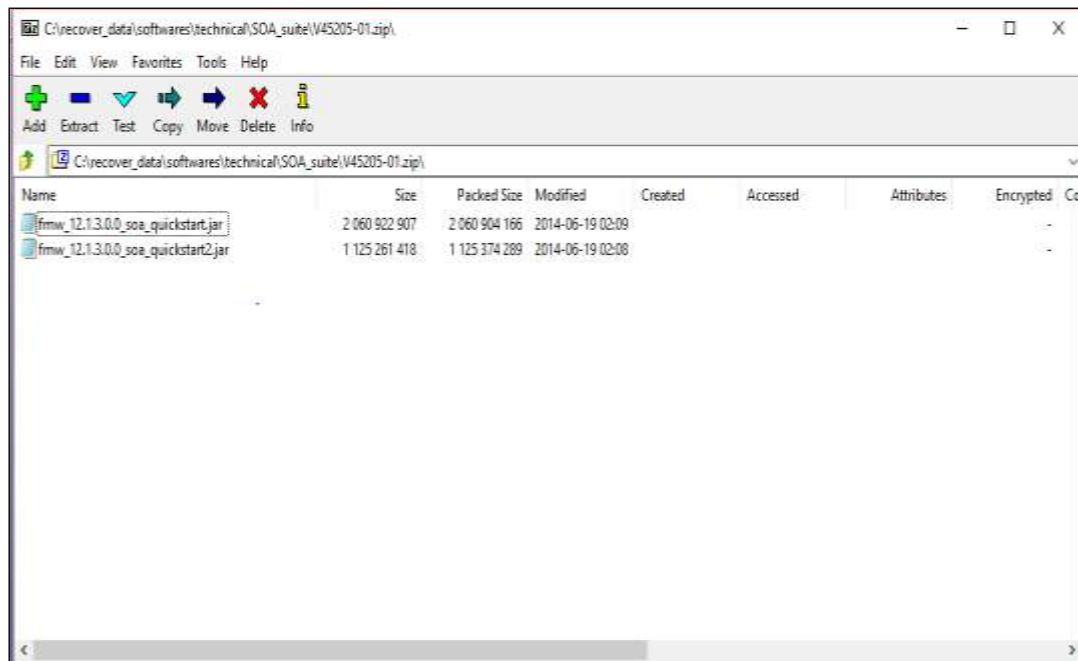


Fig. 4.1: Oracle SOA Suite 12c Zipped File

Choose an application express and provide oracle username and password to login to the oracle account. After the successful login, we have to create an application express workspace by providing database username, application express username and password as shown in Fig. 4.3.

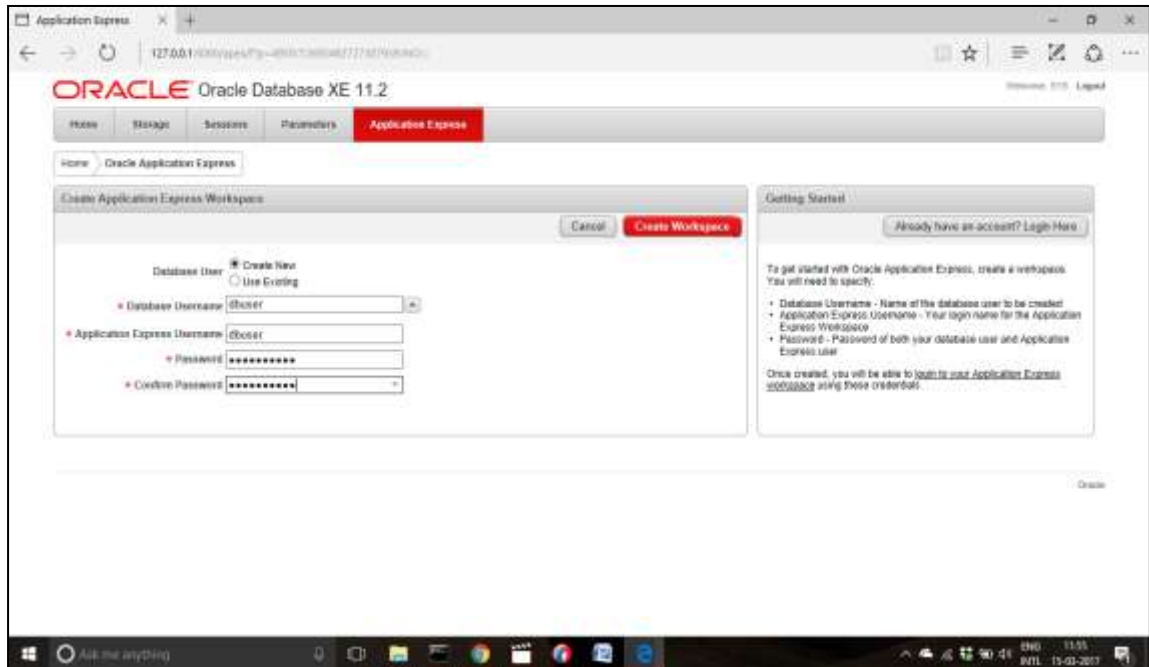


Fig 4.3: Application Express Workspace

4.3 Install Repository Creation Utility

Before the configuration of Oracle Fusion Middleware products like SOA suite 12c requires the existence of certain schemas in database. By using the Repository Creation Utility (RCU), we can create and load these database schemas for SOA in the database. Navigate to C:\Oracle\Middleware\Oracle_Home\oracle_common\bin folder and run rcu.bat file by clicking it or through command prompt. This will launch the RCU wizard installation with two options create schema and drop schema. We can drop the already created schema in the database by opting drop schema option. We have three options to create schemas:

- 1. System Load and Product Load:** If we have database privileges as SYS or SYSDBA credentials then we can select this option.

2. **Prepare Scripts for System Load:** Choose this option if we do not have full SYS or SYSDBA privileges on the database. SQL scripts will generated by this option which can be provided to our database administrator.

3. **Perform Product Load:** Use this option if we are a user with limited database privileges. On choosing this option, it can create and load tables, procedures, indexes, functions and other objects within schema and run any action.

After choosing the schema from above mentioned methods, database connection details appear to connect RCU with our database. In database type field, we should mention the type of the database we are using. It can be Oracle, IBM DB2 and SQL servers etc. Database is running on which host name or IP address is specified in host name field. Port specifies that at which port the database is listening the requests. Service name specifies the service name of the database. If we are using Oracle XE database then the service name is XE. In username field we can specify the username of the database. It can be SYS or SYSDBA. Password of the database is provided in the password field. We can provide a role as SYSDBA or normal in the role field. It is better to provide a SYSDBA role to the user as shown in Fig. 4.4.

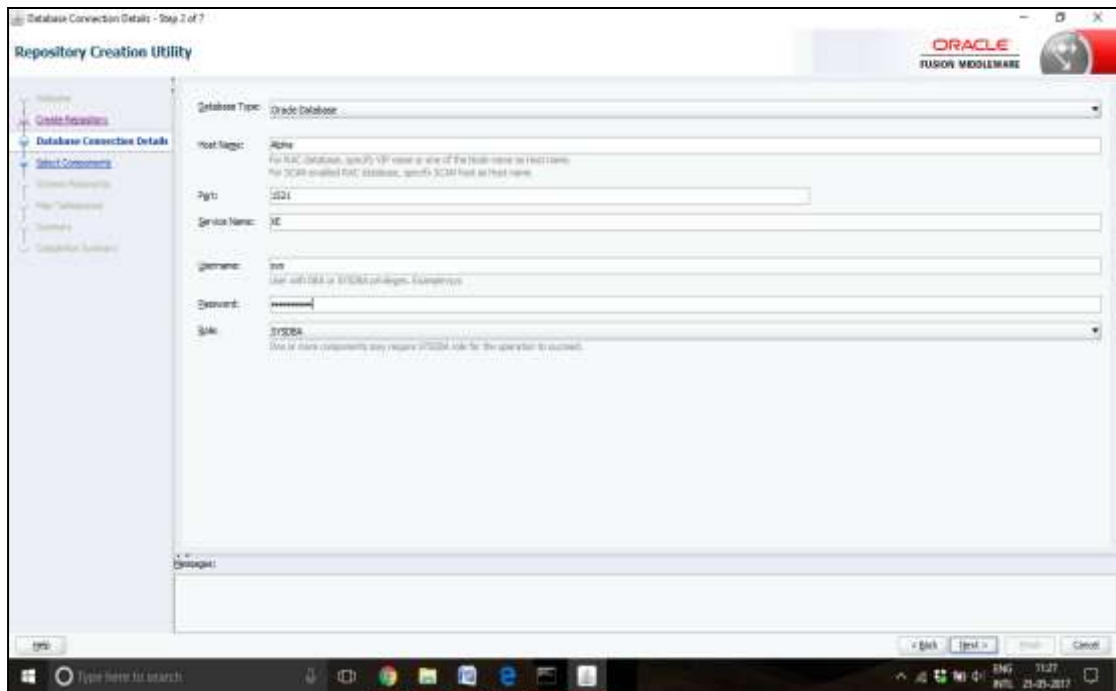


Fig. 4.4: Database Connection Details

After providing the entire database credentials details, a dialog box is appeared on the screen that checks the RCU connectivity and database prerequisites like initializing repository configuration metadata, obtain properties of the database etc. We can choose the components for which we want to create schemas and we can define a prefix to group these components together. We specify a prefix for the new schemas such as DEV, DEV1 etc. By defining a prefix for the schema we can easily access them while configuring SOA domain. In the component section we can select the SOA Suite components. By selecting SOA Suite it can select AS Common Schemas and all of their dependencies by default as shown in Fig. 4.5.

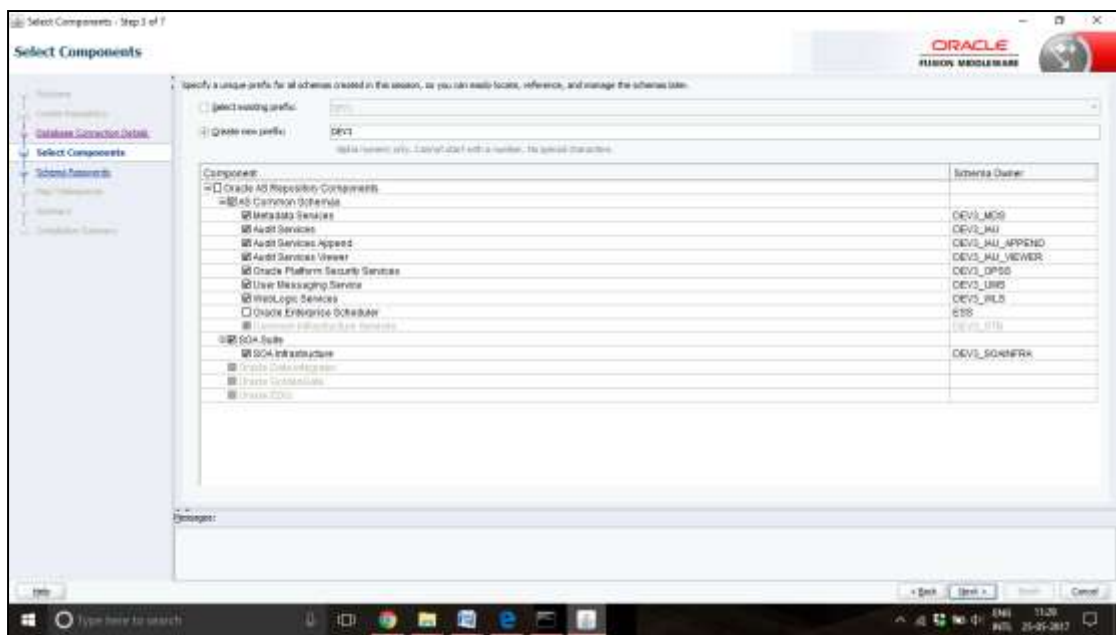


Fig. 4.5: Select Components

In the component selection screen if we select Oracle SOA Suite or Oracle Data Integrator component only then the Custom variable screen appears. If we choose Oracle SOA Suite, we will see the custom variables for Oracle SOA suite. If we select Oracle Data Integrator, custom variables for Oracle Data Integrator will appear on the screen. If we select the both components, then custom variables of both components appears on the screen. Map tablespaces screen review the desired tablespace mapping for the components. In summary screen, verify the information appeared on the screen and then click on create to begin the schema creation. Completion summary screen shows the information about the log files that are created from the RCU operation. We can view the contents of particular log file by clicking on the name of that file. If

RCU creation fails for any component or any error occurs during the create operation, then the cleanup for failed components checkbox appears on this screen.

4.4 WebLogic Domain Configurations

Oracle weblogic domain consists of particular Oracle weblogic server instance known as the administration server, which acts as a central point from which we can manage, create and configure the all resources of the domain. Weblogic server resources which are logically related forms a group that is known as Oracle weblogic server administration domain. While configuring the weblogic domain we can add additional Oracle weblogic server instances that are known as managed servers. For management and configuration of web services and servers we use administration server. We can deploy the web services, web applications, EJB's and other resources on managed servers.

To start the weblogic domain configuration, we can navigate to the C:\Oracle\Middleware\Oracle_Home\oracle_common\common\bin folder and run config.cmd or we can run the config.cmd through command prompt having administration privileges by navigating to the folder where config.cmd reside. We can create a new domain by selecting Oracle SOA Suite – 12.1.3.0, Oracle Enterprise Manager - 12.1.3.0, Oracle WSM Policy Manager - 12.1.3.0, Oracle JRF - 12.1.3.0 and WebLogic Coherence Cluster Extension – 12.1.3.0. We can provide username and password for authentication and domain mode can be production and Oracle hotspot JDK as JDK field. We can connect configuration wizard to the database by providing RCU data information. Schema information is automatically retrieved by Service Table (STB) schema that is required to configure the domain. While RCU data configuration provide the vendor of database, DBMS or service name, driver, host name, port number, schema owner and password of the schema for connecting to the database service table schema. Advance configuration consist managed servers, node manager server and administration server. While configuring the administration server, specify the administration server name, listen address and listen port number and specify the server group of the administration server as shown in Fig. 4.6.



Fig. 4.6: Admin Server Configuration

Administration server is required to configure the listen address. There must be one server instance in each weblogic server domain that acts as an administration server of that domain. Administration server maintains documents related to configuration of domain and communicate changes to the managed server if there is any change in the documents. We can use administration console to interact with the administration server.

We can start, shut down and restart the managed servers and administration server from a remote location with the help of node manager. We can manage the running state of different server instances that are distributed over the network from a centralized location that is managed server. Server instances are distributed over various domains and machines. We can monitor the server state and view log data of the servers with the help of node manager. A log file is created for a node manager process and log file of the server output is created for each server instance that is controlled by the node manager. We can configure the node manager by specifying the node manager type and credentials as shown in Fig. 4.7.

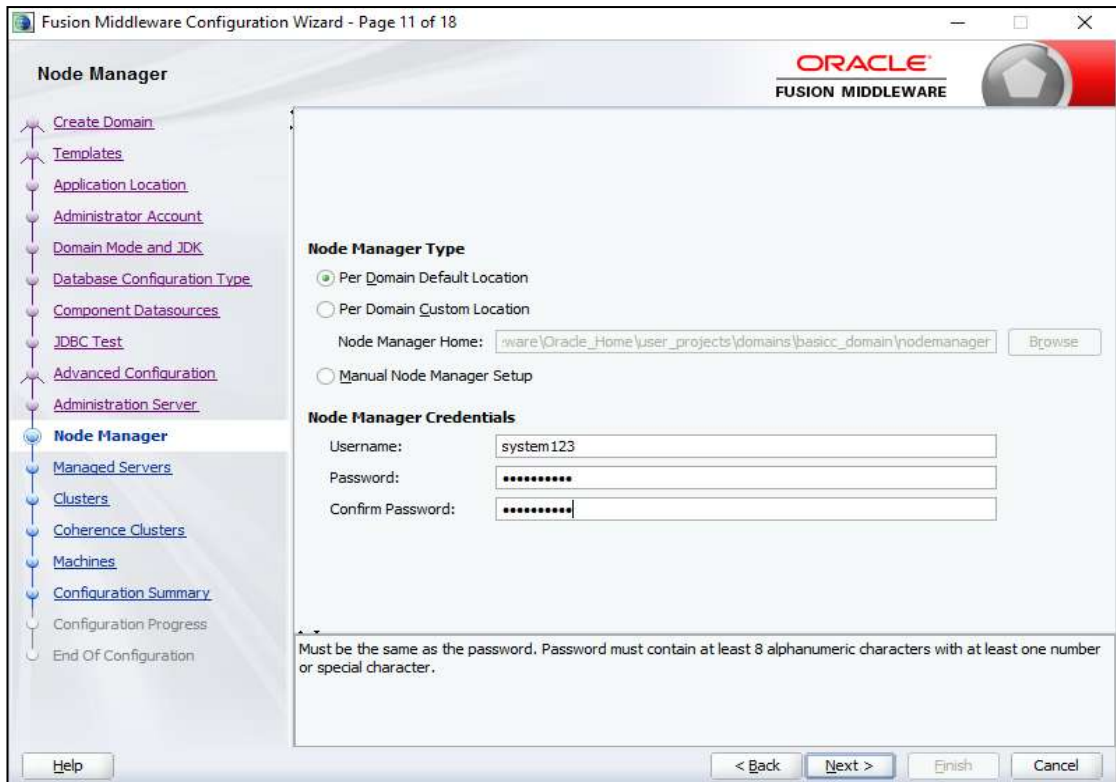


Fig. 4.7: Node Manager Configuration

Web services, business applications and application components are host by managed servers. Each manage server have a read only copy of the domain configuration document for better performance. When we start a managed server, it connects to the administration server to synchronize its configuration document with the administration server document. We configure managed server and server group is SOA-MGD-SVRS as shown in Fig. 4.8.

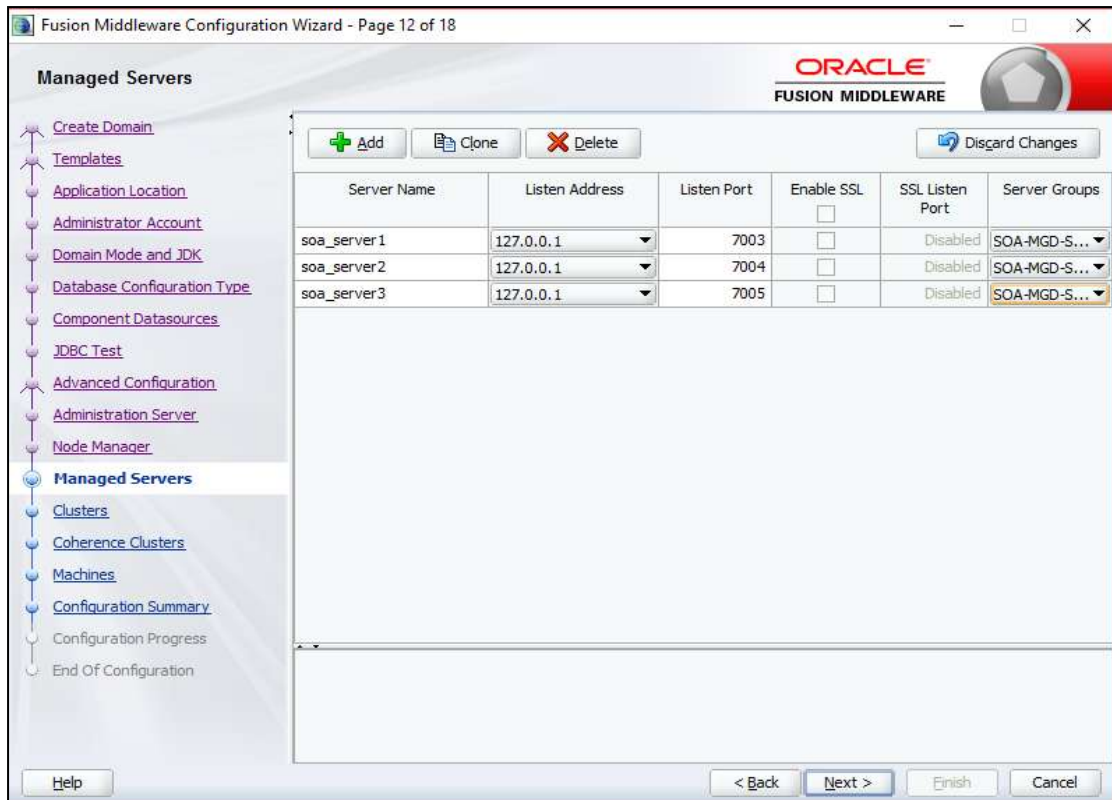


Fig. 4.8: Managed Servers Configurations

Two or more number of managed servers are configured as a cluster. Collection of multiple weblogic server instances is known as cluster. It will help us to provide high throughput, availability and high performance of applications through load balancing and handle failover. We can create cluster and assign managed servers to the cluster. These server instances can run simultaneously. By default coherence cluster is added to the weblogic domain. We can create machine in the domain in order to start and stop the server instances by node manager. Select the IP address of the machine on which we configure the node manager in the listen address field of node manager and specify the node manager listen port field. Assign the managed servers to the machines. In configuration success screen, it provides the URL to access the administration server and domain location to access the scripts to start and shut down the administration server, node manager and managed server as shown in Fig. 4.9.

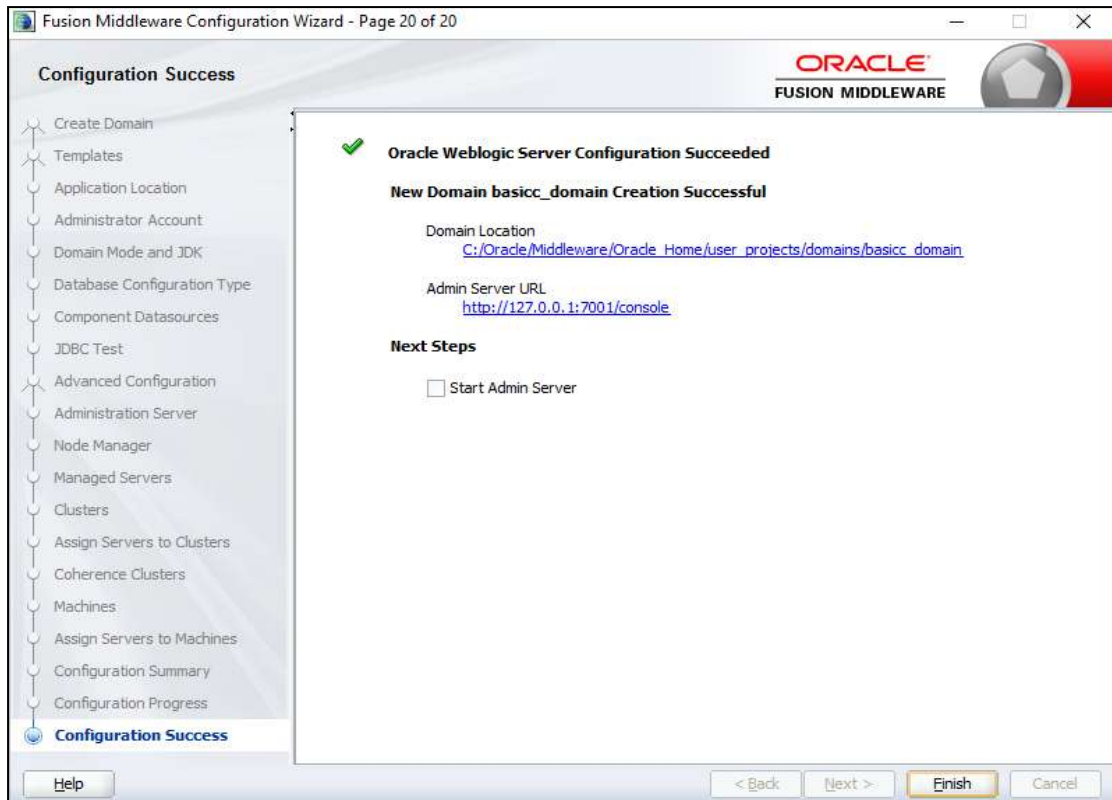


Fig. 4.9: Configuration Success Screen

4.5 Starting the Servers

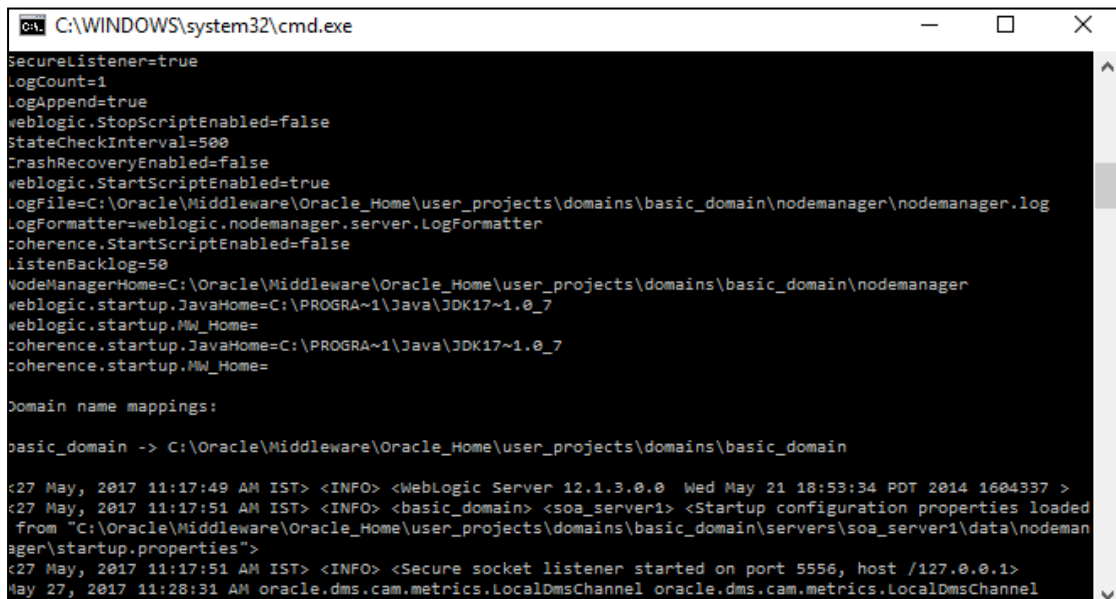
Once the domain configuration is completed, we can start the servers to access the tools of SOA Suite 12c.

4.5.1 Starting the Node Manager

In a weblogic server production environment, server instances are distributed over different machines, geographic areas and domains. We can start, restart and shut down the managed servers and administration servers with the help of node manager from remote locations. Node manager is connected with a machine. It is not connected with a particular weblogic domain. If server instances lie on the same machine on which node manager lies, then we can control the instances of server on any weblogic server domain with the same node manager.

To start a node manager we can navigate to the C:\Oracle\Middleware\Oracle_Home\user_projects\domains\basic_domain\bin folder and then click on the startNodeManager or we can run the startNodeManager.cmd on

command prompt. When the node manager successfully runs it shows the message in command prompt as shown in Fig. 4.10.



```
C:\WINDOWS\system32\cmd.exe
secureListener=true
logCount=1
logAppend=true
weblogic.StopScriptEnabled=false
stateCheckInterval=500
crashRecoveryEnabled=false
weblogic.StartScriptEnabled=true
LogFile=C:\Oracle\Middleware\Oracle_Home\user_projects\domains\basic_domain\nodemanager\nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
coherence.StartScriptEnabled=false
listenBacklog=50
NodeManagerHome=C:\Oracle\Middleware\Oracle_Home\user_projects\domains\basic_domain\nodemanager
weblogic.startup.JavaHome=C:\PROGRA~1\Java\JDK17~1.0_7
weblogic.startup.MW_Home=
coherence.startup.JavaHome=C:\PROGRA~1\Java\JDK17~1.0_7
coherence.startup.MW_Home=

Domain name mappings:
basic_domain -> C:\Oracle\Middleware\Oracle_Home\user_projects\domains\basic_domain

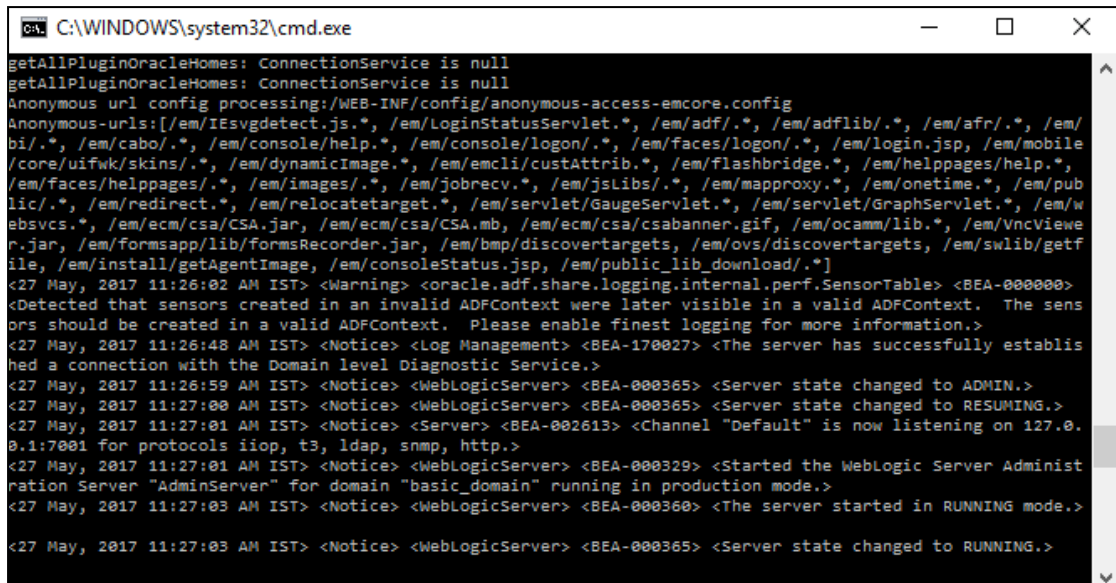
<27 May, 2017 11:17:49 AM IST> <INFO> <WebLogic Server 12.1.3.0.0 Wed May 21 18:53:34 PDT 2014 1604337 >
<27 May, 2017 11:17:51 AM IST> <INFO> <basic_domain> <soa_server1> <Startup configuration properties loaded
from "C:\Oracle\Middleware\Oracle_Home\user_projects\domains\basic_domain\servers\soa_server1\data\nodemanager\startup.properties">
<27 May, 2017 11:17:51 AM IST> <INFO> <Secure socket listener started on port 5556, host /127.0.0.1>
May 27, 2017 11:28:31 AM oracle.dms.cam.metrics.LocalDmsChannel oracle.dms.cam.metrics.LocalDmsChannel
```

Fig. 4.10: Node Manger Running Screen

4.5.2 Starting the Administration WebLogic Server

In each domain, one instance of weblogic server is organized as an administration server. All other instances of weblogic server are known as managed servers of weblogic domain. If there is a single weblogic server in the weblogic domain then that single server act as both administration server and managed server. The administration server acts as a central point that manages the weblogic server domain. It deals with the whole configuration of domain. The administration server hosts the administration console. Administration console is a web application based graphical user interface. Any web browser having network access to the administration server can access the administration console.

To start a weblogic server we can navigate to the C:\Oracle\Middleware\Oracle_Home\user_projects\domains\basic_domain\bin folder and then click on the startWebLogic or we can run the startWebLogic.cmd on command prompt. When the WebLogic successfully runs then its state changes from starting to running as shown in Fig. 4.11.



```
C:\WINDOWS\system32\cmd.exe
getAllPluginOracleHomes: ConnectionService is null
getAllPluginOracleHomes: ConnectionService is null
Anonymous url config processing:/WEB-INF/config/anonymous-access-emcore.config
Anonymous-urls:[/em/IEsvgdetect.js.*, /em/LoginStatusServlet.*, /em/adf/.*, /em/adflib/.*, /em/afr/.*, /em/
bi/.*, /em/cabo/.*, /em/console/help.*, /em/console/logon.*, /em/faces/logon/.*, /em/login.jsp, /em/mobile
/core/uifwk/skins/.*, /em/dynamicImage.*, /em/emcli/custAttrib.*, /em/flashbridge.*, /em/helppages/help.*,
/em/faces/helppages/.*, /em/images/.*, /em/jobrecv.*, /em/jsLibs/.*, /em/mapproxy.*, /em/onetime.*, /em/pub
lic/.*, /em/redirect.*, /em/relocatetarget.*, /em/servlet/GaugeServlet.*, /em/servlet/GraphServlet.*, /em/w
ebsvcs.*, /em/ecm/csa/CSA.jar, /em/ecm/csa/CSA.mb, /em/ecm/csa/csabanner.gif, /em/ocamm/lib.*, /em/VncViewe
r.jar, /em/formsapp/lib/formsRecorder.jar, /em/bmp/discovertargets, /em/ovs/discovertargets, /em/swlib/getf
ile, /em/install/getAgentImage, /em/consoleStatus.jsp, /em/public_lib_download/*]
<27 May, 2017 11:26:02 AM IST> <Warning> <oracle.adf.share.logging.internal.perf.SensorTable> <BEA-000000>
<Detected that sensors created in an invalid ADFContext were later visible in a valid ADFContext. The sens
ors should be created in a valid ADFContext. Please enable finest logging for more information.>
<27 May, 2017 11:26:48 AM IST> <Notice> <Log Management> <BEA-170027> <The server has successfully establis
hed a connection with the Domain level Diagnostic Service.>
<27 May, 2017 11:26:59 AM IST> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to ADMIN.>
<27 May, 2017 11:27:00 AM IST> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RESUMING.>
<27 May, 2017 11:27:01 AM IST> <Notice> <Server> <BEA-002613> <Channel "Default" is now listening on 127.0.
0.1:7001 for protocols iiop, t3, ldap, snmp, http.>
<27 May, 2017 11:27:01 AM IST> <Notice> <WebLogicServer> <BEA-000329> <Started the WebLogic Server Administ
ration Server "AdminServer" for domain "basic_domain" running in production mode.>
<27 May, 2017 11:27:03 AM IST> <Notice> <WebLogicServer> <BEA-000360> <The server started in RUNNING mode.>
<27 May, 2017 11:27:03 AM IST> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RUNNING.>
```

Fig. 4.11: Running WebLogic Server

4.5.3 Starting the Managed Server

There is exactly one administration server in the weblogic domain. All other servers in the weblogic domain are known as managed servers. Weblogic domain can have any number of managed servers. Managed servers are those servers on which your application can be executed. Applications are hosted on managed servers. Collection of logically related weblogic server resources is known as a weblogic server domain which manages as a single unit. There are one or more weblogic servers there in a domain and there can be one or more weblogic server clusters. Clusters comprise a group of weblogic server instances. These server instances work together in a group. Clusters provide high availability and scalability to the applications. Applications are managed and deployed as a component of weblogic domain.

To start a managed server we can navigate to the C:\Oracle\Middleware\Oracle_Home\user_projects\domains\basic_domain\bin folder and then launch the command prompt with administrator rights. In the command prompt we can write 'startManagedWebLogic.cmd managed_server_name http://administration_server_host:administration_port/console' to start the managed server. We can start the managed server from Oracle Enterprise Manager Fusion Middleware Control 12c. We can log in to Oracle Enterprise Manager Fusion Middleware Control 12c and then explore the weblogic domain under the target

navigation. Then select the name of managed server and select Start Up next to the weblogic server menu. When managed server starts successfully then its state changes from starting to running as shown in Fig. 4.12.

```
Administrator: Command Prompt - startManagedWebLogic.cmd soa_server1 http://127.0.0.1:7001/co...
ingframework.org/schema/tool/spring-tool-3.1.xsd'.>
#### On FabricKernelInitializerServlet, DiagnosticsFramework loaded from file:C:/Oracle/Middleware/Oracle_Home/oracle_
common/modules/oracle.dms_12.1.3/dms.jar
#### On MBeanManager, DiagnosticsFramework loaded from file:C:/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.dms_12.1.3/dms.jar
[EL Info]: 2017-05-27 11:39:25.235--ServerSession(1830936556)--EclipseLink, version: Eclipse Persistence Services -
2.5.2.v20140319-9ad6abd
[EL Info]: connection: 2017-05-27 11:39:25.298--ServerSession(1830936556)--file:C:/Oracle/Middleware/Oracle_Home/so
a/soa/modules/oracle.soa.fabric_11.1.1/tracking-fabric.jar_soa_purge_persistence_unit login successful
<27 May, 2017 11:40:59 AM IST> <Warning> <oracle.adf.share.logging.internal.perf.SensorTable> <BEA-000000> <Detected
that sensors created in an invalid ADPContext were later visible in a valid ADPContext. The sensors should be crea
ted in a valid ADPContext. Please enable finest logging for more information.>
<27 May, 2017 11:41:20 AM IST> <Notice> <Cluster> <BEA-000197> <Listening for announcements from cluster using unica
st cluster messaging>
<27 May, 2017 11:41:36 AM IST> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to ADMIN.>
<27 May, 2017 11:41:38 AM IST> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RESUMING.>
<27 May, 2017 11:41:41 AM IST> <Notice> <Server> <BEA-002613> <Channel "Default" is now listening on 127.0.0.1:7003
for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<27 May, 2017 11:41:41 AM IST> <Notice> <WebLogicServer> <BEA-000330> <Started the WebLogic Server Managed Server "s
oa_server1" for domain "basic_domain" running in production mode.>
<27 May, 2017 11:41:44 AM IST> <Notice> <Cluster> <BEA-000162> <Starting "async" replication service with remote clu
ster address "null">
<27 May, 2017 11:41:57 AM IST> <Notice> <WebLogicServer> <BEA-000360> <The server started in RUNNING mode.>
<27 May, 2017 11:41:57 AM IST> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RUNNING.>
<27 May, 2017 11:42:02 AM IST> <Notice> <Log Management> <BEA-170027> <The server has successfully established a con
nection with the Domain level Diagnostic Service.>
```

Fig. 4.12: Running Managed Server

In Oracle weblogic administration console, explore environment under the domain name and then select servers. If all the servers are running successfully, it will show the state of all servers running and health is OK as shown in Fig. 4.13.

Name	Type	Cluster	Machine	State	Health	Listen Port
AdminServer(admin)	Configured		machine1	RUNNING	OK	7001
soa_server1	Configured	cluster1	machine1	RUNNING	OK	7003
soa_server2	Configured	cluster1	machine1	SHUTDOWN	Not reachable	7004
soa_server3	Configured	cluster2		SHUTDOWN	Not reachable	7005

Fig. 4.13: Server states in WebLogic Administration Console

4.6 Launching WebLogic Server in jdeveloper

We can configure the weblogic server domain in jdeveloper with two options: first configure the integrated weblogic server and second is configure the standalone weblogic server. In integrated weblogic server, it creates the default domain in the jdeveloper. When we start a new jdeveloper session you need to relaunch the integrated weblogic server. When we close the jdeveloper it automatically shut down the integrated weblogic server. By using run or debug command applications can be automatically deployed to the servers. If we choose standalone domain for the weblogic server configuration in the jdeveloper, then server domain is independent of the jdeveloper. Jdeveloper doesn't start, stop and configure the standalone domain. We can manually deploy the application on the server. It doesn't allow deploying the application on the server by run or debugging command. Navigate to the C:\Oracle\Middleware\Oracle_Home\soa\common\bin folder using command prompt. Set the QS_template environment variables and then launch the standalone domain. Provide the username and password of the weblogic server for authentication. Specify the connection name and connection type. Give the configuration of the weblogic server as shown in Fig. 4.14.

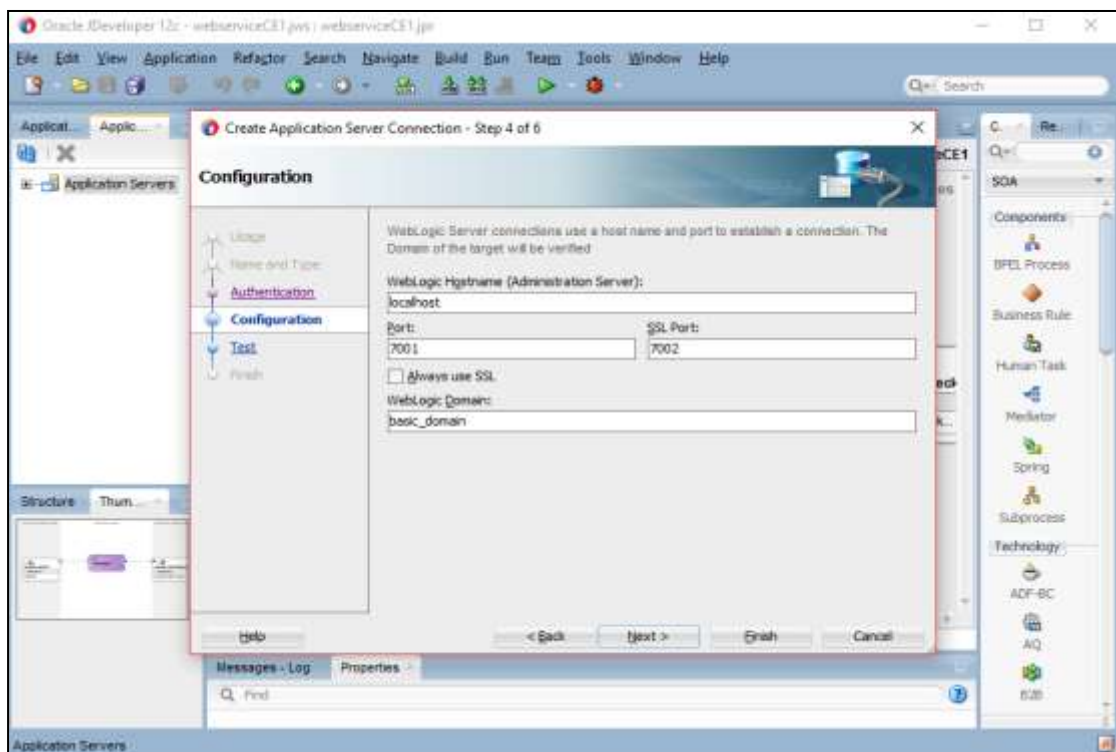
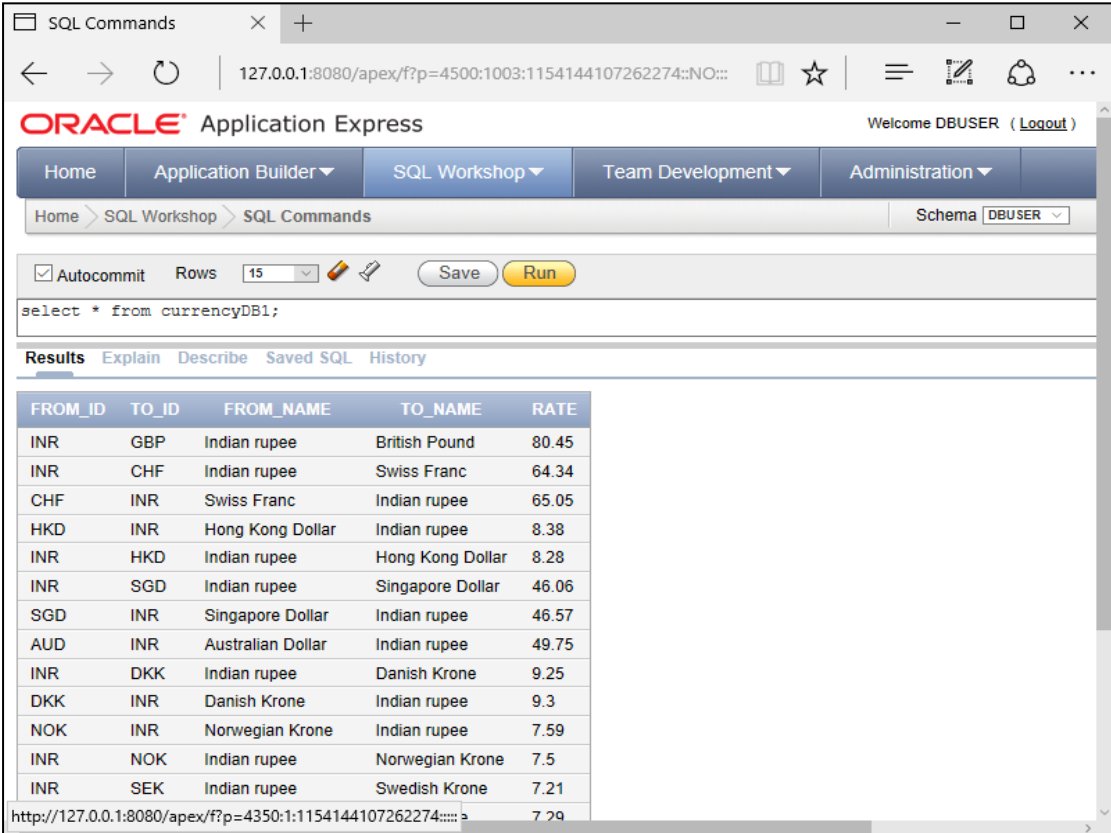


Fig. 4.14: Configuration of WebLogic Server

Test connection determines that information specified in the configuration can successfully establish a connection with the application server.

4.7 Currency Exchange Database

Oracle 11g XE database is used to create database of currency exchange SOA application. This database contains the tables such as exchange_rate, to_currency_name, from_currency_name, to_currency_code and from_currency_code as a composite primary key. From_id and to_id specifies the currency code of the currency. From_name specifies the name of the currency and rate value specifies the exchange rate of the currency from from_id to to_id. We create a composite primary key of from_id and to_id attributes as shown in Fig. 4.15. Service consumer gave the input from currency name and to currency name values and send the request to the currency exchange application, then application sends the exchange rate value as a response to service consumer. All the services gets consolidate to form the Currency Exchange application.



The screenshot shows the Oracle Application Express interface. The SQL Commands window displays the query: `select * from currencyDB1;`. The results are shown in a table with the following data:

FROM_ID	TO_ID	FROM_NAME	TO_NAME	RATE
INR	GBP	Indian rupee	British Pound	80.45
INR	CHF	Indian rupee	Swiss Franc	64.34
CHF	INR	Swiss Franc	Indian rupee	65.05
HKD	INR	Hong Kong Dollar	Indian rupee	8.38
INR	HKD	Indian rupee	Hong Kong Dollar	8.28
INR	SGD	Indian rupee	Singapore Dollar	46.06
SGD	INR	Singapore Dollar	Indian rupee	46.57
AUD	INR	Australian Dollar	Indian rupee	49.75
INR	DKK	Indian rupee	Danish Krone	9.25
DKK	INR	Danish Krone	Indian rupee	9.3
NOK	INR	Norwegian Krone	Indian rupee	7.59
INR	NOK	Indian rupee	Norwegian Krone	7.5
INR	SEK	Indian rupee	Swedish Krone	7.21
http://127.0.0.1:8080/apex/f?p=4350:1:1154144107262274:::...				7.29

Fig. 4.15: Database of CurrencyDB1

4.8 Data Source Connection at Weblogic Server

To receive a database connection, we must have a Java Database Connectivity (JDBC). Data source enables java database connectivity to client. The data source has a connection pool that consist a collection of database connections. Whenever an application requires a connection, it requests a connection from the data source. Application uses the connection to access the database and after the use it close the connection. Closing the connection means data source can place the connection in the connection pool, so that it can be further used when needed. We can create data source as one of the instance of weblogic server configuration. We deployed the data source on the weblogic server. To fill the connection pool with connections weblogic server creates database connections.

In weblogic server administration console 12c, under the services create a new generic data store. Specify the name of the data source and JNDI name shown in Fig. 4.16.

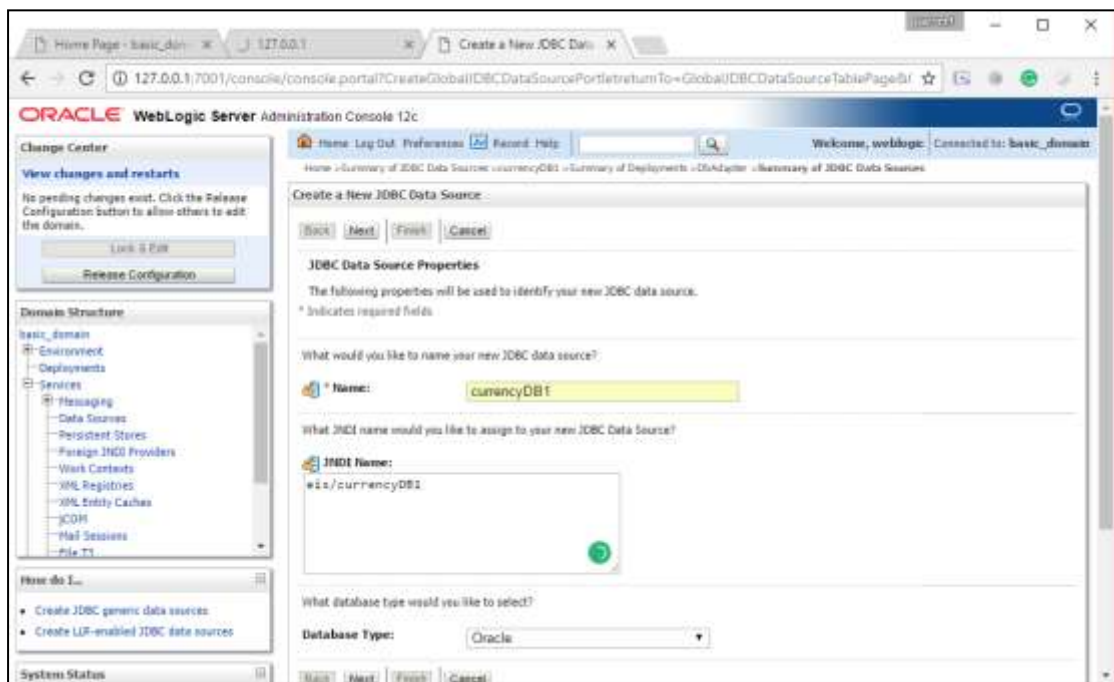


Fig. 4.16: Data Source Properties

In connection properties provide the database details, database driver is Oracle's Driver (Thin) for Instance connections and database credentials. If a connection to the database can be made then test configuration must be succeeded. We can specify the

targeted server for data source. Data source is available as one of the resource of the targeted server and data source is managed by the targeted server. Data source is shown in the data source under services as shown in Fig. 4.17.

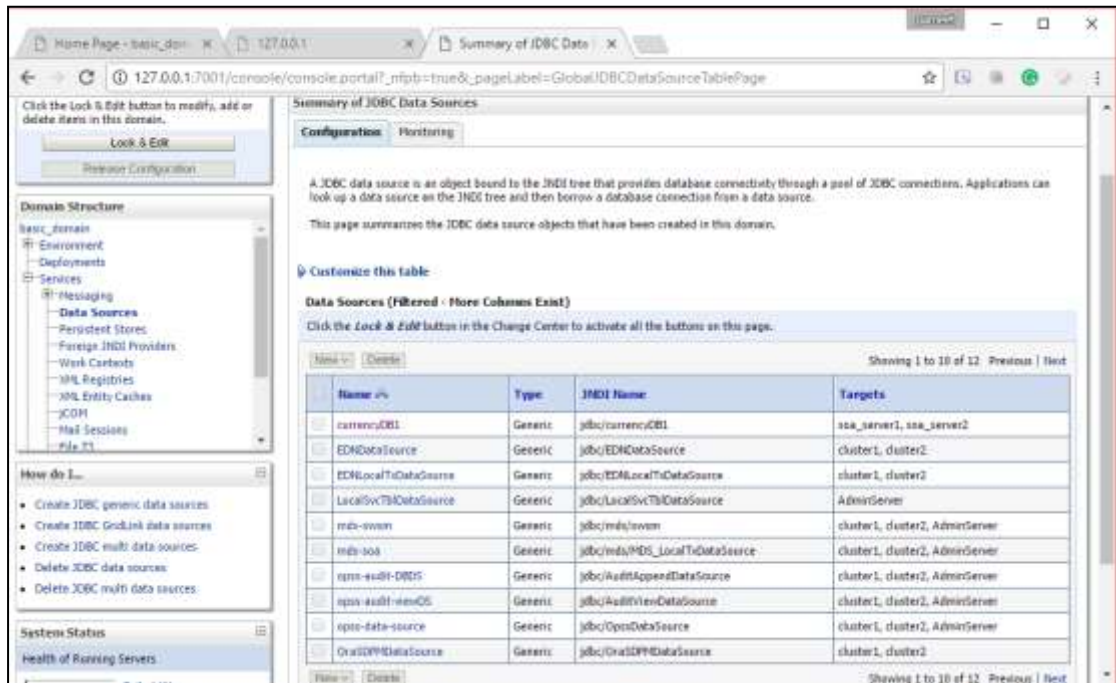


Fig. 4.17: Data Source List

We can create an outbound connection pool under deployment's dbadapter. Select outbound connection group and specify the JNDI name that is used to obtain the connection pool at runtime. Property value is specified for the data source in outbound connection pool properties as shown in Fig. 4.18.

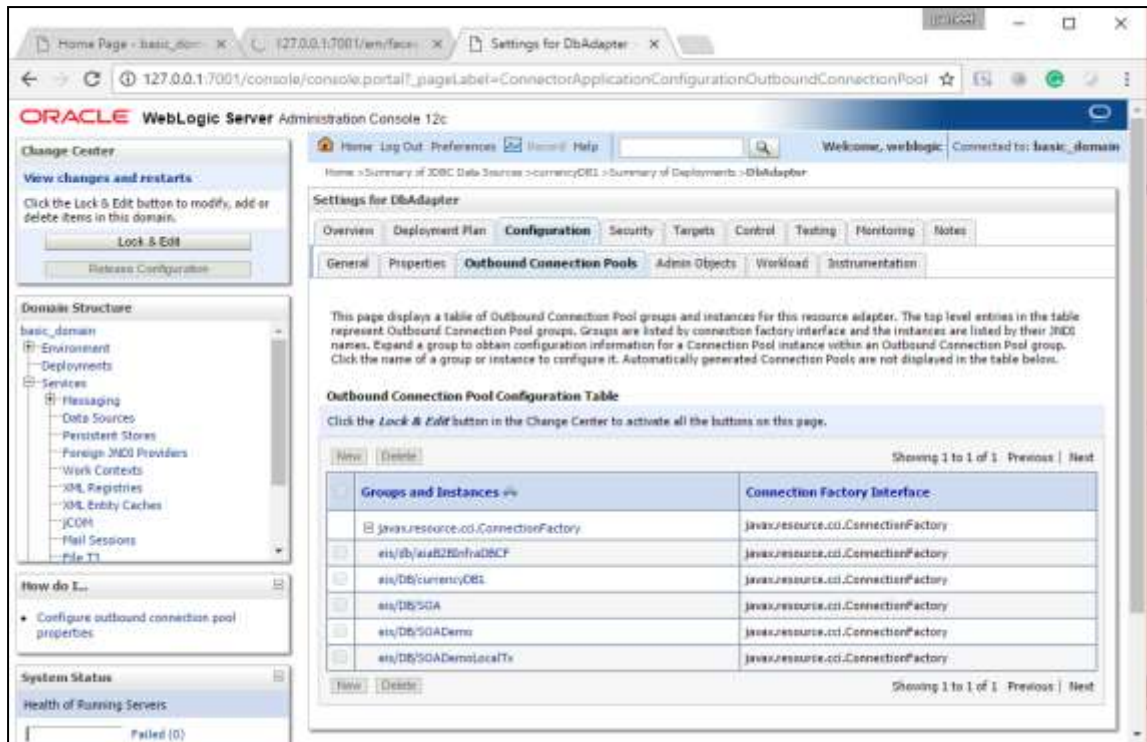


Fig 4.18: Outbound Connection Pool

4.9 Currency Exchange SOA Service Based on SOAP

4.9.1 XML Schema for SOAP based SOA Service

Oracle SOA Suite 12c is used to develop SOAP based currency exchange SOA service. We can create a SOA project in the SOA suite 12c. Create a XML schema for the service. XML schema is a container. We will take input and invoke database from other place and the response came to the XML container. This contains XSD input elements and XSD output elements and the data type of these elements. Target namespace defines how we can access the XML schema. One XML schema is created for the database automatically. Design view of XML schema is shown in Fig. 4.19.

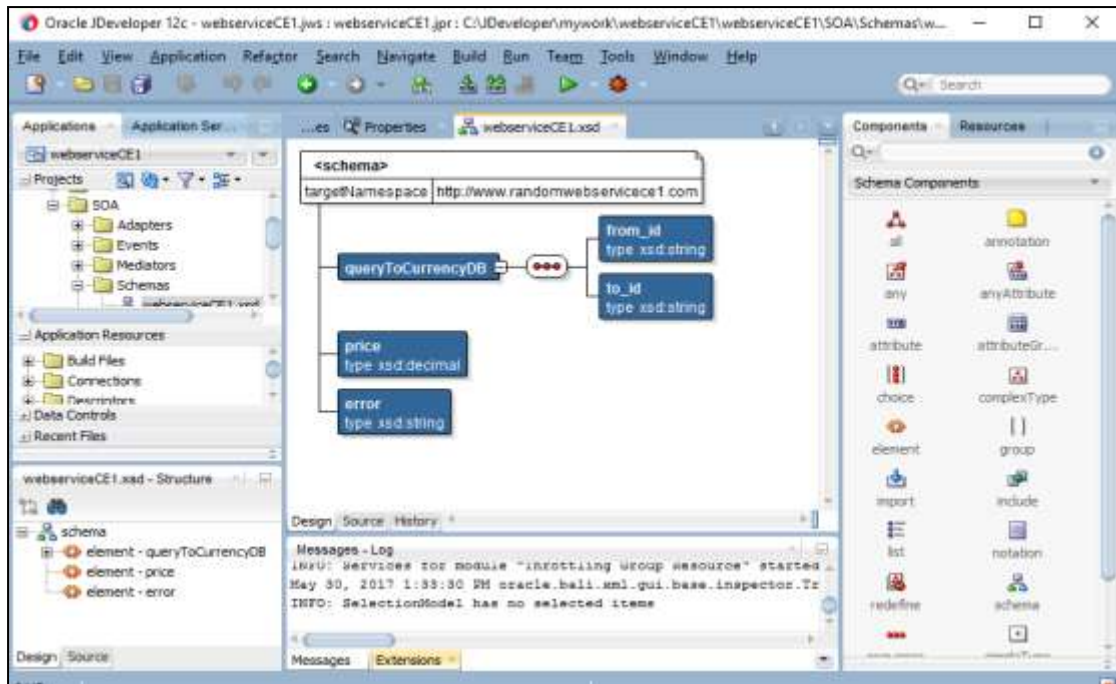


Fig. 4.19: Design View of XML Schema

4.9.2 BPEL Process

Select BPEL process and place it into components. Web services can interact and share data using BPEL process. BPEL allows developers to develop processes by defining the order in which services will be invoked. BPEL process consolidates the SOAP based Currency Exchange SOA service. In our SOA service, we give the request to the SOA service and get the response and we do not do anything in between. So our SOA service interface should be synchronous as shown in Fig. 4.20.

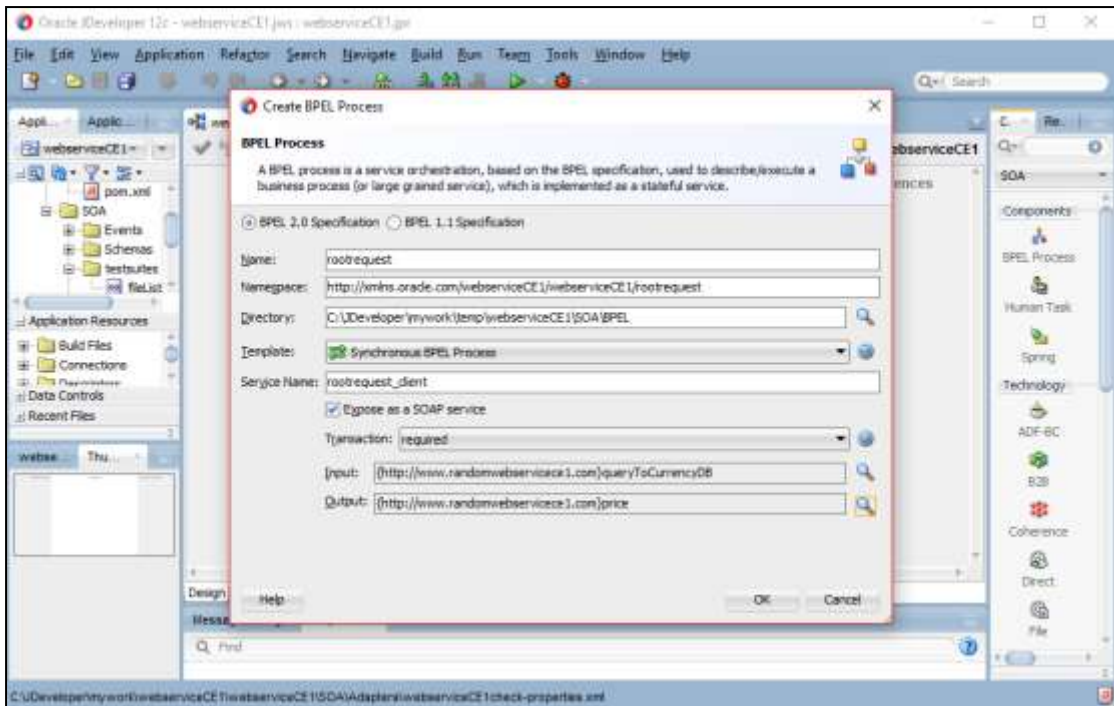


Fig. 4.20: BPEL Process

Select SOAP from components in the exposed services to create a SOAP service. Import XML schema file under schema folder of the created project. WSDL URL is generated from the schema. Access the XML schema for input and output. For input, add message part to the web service and choose its type as XSD input element from the project XML schema file. For output, add message part to the web service and select its type from XML schema file as shown in Fig. 4.21. Similarly for fault add message part. After assigning input and output, WSDL URL is generated.

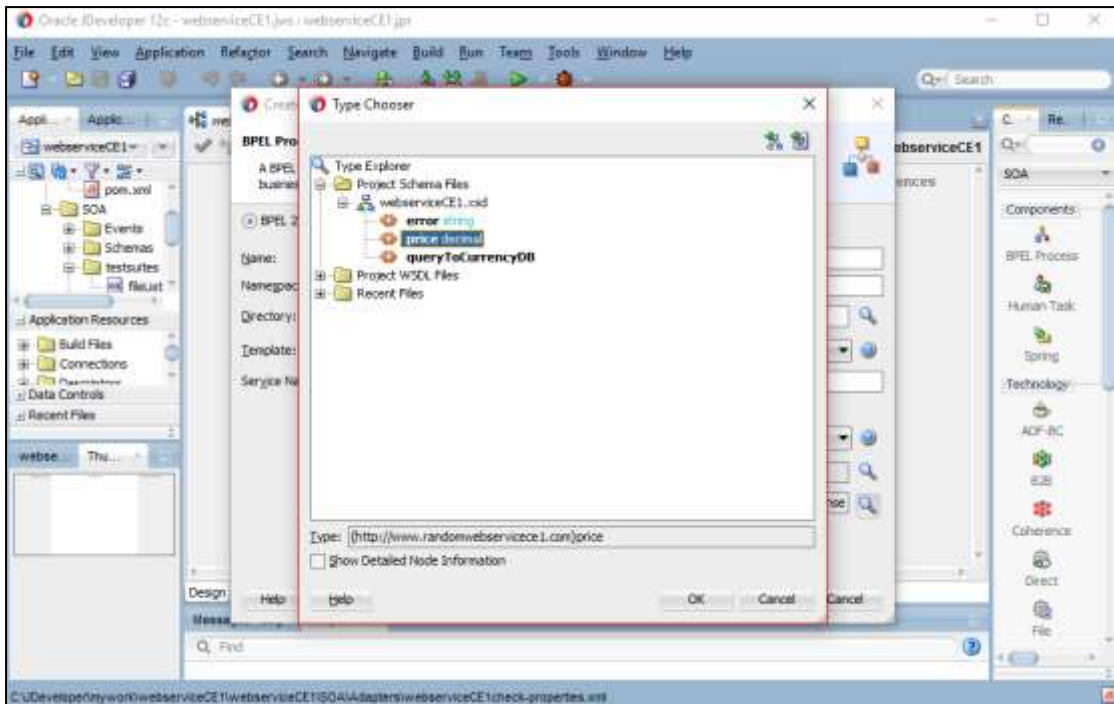


Fig. 4.21: Output Element of BPEL Process

4.9.3 Database Adapter

For database, select database from components in the external references. In connection name, give the one that we setup as a data source name in weblogic server. We can select the data source name from JNDI tree through JNDI browser or write it manually. Provide a username, password of Oracle workspace account. Driver type, port number and host name of the database is provided to create a successful test connection as shown in Fig. 4.22.

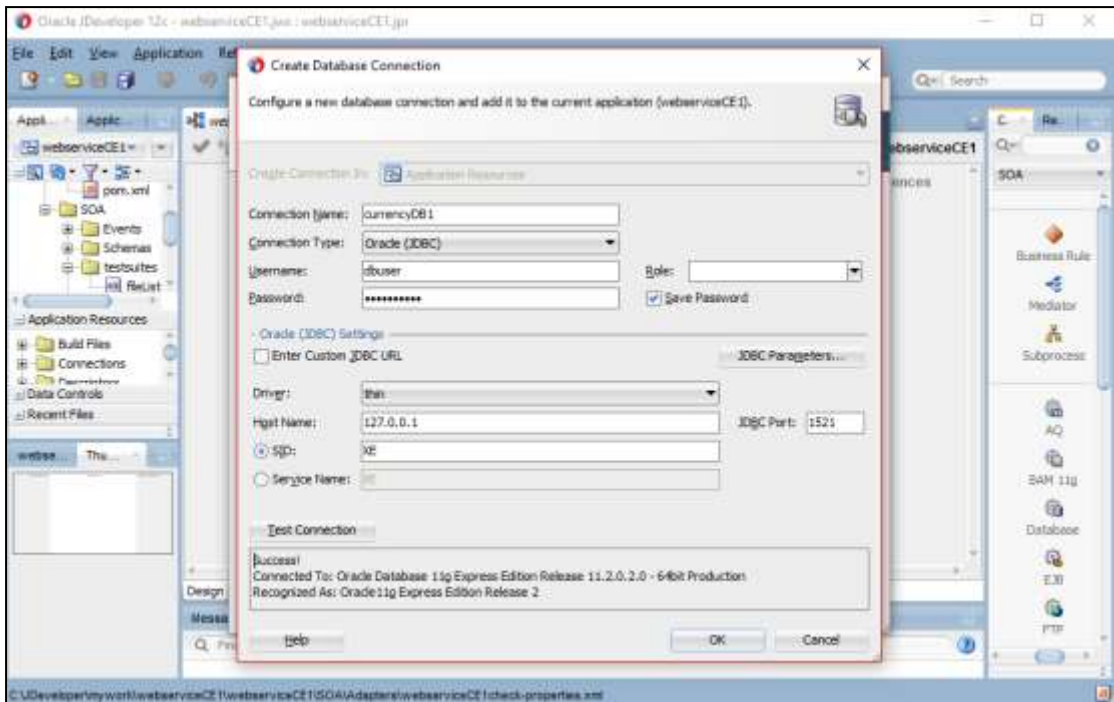


Fig. 4.22: Database Connection Details

After the successful test connection, it fetches the data source path in the JNDI name as shown in Fig. 4.23.

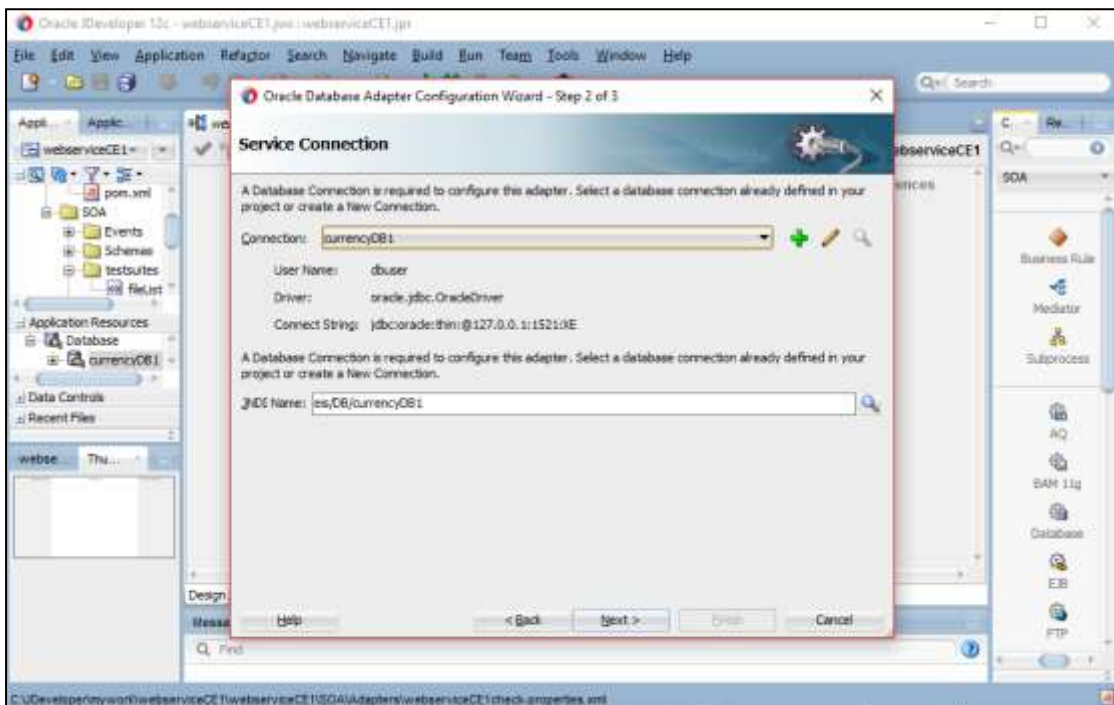


Fig. 4.23: Service Connection Detail

Define the operation and then import the table from the database schema. We can select the primary key of the tables if it is not defined earlier while table creation. In attribute filtering, select those attributes that are required in operation clause. Add parameters name and add query while defining the selection criteria. Add arguments as query key. The input parameter of external reference is linked to the input parameter of the exposed services. After successful creation of database connectivity external service is ready.

4.9.4 Integrate BPEL Process and Database Adapter

We can integrate BPEL process and database adapter by adding a wire mediator to the SOAP service and BPEL process and invoking the database adapter. Plan will help us to decide which variable of SOA service is mapped to the variable of database adapter and how should it mapped. We have routing rules in the plan. On clicking transformation, we can create a new mapper file for input. Similarly a new mapper file is created for the synchronous reply as shown in Fig. 4.24.

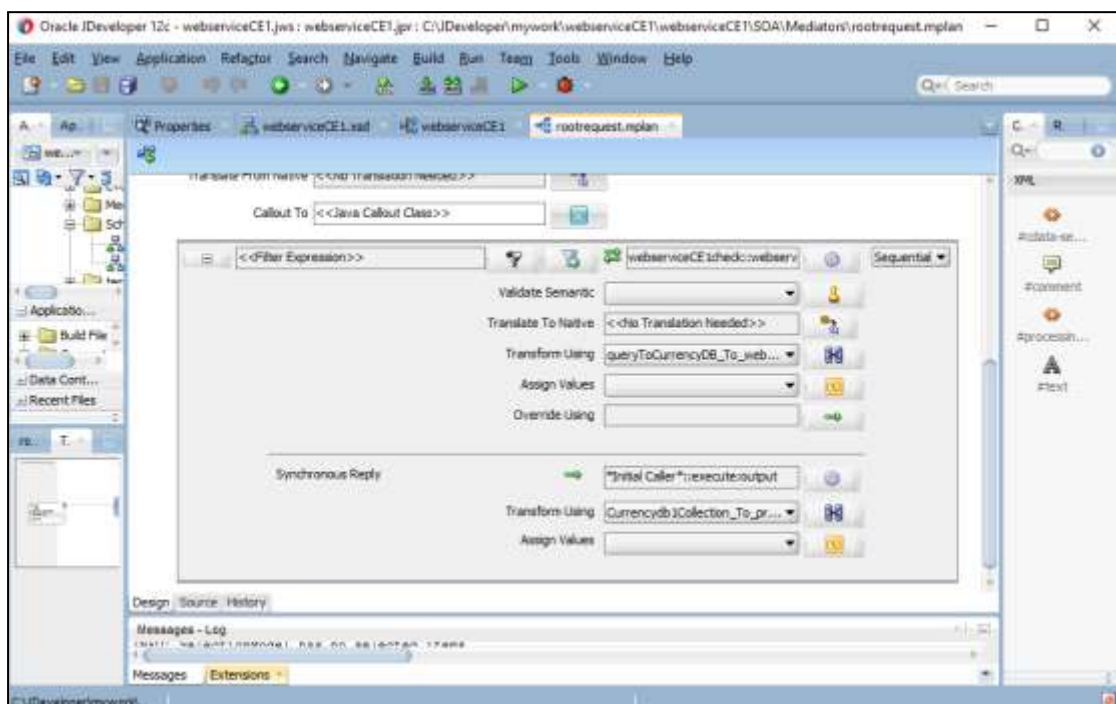


Fig. 4.24: Routing Rules

We can map the XML input to the database input and we can map the database output to the XML output. We can create transformation map for the input. Select the input of XML tree and map it to the user defined input variable as shown in Fig. 4.25.

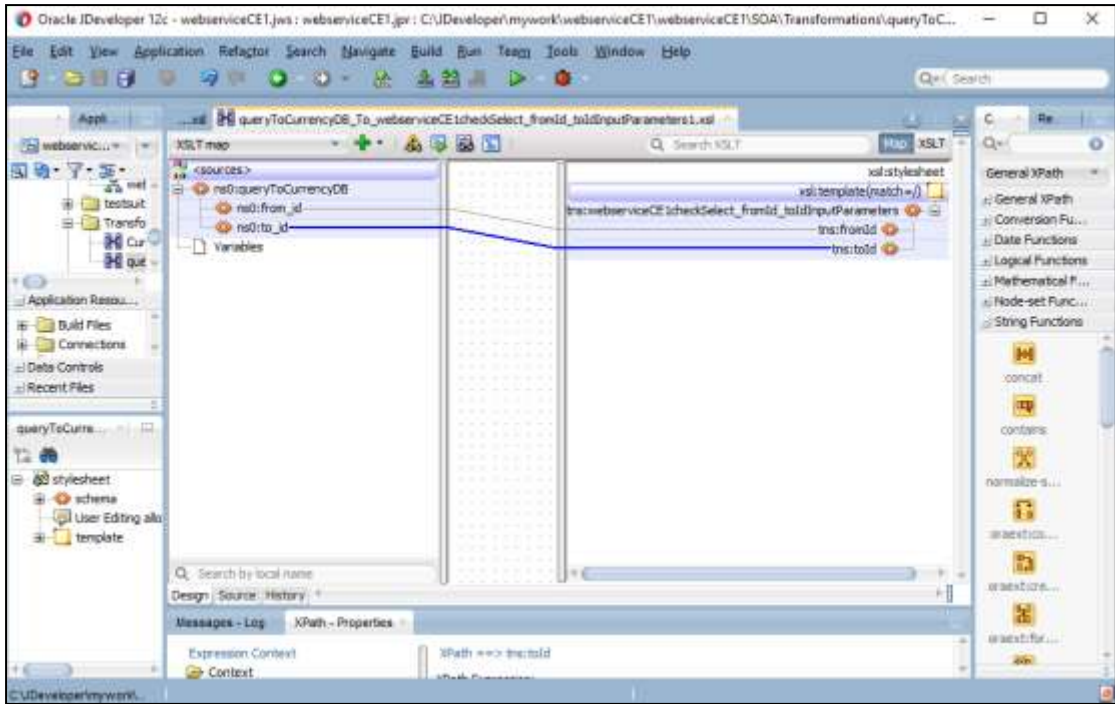


Fig. 4.25: Request Transformation Map

XML tree of the database is automatically generated. Database output variable is mapped to the XML output variable. The price value of one currency to another currency from database output variable is mapped to the output variable of the web service as shown in Fig. 4.26.

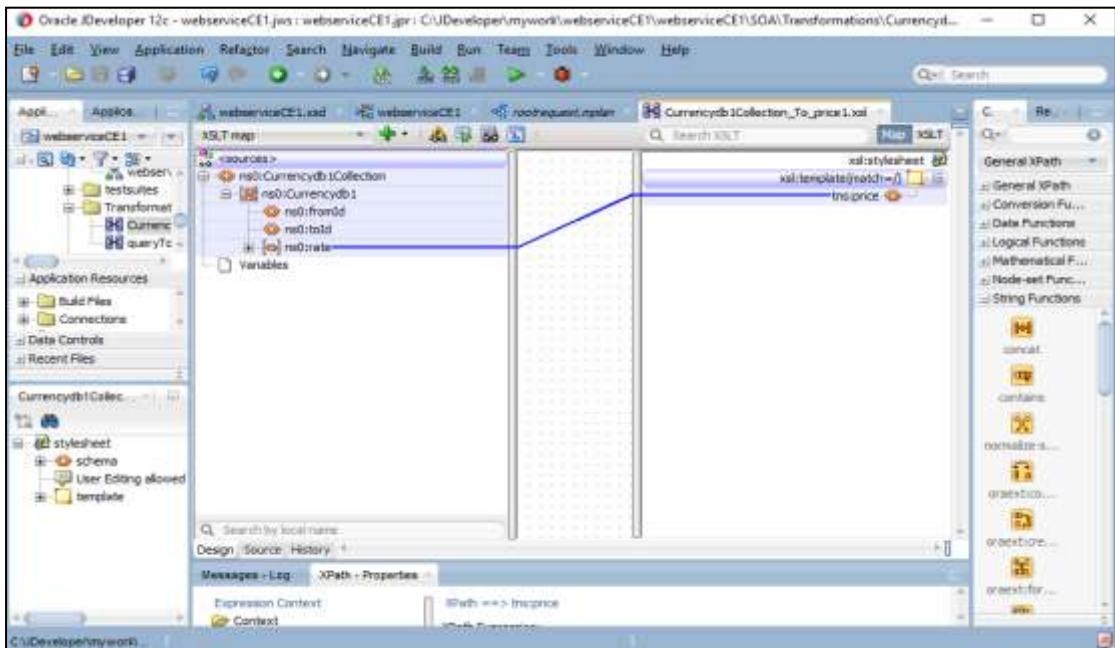


Fig. 4.26: Reply Transformation Map

4.9.5 Deploy SOAP based SOA Service

There are two options to deploy SOAP service to the server. We deploy the SOA service through the Oracle Jdeveloper 12c or we can deploy the SOA service through Oracle Enterprise Manager. We deploy SOAP based currency exchange SOA service in Oracle Jdeveloper 12c. We provide the revision id to the SOA service. We can select the server on which we want to deploy our service as shown in Fig. 4.27.

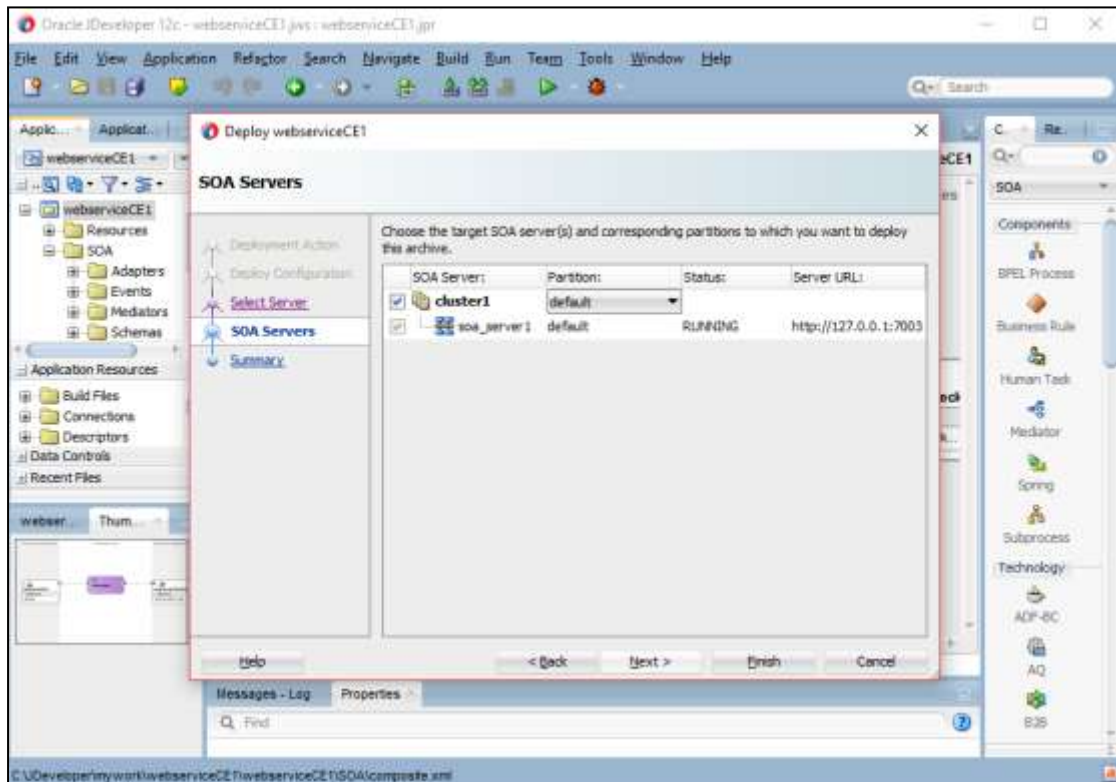


Fig. 4.27: Deploy Currency Exchange Web Service based on SOAP

If a SOA service is successfully deployed on the server without showing any error in the message section, a WSDL address is created by the SOA server. By using WSDL address, services communicate with each other by exchanging SOAP messages. The WSDL address that is generated after running the web service is known as SOAP endpoint.

4.10 Currency Exchange SOA Service Based on REST

4.10.1 XML Schema for REST based SOA Service

We can create a SOA project in the SOA Suite 12c. Create a XML schema for the REST based SOA service as specified in section 4.9.1. When XML schema is created successfully, it automatically generates a XML schema for the database.

4.10.2 BPEL Process

BPEL allows developers to define the order in which services are invoked. BPEL process consolidates the REST based Currency Exchange SOA service. We can create a BPEL process for REST based SOA service as specified in section 4.9.2. Select REST from components in the exposed services to create a SOA service. After assigning input and output, a WSDL URL is generated.

4.10.3 Database Adapter

We can add database adapter by selecting database from the components. Provide the data source name that we configured in the weblogic server console. Provide a username, password of Oracle workspace account and driver type, port number and host name of the database is provided to create a successful test connection as specified in section 4.9.3. After successful creation of database connectivity external service is ready.

4.10.4 Create REST Binding

We can provide external connectivity for REST over HTTP using REST binding. REST binding supports HTTP headers and parameters. REST supports main four methods such as GET, POST, PUT and DELETE methods. Various message formats such as XML, JSON and plain text format. Provide the REST binding name and description and then add a resource in the resource pane. In the relative path field specify the name and path as shown in Fig. 4.28.

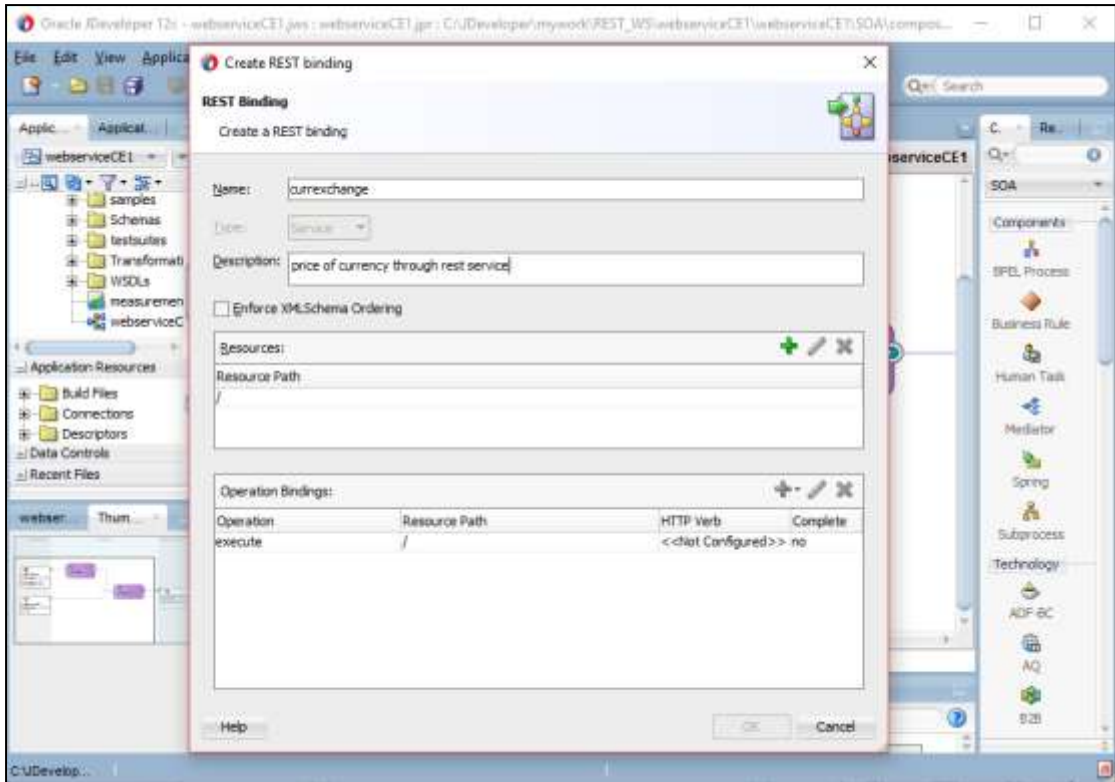


Fig. 4.28: REST Binding

4.10.5 Adding a GET and POST Operation Binding

We can add operation binding by providing operation name, resource, description and type as service. We must provide HTTP verb as GET. Now we have to add request and response variables. In response field choose the type of variable from project schema file. Select the field that are required as shown in Fig. 4.29.

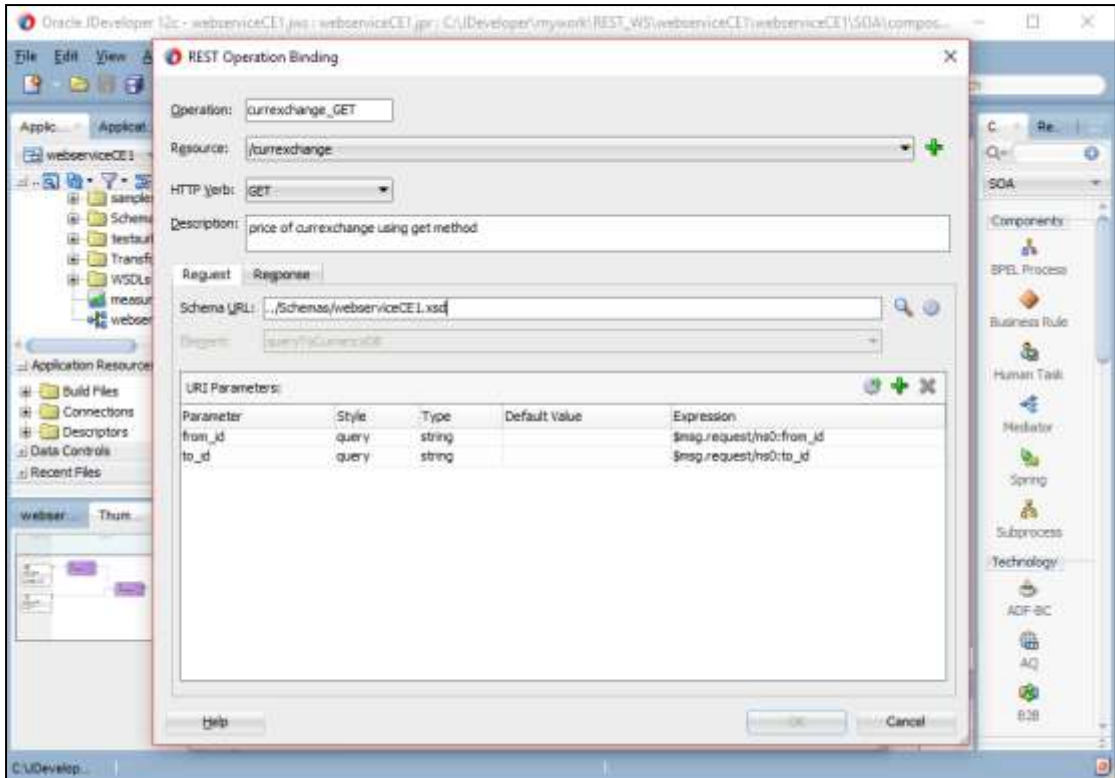


Fig. 4.29: Operation Binding for GET Method

Similarly add response value and add HTTP status is 200 and select the payload message format. Now add operation binding for POST method. Select HTTP verb as POST. Choose the request and response from the project schema file for POST method same as we choose for GET method. In operation binding of REST SOA service, we have both GET and POST method as shown in Fig. 4.30.

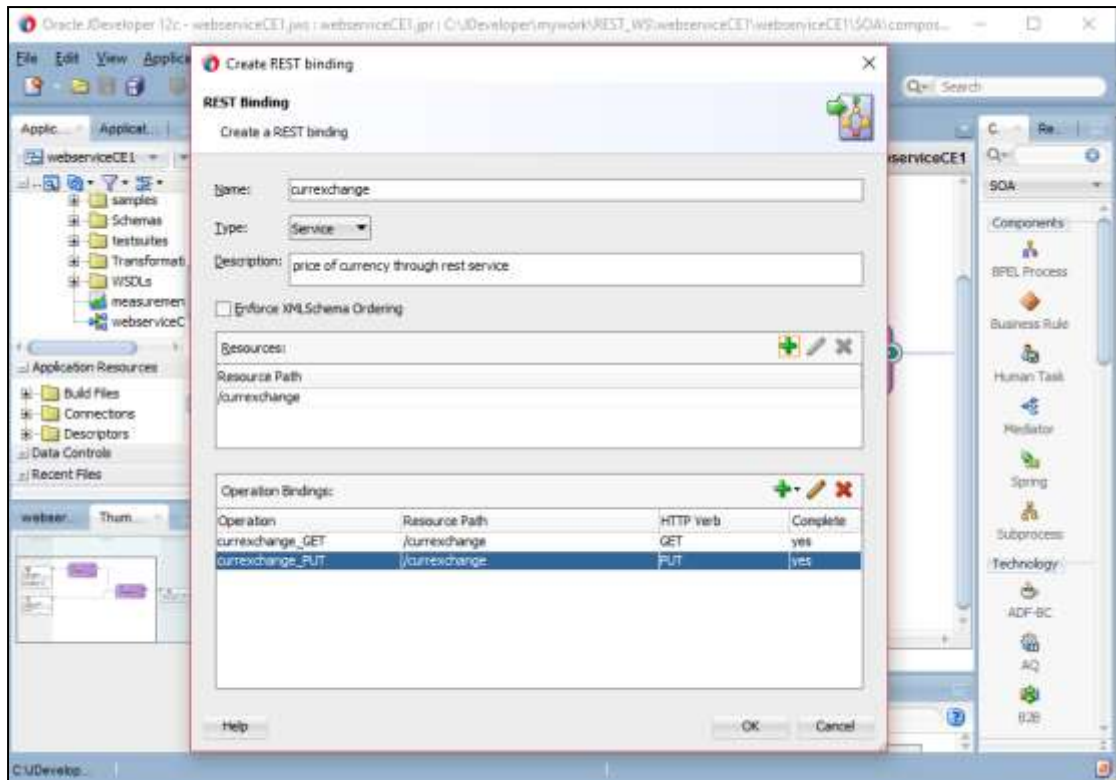


Fig. 4.30: Operation Binding for PUT Method

4.10.6 Adding a Mediator with Routing Rules

From components select mediator and provide name, directory and template to the mediator. Wire the GET and POST method to the mediator to add routing rules to these methods. In the mplan page of the mediator, add routing rules to GET and POST methods. In mplan file, create a new routing rule for GET method. Target type can be existing service and expand the mediators then services and select execute as shown in Fig. 4.31.

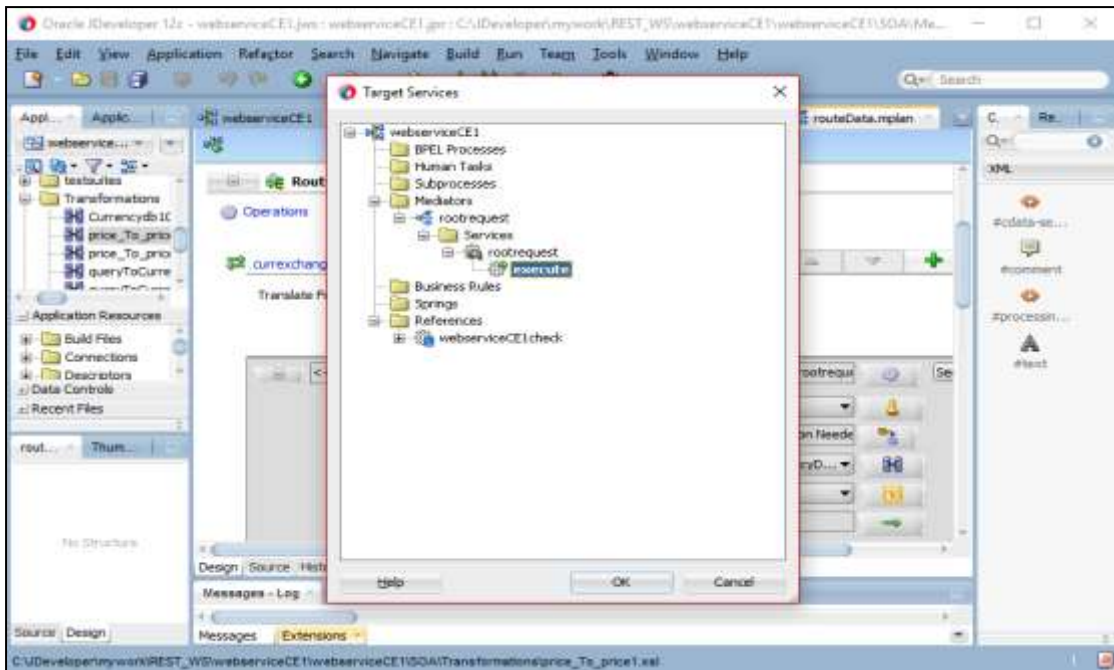


Fig. 4.31: Target Services for Routing Rules

Create a mapper file for request transformation map. In transformation map, select type as XSLT and default values and the mapper file is created as shown in Fig. 4.32.

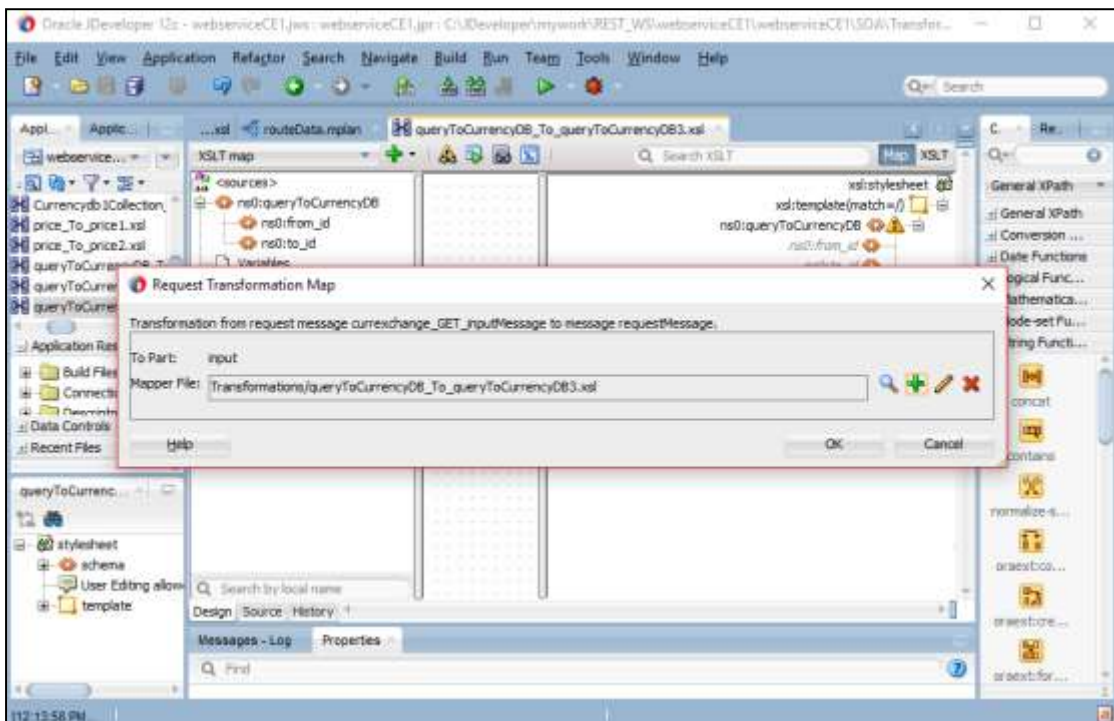


Fig. 4.32: Mapper File for Request

We can map the XML input for GET method to the database input and the database output to the XML output. Select the input of XML tree and map it to the user defined input variable. Map the from_id key of the XML input to the from_id key of the database. Similarly map the to_id key of the XML input to the to_id key of the database. Create a mapper file for the synchronous reply.

Now add a routing rule to the mediator for the POST operation. In the mplan file, scroll down to the POST pane and add a new routing rule for POST method. Similarly select the target services and target type as chosen in the GET method as shown in Fig. 4.33.

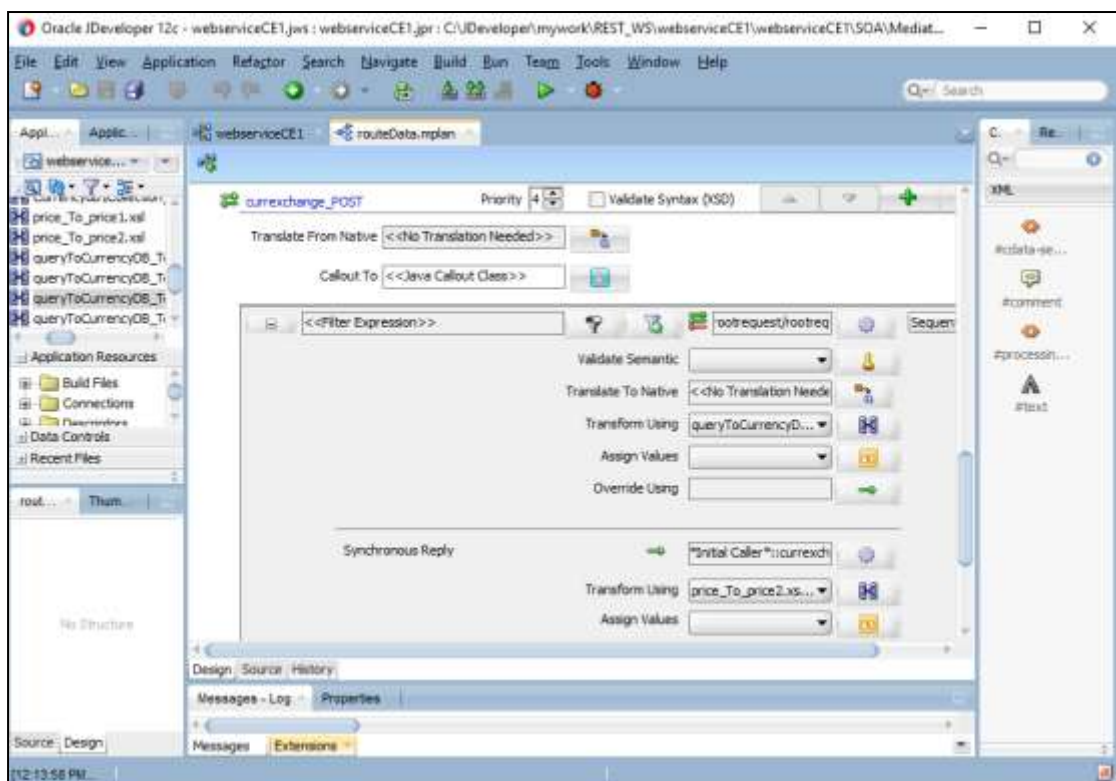


Fig. 4.33: Transformation Rules

Create a request transformation map and create a request transformation mapper file and map the ns0:from_id element of the XML file for POST method to the database input element ns0:from_id. Similarly map to_id input element of the XML file to the to_id input element of the database as shown in Fig. 4.34.

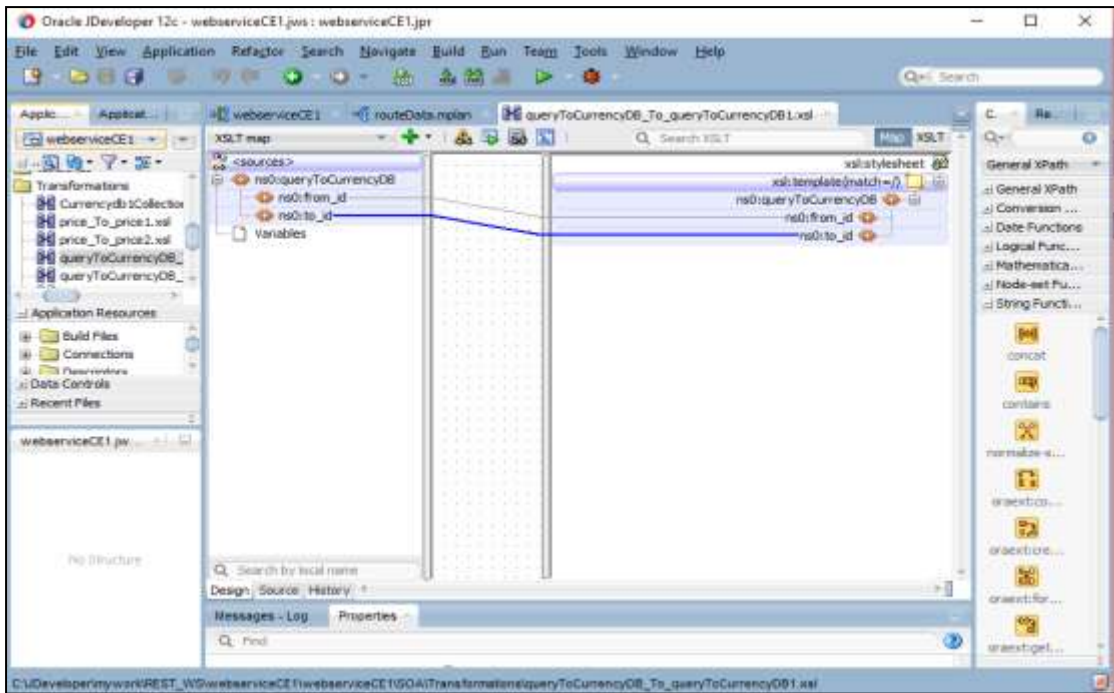


Fig. 4.34: Request Transformation Map

Create a mapper file for the reply transformation map. Map the output element ns0:price of the database to the output element ns0:price of the SOA service. After providing the routing rules to the both GET and POST method, it can automatically its wiring with root request as shown in Fig. 4.35.

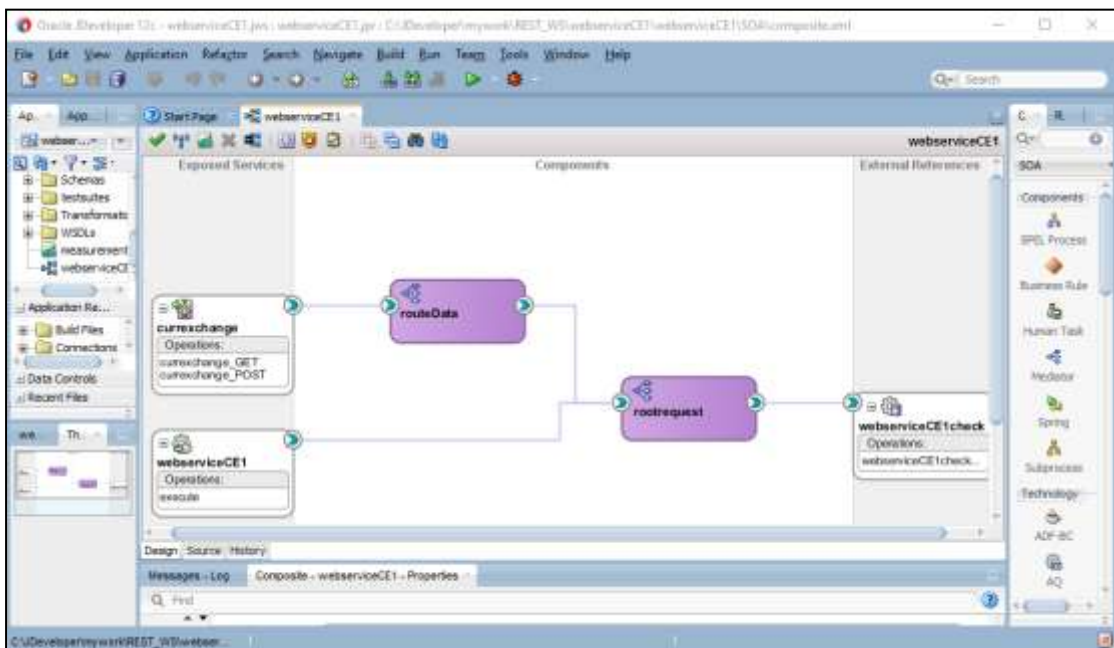


Fig. 4.35: Design View of REST SOA Service

4.10.7 Deploy REST based SOA Service

We deploy the REST based SOA service in Oracle Enterprise Manager 12c. Under target navigation, select soa- infra from SOA. Select deploy under SOA deployment. Upload the sca_webserviceCE1_rev1.0.jar file. We can select a target managed server or cluster to which we deploy the SOA service. Select default from configured SOA partition. Composites are logical grouped into partitions. Large deployments are managed through partitions. Choose default revision id for the web service as shown in Fig. 4.36. On the successful deployment of web service it can displayed in the Oracle administration console 12c under soa- infra in target navigation.

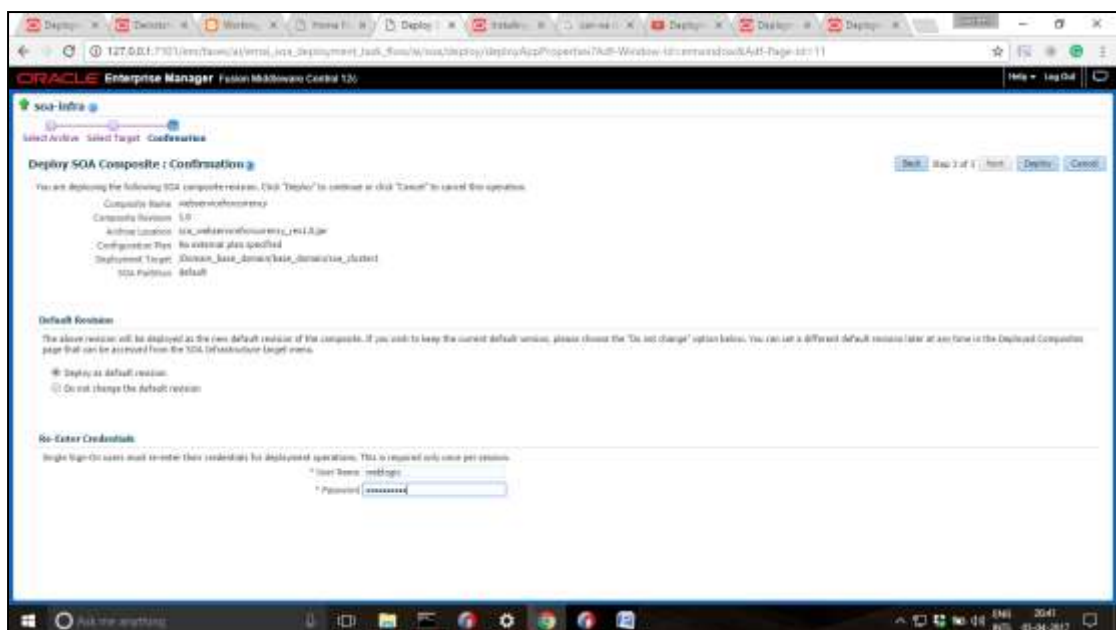


Fig 4.36: Deploy Currency Exchange SOA Service based on REST

If a web service is successfully deployed on the server without showing any error in the message section, a WSDL address is created by the SOA server.

5.1 Currency Exchange SOA Service based on SOAP

The final phase SOA web service is deployment and execution of the SOA services. Response time is calculated for SOAP based SOA service and REST based SOA service. We can test the currency exchange SOA service based on SOAP technology in Oracle JDeveloper 12c. We can select the SOA service and click on test. We provide the payload in the specified fields and then click on send request. After successful testing of the SOA service, output is displayed as shown in Fig. 5.1.

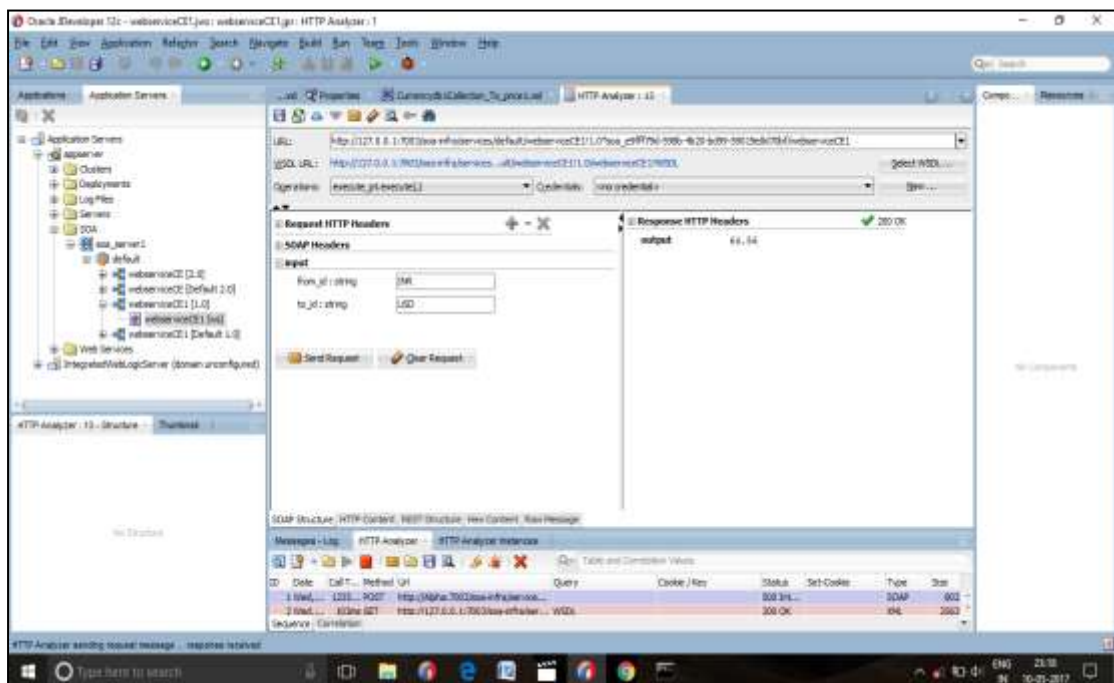


Fig. 5.1: Output of Currency Exchange SOA Service based on SOAP

5.2 Currency Exchange SOA Service based on REST

We can test the currency exchange SOA service based on REST technology in Oracle Enterprise Manager 12c, we can select the SOA service and provide the payload in the specified fields and then click on send request. After successful testing of the SOA service, output is displayed as shown in Fig. 5.2.

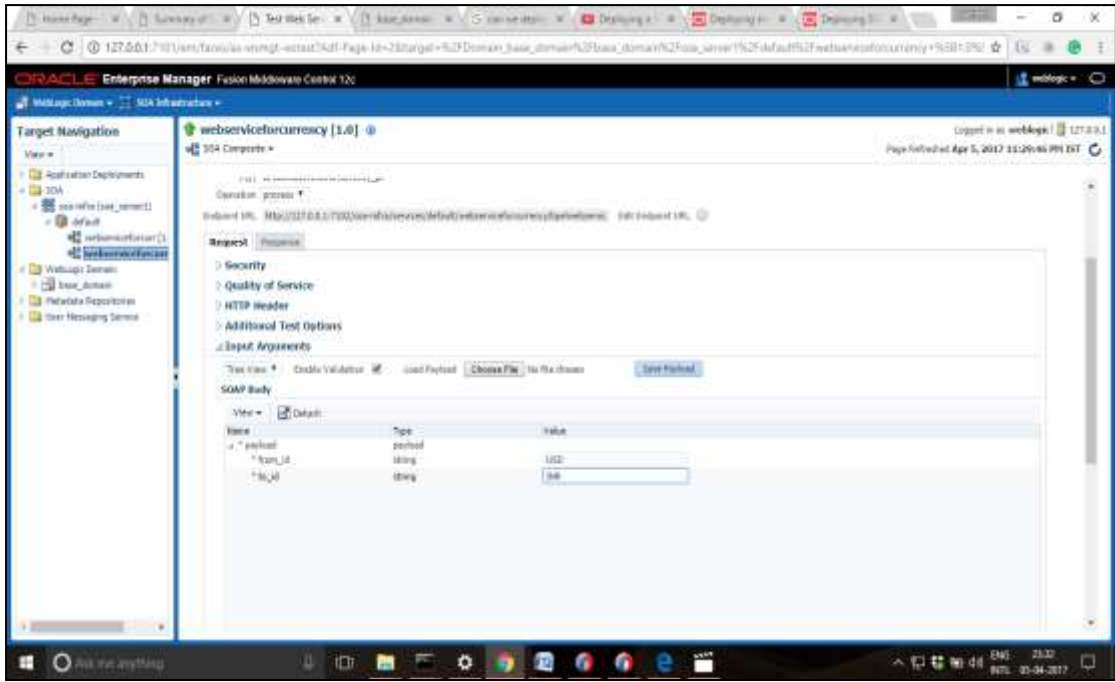


Fig. 5.2: Output of Currency Exchange SOA Service based on REST

We provide a request with from_id and to_id values and request for the currency exchange price value. The service fetches the information about the price value from the database and sends the response to the response to the client in terms of price value. Response time values of SOAP and REST for composite service i.e. Currency Exchange SOA Service is shown in Table 5.1.

Table 5.1: Service Response Time for REST and SOAP Methods

No. Of Clients	Response Time (in ms)	
	SOAP	REST
1	2723	76
2	2828	166
3	3988	228
4	4921	1881
5	4994	1995

Comparison of response time for SOAP and REST for composite service i.e. Currency Exchange SOA Service is shown in Fig. 5.3.

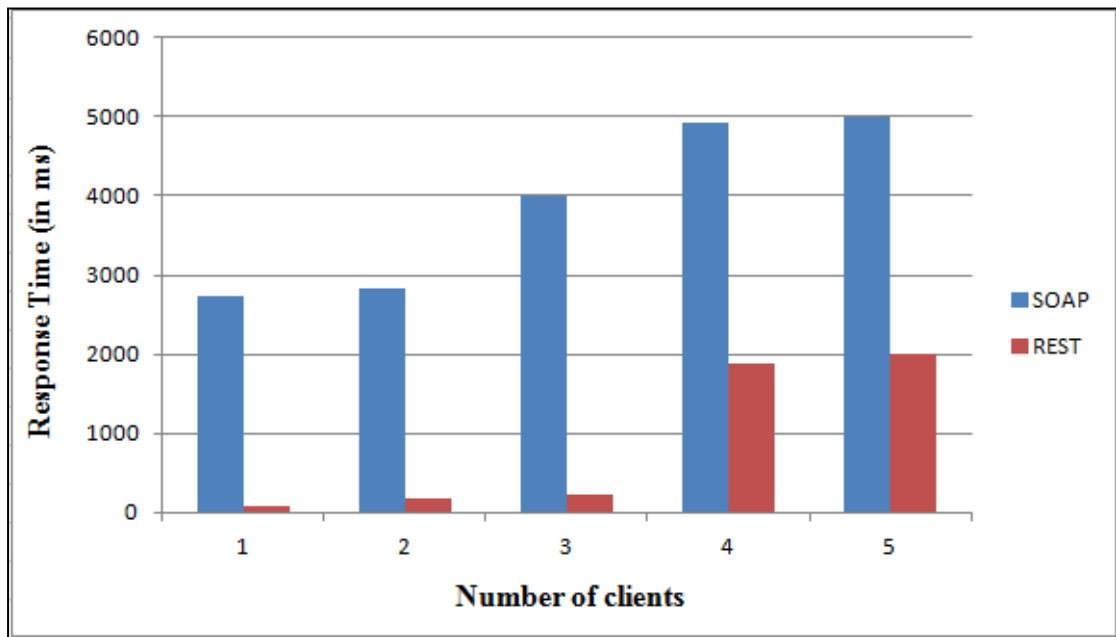


Fig. 5.3: Response Time for SOAP and REST

Response time is computed in milliseconds (ms) in Fig 5.3. The number of clients in the x-axis shown and the y-axis shows the response time. Since the numbers of clients has increased in SOAP, response time increases. Increasing the number of clients does not affect response time of REST service in a drastic manner. It means that large number of requests can handled by REST service in the more efficient way than the SOAP service. As the number of simultaneous clients increased, currency exchange SOA service based on REST service has a better response time than SOAP based currency exchange SOA service.

6.1 Conclusion

Service Oriented Architecture (SOA) is a constitution of sovereign services that groups together to design service oriented business process. Services are the core components to build service oriented architecture. SOA established on the principle of loose coupling, service reusability, service discoverability, service abstraction, service statelessness, platform independence and service autonomy. SOA assures huge benefits such as rapid application development, parallel and independent development, fast response to business services, increased reusability, simplified maintenance and reduced costs. Along with benefits SOA has many flaws like security, web service open standards, governance, application ownership, organizational issues and metadata management issues.

In this thesis, we have discussed the potential benefits, limitations and inherent issues in detail. We have considered the Currency Exchange example for evaluating Enterprise service bus (ESB) architecture, messaging flow. Oracle SOA suite 12c tool is used for building Currency Exchange application. Oracle XE 11 g database is used to create database for currency exchange application. By using the Repository Creation Utility (RCU), we can create and load the database schemas for SOA in the database. Oracle web logic domain is configured to create one server instance as an administration server and create managed SOA servers and node manager. Start an administration server, SOA servers and node manager to run the applications. Configured servers are launched in Oracle Jdeveloper. To receive a database connection, we must have a Java Database Connectivity (JDBC). Data source enables java database connectivity to client. The data source has a connection pool that consist a collection of database connections.

We have created Currency Exchange service with two methods, one is currency exchange SOA service based on SOAP and another is currency exchange SOA service based on REST. We create XML schema, BPEL process and database adapter for the currency exchange web service. We integrate the BPEL process and database adapter. BPEL process consolidates the Currency exchange application. In which

sequence or order the services are consolidated in the BPEL process, same sequence of services is followed to access the services. We deploy the both currency exchange SOA services based on SOAP and REST on SOA server in Oracle enterprise manager 12c or Oracle jdeveloper 12c. There are sufficient tools available to build SOAP services. So it is convenient to build SOAP based application. We provide the payload to the SOA services to get the output as a price value of currency exchange. We compare the both SOA services in terms of response time. Since currency exchange SOA service based on REST has a better response time than SOAP service, the increase in the number of clients simultaneously.

6.2 Future Scope

SOA is the succeeding enormous thing for future integration of enterprises. It is continually advancing as it gets embraced increasingly by enterprises and organizations. In future, SOA would significantly affect the advancement of big business application incorporation foundation as more and more technologies and vendors start providing solutions. For security purpose, authentication and identification of the user can be provided by matching algorithm using fingerprint matching, face and eye etc. In future caching efficiency of the web services and security of web services can be evaluated for SOA. It is extended in financial sector by providing banking service like issuance of check books, provides personal and commercial loans and insurance services like insurance brokerage and reinsurance, bill discounting and credit cards.

REFERENCES

- [1] M. Knutsson and T. Glennow, “Challenges of service-oriented architecture (soa)-from the public sector perspective,” 2015.
- [2] Y.-P. Huang, “The service oriented architecture in decision support system,” in *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on*. IEEE, 2016, pp. 868–872.
- [3] <http://searchmicroservices.techtarget.com/definition/BPEL-Business-Process-Execution-> (accessed on 12/04/2017).
- [4] M. A. Al Sheikh, H. A. Aboalsamh, and A. Albarrak, “Migration of legacy applications and services to service-oriented architecture (soa),” in *Current Trends in Information Technology (CTIT), 2011 International Conference and Workshop on*. IEEE, 2011, pp. 137– 142.
- [5] R. Ramanathan and T. Korte, “Software service architecture to access weather data using restful web services,” in *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*. IEEE, 2014, pp. 1– 8.
- [6] X. Dan, Y. Shi, Z. Tao, J. Xiang-Yang, L. Zao-Qing, and Y. Jun-Feng, “An approach for describing soa,” in *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006. International Conference on*. IEEE, 2006, pp. 1–4.
- [7] N. Basias, M. Themistocleous, and V. Morabito, “An investigation of benefits affecting soa adoption in ebanking,” *International Journal of e-Education, e-Business, e-Management and e-Learning*, vol. 4, no. 3, p. 174, 2014.
- [8] O. Fakorede, “An investigation into the implementation issues and challenges of service oriented architecture,” Ph.D. dissertation, Bournemouth University, 2007
- [9] Z. Mahmood, “The promise and limitations of service oriented architecture,” *International journal of Computers*, vol. 1, no. 3, pp. 74–78, 2007.
- [10] M. Waris, S. A. Khan, and M. Z. Fakhar, “Factors effecting service oriented architecture implementation,” in *Science and Information Conference (SAI), 2013. IEEE, 2013*, pp. 1–8.
- [11] P. Kaur and H. Singh, “An analytical review of quality attributes of service-oriented architecture,” *Trends in Information Management*, vol. 10, no. 1, 2015.

- [12] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end qos constraints," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, p. 6, 2007.
- [13] S. Shrivastava, A. Sharma, and D. Shrivastava, "An approach for qos based fault reconfiguration in service oriented architecture," *Procedia Computer Science*, vol. 46, pp. 766–773, 2015.
- [14] Y. Wang, Y. Chen, X. Tong, Y. Lee, and J. Yang, "Robot as a service in information science & electronic engineering education," in *Autonomous Decentralized System (ISADS), 2017 IEEE 13th International Symposium on*. IEEE, 2017, pp. 223–228.
- [15] M. K. Haki and M. W. Forte, "Service oriented enterprise architecture framework," in *Services (SERVICES-1), 2010 6th World Congress on*. IEEE, 2010, pp. 391–398.
- [16] I. M. Delamer and J. L. M. Lastra, "Service-oriented architecture for distributed publish/subscribe middleware in electronics production," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 4, pp. 281–294, 2006.
- [17] X. Qiao, X. Li, and J. Chen, "Telecommunications service domain ontology: semantic interoperation foundation of intelligent integrated services," *Telecommunications Networks-Current Status and Future Trends*, pp. 183–210, 2012.
- [18] J. Zhang, D. Kuc, and S. Lu, "Confucius: a scientific collaboration system using collaborative scientific workflows," in *Web Services (ICWS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 575–583.
- [19] C. Wang, X. Li, Y. Chen, Y. Zhang, O. Diessel, and X. Zhou, "Service-oriented architecture on fpga-based mpso," *IEEE Transactions on Parallel and Distributed Systems*, 2017.
- [20] N. S. RAJINI, "Development of healthcare environment using service oriented architecture," 2013.
- [21] S. Clement, D. McKee, and J. Xu, "Service-oriented reference architecture for smart cities," in *Service-Oriented System Engineering (SOSE), 2017 IEEE Symposium on*. IEEE, 2017, pp. 81–85.
- [22] J. S. Metcalfe and I. Miles, *Innovation systems in the service economy: measurement and case study analysis*. Springer Science & Business Media, 2012, vol. 18.

- [23] D. McKee, S. Clement, X. Ouyang, J. Xu, R. Romanoy, and J. Davies, “The internet of simulation, a specialisation of the internet of things with simulation and workflow as a service (sim/wfaas),” in *Service-Oriented System Engineering (SOSE), 2017 IEEE Symposium on*. IEEE, 2017, pp. 47–56.
- [24] A. Karande, M. Karande, and B. Meshram, “Choreography and orchestration using business process execution language for soa with web services,” *International Journal of Computer Science Issues IJCSI*, vol. 11, pp. 224–232, 2011.
- [25] <http://searchmicroservices.techtarget.com/opinion/The-future-of-financial-services-may-be-banking-on-SOA> (accessed on 04/05/2017).
- [26] D. Moitra and J. Ganesh, “Web services and flexible business processes: towards the adaptive enterprise,” *Information & Management*, vol. 42, no. 7, pp. 921–933, 2005.
- [27] K. M. Eisenhardt and J. A. Martin, “Dynamic capabilities: what are they?” *Strategic management journal*, pp. 1105–1121, 2000.
- [28] J. B. Harreld, C. A. O’Reilly, and M. L. Tushman, “Dynamic capabilities at ibm: Driving strategy into action,” *California Management Review*, vol. 49, no. 4, pp. 21–43, 2007.
- [29] P. Markey and G. Clynych, “A performance analysis of ws-*(soap) and restful web services for implementing service and resource orientated architectures,” 2013.
- [30] A. S. Johal and B. Singh, “Performance analysis of web services for android based devices,” *International Journal of Computer Applications*, vol. 92, no. 11, 2014.
- [31] S. Kumari and S. K. Rath, “Performance comparison of soap and rest based web services for enterprise application integration,” in *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*. IEEE, 2015, pp. 1656–1660.
- [32] D. Ruan, B. Zhang, and Y. Hua, “The realization of shopping sites based on soa architecture,” in *Computer Application and System Modeling (ICASM), 2010 International Conference on*, vol. 11. IEEE, 2010, pp. V11–81.
- [33] C. Gomathy and D. S. Rajalakshmi, “A software design pattern for bank service oriented architecture,” *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, vol. 3, pp. 1302–1306.

- [34] M. Brandner, M. Craes, F. Oellermann, and O. Zimmermann, “Web services-oriented architecture in production in the finance industry,” *Informatik-Spektrum*, vol. 27, no. 2, pp. 136–145, 2004.
- [35] A. W. Brown, M. Delbaere, P. Eeles, S. Johnston, and R. Weaver, “Realizing service-oriented solutions with the ibm rational software development platform,” *IBM systems journal*, vol. 44, no. 4, pp. 727–752, 2005.
- [36] J.W.Ross, “Creating a strategic it architecture competency: Learning in stages,” 2003.
- [37] S. Jones, *Enterprise SOA adoption strategies*. Lulu. com, 2006.
- [38] W. Vos and M. Matthee, “Towards a service-oriented architecture: A framework for the design of financial trading applications in the south african investment banking environment,” *South African Journal of Industrial Engineering*, vol. 22, no. 1, pp. 109–122, 2011.
- [39] P. Donachy, R. H. Perrott, T. J. Harmer, and F. Sharkey, “Finance sector: Requirements for high assurance within spatial soa based grid infrastructures,” in *High Assurance Systems Engineering Symposium, 2007. HASE’07. 10th IEEE*. IEEE, 2007, pp. 387–388.
- [40] F. M. Pinguelo and B. W. Muller, “Virtual crimes, real damages part ii: What businesses can do today to protect themselves from cybercrime, and what public-private partnerships are attempting to achieve for the nation of tomorrow,” 2012.
- [41] A. Masood, “Cyber security for service oriented architectures in a web 2.0 world: an overview of soa vulnerabilities in financial services,” in *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.
- [42] http://www.service_architecture.com/articles/web_services/business_process_execution_language_for_web_services_bpel4ws.html (accessed on 28/04/2017).
- [43] <http://www.oracle.com/us/products/middleware/soa/inside-look-soa-suite-wp-2217219.pdf> (accessed on 24/05/2017).

LIST OF PUBLICATIONS

- [1] P. Kaur, A. K. Verma, R. Kumar, “The Service Oriented Architecture in Financial Sector”, Journal of Computer Science and Engineering, vol. 2, no. 2, pp. 1-16, 2017.

LINK TO VIDEO

<https://www.youtube.com/watch?v=Tx13wF0SbPk>