Analyzing Service Oriented Architecture (SOA) in Open Source Products

MASTER THESIS IN SOFTWARE ENGINEERING
30 CREDITS, ADVANCE LEVEL

Carried out at: Vattenfall Business Services Nordic AB
Advisor at MDH: Sasikumar Punnekkat, sasikumar.punnekkat@mdh.se
Advisors at Vattenfall:
Alaa Karam, alaa.karam@vattenfall.com,
Arash Rassoulpour, arash.rassoulpour@vattenfall.com
Examiner: Sasikumar Punnekkat
Date: 07th October 2010

Author: Adnan Gohar
{(This page is intentionally left blank)}
ABSTRACT

Service Oriented Architecture (SOA) is an architectural paradigm that allows building of infrastructures for diverse application interaction and integration via services across different platforms, domains of technology and locations. SOA differs from traditional architectures, as it focuses on integrating capabilities that are distributed and implemented using a mixture of technologies. SOA provides a set of methodologies and strategies to accomplish interoperability and integration among different technology stacks.

Vattenfall is the fifth the largest energy supplier within Europe. Having operational systems in different countries brings the challenge of integrating all these distributed systems and this integration is a vital requirement for Vattenfall. The company is currently using Microsoft proprietary products to achieve integration across different technological platform, but requires a better integration infrastructure which is easily extensible and cost effective.

This thesis investigates the impact of implementing Service Oriented Architecture (SOA) using open source or proprietary software products within Vattenfall, from technological and financial perspectives. For this purpose, different technical and non-technical function blocks are identified which are essential for the implementation of SOA. These function blocks are mapped with SOA solutions provided by Red Hat’s JBoss Open Source SOA Platform and Microsoft’s SOA Platform. After mapping, a vendor specific technical and non-technical comparative analysis is carried out based on the function blocks, highlighting the strengths and weaknesses of each vendor.

Finally, an evaluation scheme is purposed based on the technical comparative analysis of vendors, SOA solution cost and SOA competence required. The results from this evaluation scheme are used to recommend the best solution vendor for Vattenfall Nordic. Moreover, this evaluation scheme can also be used to facilitate management in arriving at an appropriate decision about implementation of SOA, while remaining within their requirements and constraints.

Key words: Service Oriented Architecture (SOA), Enterprise Application Integration (EAI), Integration, Open Source, Enterprise Service Bus, Services, Red Hat JBoss, Microsoft BizTalk Server, Microsoft Windows Server AppFabric, Windows Communication Foundation (WCF)
{(This page is intentionally left blank)}
PREFACE

This document is a Master Thesis in the field of Software Engineering. The work described here has been accomplished at Vattenfall Nordic Business Services in Stockholm and supported by Mälardalen University in Västerås in a time span between April 2010 and October 2010.

I would like to thank All Might Allah for giving me the strength and opportunity to pursue this thesis. I would like to thank my family, who are always there for me in every situation. I cannot ask for a better family and today what I am is all because of their support and prayers.

A special thanks goes out to Arash Rassoulpour and Alaa Karam my supervisors at Vattenfall Nordic who have dedicated their time in supervising and mentoring me in the right direction at all times. Sasikumar Punnekkat my advisor at MDH also deserves special thanks for providing me valuable suggestions on how to proceed and for helping me to identify the finer aspects of report writing. Lastly, a thanks to EURECA Project (www.mrtc.mdh.se/eureka) funded by the Erasmus Mundus External Cooperation Window (EMECW) of the European Commission, for providing me with the opportunity to study at Mälardalen University.

A special thanks to all my friends especially to Sajid Ali my thesis partner from KTH for working with me on this thesis and others who motivated me when I needed it the most.

Adnan Gohar

agr09002@student.mdh.se
## CONTENTS

### Chapter 1  INTRODUCTION

1.1 Background ...................................................................................................................... 1
1.2 Objectives ......................................................................................................................... 1
1.3 Problem Formulation ....................................................................................................... 2
1.4 Limitations ....................................................................................................................... 3

### Chapter 2  INTRODUCTION TO SERVICE ORIENTED ARCHITECTURE (SOA)

2.1 History of SOA .................................................................................................................. 4
   2.1.1 Object Orientation ........................................................................................................ 4
   2.1.2 Enterprise Application Integration (EAI) ........................................................................ 5
   2.1.3 Web Services ................................................................................................................ 5
   2.1.4 Business Process Management ..................................................................................... 6
2.2 SOA Entities ....................................................................................................................... 6
   2.2.1 Service Consumer ......................................................................................................... 6
   2.2.2 Service Provider ............................................................................................................ 7
   2.2.3 Service Registry ........................................................................................................... 7
   2.2.4 Service Contract .......................................................................................................... 7
   2.2.5 Service Proxy ................................................................................................................ 7
   2.2.6 Service Lease ................................................................................................................ 7
2.3 SOA Characteristics .......................................................................................................... 7
   2.3.1 Services are discoverable and dynamically bound ........................................................ 7
   2.3.2 Services are self contained and modular ..................................................................... 7
   2.3.3 Services are interoperable ............................................................................................ 8
   2.3.4 Services are loosely coupled ....................................................................................... 8
   2.3.5 Services have a network-addressable interface ............................................................ 8
   2.3.6 Services have coarse-grained interfaces ...................................................................... 8
   2.3.7 Services are location transparent ............................................................................... 9
   2.3.8 Services can be composed into new applications ........................................................ 9
   2.3.9 SOA supports self-healing ......................................................................................... 9
2.4 Why SOA? .......................................................................................................................... 9
   2.4.1 Improvement in Business Decisions ........................................................................... 9
   2.4.2 Improved employee productivity ............................................................................... 9
   2.4.3 Easy integration of customer and supplier systems ....................................................... 9
   2.4.4 More productive, more flexible applications ............................................................... 10
   2.4.5 Faster, more cost-effective application development .................................................. 10
   2.4.6 More manageable and secure applications ................................................................. 10
2.5 SOA from a Technical Perspective .................................................................................. 10
   2.5.1 Services ....................................................................................................................... 10
   2.5.2 Enterprise Service Bus ............................................................................................... 12
   2.5.3 Portals .......................................................................................................................... 13
   2.5.4 Registry ....................................................................................................................... 14

### Chapter 3  SERVICE ORIENTED ARCHITECTURE AT VATTENFALL

3.1 Definition of SOA at Vattenfall Nordic .......................................................................... 16
   3.1.1 Objectives of “Vattenfall Strategy For Service Orientation” ...................................... 16
3.2 Vattenfall Nordic Requirements and SOA ...................................................................... 16
   3.2.1 Technical Reference Architecture ............................................................................... 16
3.3 Modules of Technical Reference Architecture ................................................................ 17
3.3.1 Service Creation and Abstraction
3.3.2 Run-time Service Management
3.3.3 Service Orchestration
3.3.4 Process Control and Optimization
3.3.5 Presentation
3.3.6 Identity Management
3.3.7 Design-time Service Management

3.4 Specific Requirements

Chapter 4  SOA SOLUTION BY VARIOUS VENDORS  20

4.1 RedHat and SOA
  4.1.1 JBoss Enterprise SOA Platform
  4.1.2 JBoss Enterprise BRMS
  4.1.3 JBoss Enterprise Application Platform
  4.1.4 JBoss Enterprise Portal Platform
  4.1.5 JBoss Operation Networks Platform
  4.1.6 JBoss Developer Studio
  4.1.7 JBoss SOA Governance

4.2 Microsoft and SOA
  4.2.1 Microsoft BizTalk Server
  4.2.2 Microsoft Windows Server AppFabric

Chapter 5  ANALYSIS of SOA Solutions  38

5.1 Weighting Criteria
  5.1.1 Parameter Weightings According to VattenFall Nordic Importance

5.2 SOA Capabilities Provided by Vendors
  5.2.1 Fulfillment of Requirements By Vendors
  5.2.2 Vendor Weighting on Basis of Capabilities Provided

5.3 Vendors SOA Solutions Cost Comparison
  5.3.1 Implementation cost for building a new SOA system
  5.3.2 Implementation cost for building SOA using current system(s)

5.4 SOA Competence in the Organization

5.5 Summary of Recommendations
  5.5.1 Technological Capabilities
  5.5.2 Cost Effectiveness
  5.5.3 SOA Competence

Chapter 6  ROADMAP AND CONCLUSION  57

6.1 Vendor Products Roadmap
6.2 Conclusion

Chapter 7  REFERENCES  59
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comparison of SOA among different technologies and vendors</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>SOA components and their relationship</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Enterprise Application Integration</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Service Oriented Architecture Conceptual Model</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Service Granularity</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>Service usage in a SOA context</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Typical architecture of an Enterprise Service Bus</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>Portals in SOA</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>Working of Registry</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>Vattenfall Nordic Technical Reference Architecture</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>RedHat JBoss Enterprise Middleware</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>RedHat JBoss Enterprise SOA Platform</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>Microsoft Application Platform for SOA</td>
<td>29</td>
</tr>
<tr>
<td>14</td>
<td>Microsoft BizTalk Server and SOA</td>
<td>29</td>
</tr>
<tr>
<td>15</td>
<td>BizTalk ESB ToolKit integration and components</td>
<td>31</td>
</tr>
<tr>
<td>16</td>
<td>Windows Server AppFabric Overview</td>
<td>36</td>
</tr>
<tr>
<td>17</td>
<td>Windows Server AppFabric Architecture</td>
<td>37</td>
</tr>
<tr>
<td>18</td>
<td>Importance Level Graph</td>
<td>39</td>
</tr>
<tr>
<td>19</td>
<td>Red Hat JBoss Capabilities and Function Blocks</td>
<td>46</td>
</tr>
<tr>
<td>20</td>
<td>Microsoft BizTalk Server Capabilities and Function Blocks</td>
<td>46</td>
</tr>
<tr>
<td>21</td>
<td>Microsoft Windows Server AppFabric Capabilities and Function Blocks</td>
<td>47</td>
</tr>
<tr>
<td>22</td>
<td>Capabilities Comparison of JBoss, BizTalk Server and Windows Server AppFabric</td>
<td>47</td>
</tr>
<tr>
<td>23</td>
<td>Cost Comparison for Red Hat JBoss and Microsoft BizTalk Server for new SOA system</td>
<td>49</td>
</tr>
<tr>
<td>24</td>
<td>Cost Comparison for Red Hat JBoss and Microsoft Windows Server AppFabric for new SOA system</td>
<td>50</td>
</tr>
<tr>
<td>25</td>
<td>VattenFall Nordic’s Existing System Cost</td>
<td>51</td>
</tr>
<tr>
<td>26</td>
<td>VattenFall Nordic Existing System Cost and Red Hat JBoss</td>
<td>52</td>
</tr>
<tr>
<td>27</td>
<td>SOA Implementation cost using VattenFall Existing System and Red Hat JBoss</td>
<td>52</td>
</tr>
<tr>
<td>28</td>
<td>SOA Competence of employees within Vattenfall Nordic</td>
<td>53</td>
</tr>
<tr>
<td>29</td>
<td>Percentage of Function Blocks Satisfied by Vendor Products</td>
<td>54</td>
</tr>
</tbody>
</table>
Figure 30: SOA Implementation Cost by different Vendor Products ................................................................. 55

Figure 31: SOA Competence in Vattenfall Nordic ...................................................................................................................... 56
**LIST OF TABLES:**

Table 1: Function Block Importance Weightings ............................................................... 39
Table 2: Function Blocks and Vendor Product Capabilities .................................................. 43
Table 3: Vendor evaluation of basis of capabilities provided .................................................. 45
Table 4: H/W & S/W specification for price calculation .......................................................... 48
Table 5: Implementation cost for new SOA system ............................................................... 49
Table 6: VattenFall Nordic Current Integration Platform Cost ............................................... 51
Table 7: Comparison of SOA competence within Vattenfall Nordic ...................................... 53
Table 8: Roadmap for Vendors Products .............................................................................. 57
Chapter 1
INTRODUCTION

1.1 BACKGROUND

Information Technology (IT) has revolutionized the way we work, live or even think today. Looking back ten years from now, we could not have imagined about the tasks we can perform now by just sitting in front of a computer system. Tom Forester describes this elegantly, "If the automobile and airplane business had developed like the computer business, a Rolls Royce would cost $2.75 and would run for 3 million miles on one gallon of gas. And a Boeing 767 would cost just $500 and would circle the globe in 20 minutes on five gallons of gas." [1].

Every industry and enterprise is opting for better IT solutions. Quick and accurate information from different systems combined at one platform could prove to be vital for the survival of an enterprise. Easy accessibility of information by customers and information flow within businesses is a must for every enterprise. Information is gathered from disparate systems within an enterprise, which results in the enterprise not being confined to one technology based system.

Vattenfall is Europe's fifth largest generator of electricity and the largest generator of heat. The company operations span in different countries across Europe and in United Kingdom. Having, operations across different countries bring the need of integration and interoperability of different heterogeneous environments functioning in different countries. These heterogeneous environments comprise of legacy systems and systems developed using a variety of software technologies.

As the company is expanding its operations in more and more countries, it requires an architectural paradigm which is cost effective, adaptive to change and can easily integrate new systems irrespective of the underlying technology. Different solutions are available for resolving interoperability problems. Service Oriented Architecture (SOA) is one such solution, which pretty well caters such needs of enterprises.

Service Oriented Architecture (SOA) has become a well established concept in the field of Information Technology. SOA if applied correctly benefits enterprises immensely, in both IT and business integration sector. Lowered costs for integration and easy adaptability to change are the biggest benefits of SOA. Therefore, it is a concept of future and is being used for designing and architecting more and more IT solutions. However, if not applied correctly or according to business need, it can easily lead to a negative outcome for the enterprise business and IT infrastructure.

1.2 OBJECTIVES

The main objective of this thesis is to analyze service oriented architecture among different SOA solution vendors. The main emphasis would be on how and to what extent SOA can be achieved with open source SOA solutions and what kind of competence is required for it. A comparative analysis would be done among different SOA solution vendors (both proprietary and open source). The result of this analysis would help Vattenfall Nordic to make a decision regarding how to implement Service Oriented Architecture in the most cost effective way. This analysis would also help then in deciding either to opt for SOA solutions provided by open source solutions or upgrade their current proprietary solutions to support SOA.
1.3 PROBLEM FORMULATION

Vattenfall Nordic is currently using Enterprise Application Integration (EAI) [60] architectural technique to connect and integrate all its enterprise systems. The middleware integration platform used for EAI is Microsoft’s BizTalk Server and for front-end and business systems SAP is being used. BizTalk Server is responsible for integrating all these systems. The current architectural approach works fine for integrating the currently running systems, but as the company is expanding and acquiring new companies this approach does not fulfill the purpose of dynamic integration and cost effectiveness. Therefore, Vattenfall Nordic is exploring new integration techniques and platforms which are easy to integrate and cost effective. The aim of this thesis is to investigate about Service Oriented Architecture based integration platform, from a conceptual and practical perspective.

Service Oriented Architecture would be analyzed from a practical/implementation perspective with respect to vendors. SOA solutions from different vendors would be considered. The main vendors which would be considered and provide enterprise level SOA solutions are:

- RedHat JBoss [61]
- Microsoft [62]

For comparison of these SOA solution vendors, an evaluation scheme is devised based on the core principles of Service Oriented Architecture and general requirements and needs of any enterprise. The vendors would be evaluated on basis of fulfillment of the capabilities listed in the evaluation scheme and cost of SOA solution from each vendor.

From a technical perspective the main focus would be on the following core elements of SOA

1. Enterprise Service Bus (ESB)
2. Portals
3. Registry
4. Services

![Figure 1: Comparison of SOA among different technologies and vendors](image-url)
1.4 LIMITATIONS

SOA is a huge paradigm and it would not be possible to cover all the design principles and patterns associated with SOA. This thesis would look at specific functional areas of Service Oriented Architecture which are important for Vattenfall Nordic from the technical perspective only and the Business perspective (i.e. consultant costs, developer costs, etc) would not be covered and are beyond the scope of this thesis. Moreover, a comparison between three vendors had to be done as shown in Figure 1, but due to time constraint only two vendors are being considered. Finally, the comparison would be only among current available vendor SOA suites and for the roadmap these suite would only be discussed and not compared.
Chapter 2
INTRODUCTION TO SERVICE ORIENTED ARCHITECTURE (SOA)

SOA represents an open, agile, extensible, federated, composable architecture comprised of autonomous, QoS-capable, vendor diverse, interoperable, discoverable, and potentially reusable services, implemented as Web services [2].

Service Oriented Architecture (SOA) can be thought of as a paradigm or an architectural style. It is not a framework or product that can be bought and used. It is more of a thinking approach which helps in designing interoperable software architectures according to enterprise business needs. SOA aims to combine architectural style and methodologies to achieve interoperability among heterogeneous and homogenous systems by creating reusable services. It concentrates on the problem domain more rather than the underlying technology used to implement it.

Service Oriented Architecture (SOA) provides a design framework to integrate disparate applications so that their functionality can be exposed and made available as services on a network. Moreover, SOA divides monolithic applications into different service sets, making functionality modular to implement. There are different implementation models and techniques for SOA but the most common is through standardized Web Services which make interoperability easy to implement among disparate systems.

2.1 HISTORY OF SOA

Alexander Pasik a former analyst at Gartner coined the term SOA in 1994 while teaching middleware architecture. Pasik came up with SOA term because the traditional client/server architecture had lost its touch and people started referring to distributed computing as involving a desktop PC which ran the frontend logic and another computer running and maintaining backend operations. To avoid confusion among meanings of client/server and SOA which originated as misconceptions, Pasik stressed on “server orientation”, as he encouraged developers to design SOA based business applications [9].

To best understand SOA we need to have a little knowledge about its history. Service-orientation, concept did not get created in a day, it represents the evolution of different IT paradigms and technologies. Moreover, SOA itself is in the phase of evolution and this makes it difficult to precisely define SOA. Following are the technologies from which SOA has evolved [11, 10]:

2.1.1 OBJECT ORIENTATION

Object orientation emerged in the 90’s giving a new concept of defining how distributed solutions would be built. Object orientation had its own rules and principles, which helped ensure consistency among applications running on different environments. The main principle which came out of object orientation was of objects being
the units of solution logic, and the entire solutions would communicate and relate with each other using these objects.

Service-orientation can be considered an upgraded version of object orientation, and the principles and patterns of object-oriented analysis and design have been inherited by service-orientation, for its communication and implementation. For instance, the principles Service Reusability, Service Abstraction, and Service Composability are basically object-orientation concepts which are being used in service oriented paradigm [11].

### 2.1.2 ENTERPRISE APPLICATION INTEGRATION (EAI)

In the late 90’s when enterprises started to communicate with each other more frequently, integration became a focal point in IT. Different integration techniques were developed, and the first one was point-to-point integration channel. This technique worked for small integrations, but as soon as the enterprise grew, it became too complex to manage. The well known problems with this technique were lack of scalability, extensibility and no support for interoperability.

Enterprise Application Integration (EAI) platforms were introduced to cater the problems associated with point-to-point integration technique. This paradigm used adapter, broker component and orchestration engines to achieve application integration and business processing. EAI had its own set of problems. The biggest problems with EAI were high scalability cost and vendor locking. Due to these problems, enterprises started looking at other solutions which were vendor diverse and had low scalability cost. With the emergence of Web Services as an open framework, enterprises moved towards service orientation which gave them much more flexibility and control over their integration architecture and platform.

There are number of innovations which were developed in the era of EAI and are being used within SOA. One such innovation is the broker component, which allows services with different schemas to interoperate with each other through runtime transformation. Orchestration engine is another such inherited component. These inherited components of EAI map to several service oriented principles, like Service Abstraction, Service Statelessness, Service Loose Coupling and Service Composability [11].

![Figure 3: Enterprise Application Integration](image)

### 2.1.3 WEB SERVICES

Web services have become the main implementation technique for SOA. This is because Web Services are standardized across vendors, which allows easy interoperability among disparate systems. Most of the vendors are now using Web services as their main technology for the implementation of SOA solutions.
As SOA is a concept, it has been influenced by different vendors and frameworks, one such framework is the Web services framework, which has also influenced SOA in shaping up several of its principles including Service Abstraction, Service Loose Coupling, and Service Composability [13, 11].

2.1.4 BUSINESS PROCESS MANAGEMENT

Business processes to be running smoothly and in a correct order is essential for an enterprise. Business process management emphasis is on streamlining business processes, by making them adaptive to change and improving their efficiency. Business process layer is the main layer within service oriented architecture. It provides the capability of service composition and orchestration.

One of the main goals of Service oriented architecture is to automate business processes, which are highly adaptive to change. This goal can be achieved by abstracting business logic into one layer, and having services to reuse this logic without having to recreate it again. Service logic reusability is a key factor within Service Oriented Architecture and business process management fulfills that functionality.

2.2 SOA ENTITIES

SOA contains many entities within its architecture. The core of SOA is based on three entities (Service Consumer, Service Provider and Service Registry) and the interaction among them. The interaction is as follows: “service consumer” requests the “service provider” for a service, the provider searches the registry for the requested service which matches the consumers needs, if it is found the service information is sent to the consumer. Figure 4 shows a meta-model of the interaction.

![Figure 4: Service Oriented Architecture Conceptual Model](image)

The six main entities of a SOA conceptual model are as follows [4]:

2.2.1 SERVICE CONSUMER

Service consumer is the entity which consumes the functionality provided by a service. It calls different functions within a service for different functionality. The consumer can be a software module, a service or an application. The consumer can either directly call the service if its location is known or can lookup the service location in the registry.
2.2.2 SERVICE PROVIDER

Service provider is the main service; it accepts and executes requests from the consumer. It contains the service description and implementation details. The service provider can be a mainframe system, a component or any other type of system which executes the consumer’s requests.

2.2.3 SERVICE REGISTRY

Service registry is a network directory which contains addresses of all the available services. Its main purpose is to store and publish service contracts of different service providers, and when requested provide these contracts to interested service consumers.

2.2.4 SERVICE CONTRACT

A service contract is a specification or description of the way a service consumer would be interacting with a service provider. It possesses information about the exchanged message formats, quality of service and the pre and post conditions which need to be fulfilled before the service can be executed.

2.2.5 SERVICE PROXY

Service proxies are provided by service providers to facilitate service consumers by allowing them to call functions of the service using an API which is exposed via proxy. This API is written in the local language of the consumer. The service proxy can enhance performance by providing caching of remote references and data. This proxy is optional in SOA and services can also be called directly by the consumers.

2.2.6 SERVICE LEASE

The service lease is the amount of time for the validity of a service contract. It is granted and managed by the registry. The lease is important for the services which require maintenance of state information and about the bindings between the consumer and provider. In addition, using service lease increases loose coupling between service provider and consumer.

2.3 SOA CHARACTERISTICS

Software architectures have some specific characteristics and principles that need to be followed in order to fully utilize their capabilities [4, 3]. Service oriented architecture has the following characteristics:

2.3.1 SERVICES ARE DISCOVERABLE AND DYNAMICALLY BOUND

SOA provides a way to discover services dynamically at run-time. The service consumer asks the registry for the required service and in response receives the essential information to execute the service at runtime. Service interfaces are discovered dynamically and messages are also constructed at runtime which leads to no compile time dependency of being bound to the service. The only thing that needs to be updated is the service contracts, if they change.

2.3.2 SERVICES ARE SELF CONTAINED AND MODULAR

Services are usually self contained and modular. A service has a set of functional interfaces which collectively performs business oriented tasks. These interfaces relate to each other to form a module and have enough information to function without being authenticated or being dependent on other software applications or modules.
2.3.3 SERVICES ARE INTEROPERABLE

One of the main features of service oriented architecture is interoperability, which is the ability to communicate with different systems without being dependent on any specific platform or language. Every software module or architecture has some proprietary structure which is restricted to its own domain and is considered tightly coupled. Service based architecture achieves interoperability by supporting protocols and data formats of the service current and potential consumers.

2.3.4 SERVICES ARE LOOSELY COUPLED

Coupling refers to the level of dependencies among software modules. Coupling can be categorized as either loosely coupled or tightly coupled. Loosely coupled modules have well-defined dependencies and are more flexible. Most of the software architectures try to provide modules which are as much loosely coupled as possible. On the other hand, tightly coupled software systems are difficult to organize as they have lots of unknown requirements for each module within the software structure or architecture.

Service oriented architecture emphasis on development of loosely coupled services, meaning there should be minimal dependencies between the consumers and providers. In SOA, loose coupling is accomplished by the use of registry, which contains the binding and contract information of every service. Contracts are the main concern for the service consumer and they need no other specific information to call the service. However, tight coupling remains at the implementation level i.e. at interface definition or binding to a specific protocol.

2.3.5 SERVICES HAVE A NETWORK-ADDRESSABLE INTERFACE

A service should have a network address where its interface is published. This is one of the main design principles of service oriented architecture. The consumer should be able to invoke a service across a network in a distributed manner; this makes a service reusable and available to different consumers. Services can also be accessed through a local interface without any involvement of network; this is possible when both provider and consumer are on the same machine. This is mainly done to improve performance.

2.3.6 SERVICES HAVE COARSE-GRAINED INTERFACES

The concept of granularity with respect to services can be defined in two ways. Firstly, scope of the implementation domain of the service. Secondly, scope of interface implementation for each method within the service. In short, it can be seen as the implementation of interfaces within the system. There are levels of interface granularity, if the interface provides all the business functionality, it is considered as a coarse-grained interface. On the other hand, if the interface implements part of the business functionality, it is called fine-grained interface. A service can have granularity for all its interface methods or for some particular interface methods. A service oriented system by default supports coarse-grained interfaces with different granularity levels for the service interface. Meaning, objects within a service can be fine-grained, but these objects remain within the physical structure of the service as shown in Figure 5.

Figure 5: Service Granularity
2.3.7 SERVICES ARE LOCATION TRANSPARENT

Location transparency is one of the main characteristic which SOA promotes. Location transparency enables services to move from one location to another without affecting the consumer of the service. Consumer has no knowledge of the location of the service until it looks for it at run-time in the registry. The consumer has dependency only on the service contract; this makes it easier to change service implementation location.

2.3.8 SERVICES CAN BE COMPOSED INTO NEW APPLICATIONS

Reusability of services is also one of the key characteristics of SOA. This enables development of new applications using existing services. Composition of services is an effective design approach which concentrates on reusability of service functionality without having any knowledge of its future use.

2.3.9 SOA SUPPORTS SELF-HEALING

A self-healing system is one which has the ability to recover to normal state in case an error occurs. Applications within a SOA based system should possess the ability of self healing, as within a service based system, services are usually combined to execute business functionality. For a SOA based system to execute efficiently and properly it should always support self-healing properties.

2.4 WHY SOA?

SOA helps an enterprise to benefit both in its business and IT area. From the business perspective, SOA makes it possible to develop dynamic applications, which resolve a number of business problems that are essential for the growth of an enterprise [7].

Some of the benefits of SOA in accordance with business development are:

2.4.1 IMPROVEMENT IN BUSINESS DECISIONS

SOA helps an enterprise to combine different business centric information into one composite business application. This dramatically increases information accuracy and provides decision makers access to a lot of information. Different departmental information can be mapped onto a single web page, enabling enterprises and CEO’s to have a better understanding of their business costs and tradeoffs of decisions being made during daily business operations. Information being readily available also makes it possible for an enterprise to identify and rectify problems quickly.

2.4.2 IMPROVED EMPLOYEE PRODUCTIVITY

SOA enables employees to be more productive by providing them easy and efficient access to systems and information, which in turn improves business processes. SOA removes the IT platform integration restrictions and limitations, enabling employees to focus more on the important business processes. Moreover, clients can access their required information in any view which could be web, desktop or mobile application; this really enhances productivity.

2.4.3 EASY INTEGRATION OF CUSTOMER AND SUPPLIER SYSTEMS

SOA’s main benefit is easy integration of different systems and applications. Enterprises can easily merge with each other without any major change in their current running IT systems. Integration with other partner enterprises and streamlining of different processes within the enterprises is easily done. Customer/partner stratification increases as they get access to dynamic applications and business services, all at one place or portal.
SOA greatly helps in defining supply chain processes for specific business tasks which are independent of their underlying IT architecture, this leads to better alignment of business processes within enterprise.

SOA also contributed in documenting different business processes and models, this documentation helps in optimizing and capturing change within these business processes.

From an IT department’s perspective, service-oriented architecture provides a simplified framework for the creation and management of integrated systems and applications. It is a technique to map IT processes with business models and their changing needs.

### 2.4.4 MORE PRODUCTIVE, MORE FLEXIBLE APPLICATIONS

SOA enables exiting IT systems including legacy systems and applications to integrate without any need for adding any custom code or updating these systems to newer versions. This increases the productivity and profitability of the enterprise business. SOA also enables enterprises to develop dynamic and composite applications which provide information from different systems irrespective of their underlying environment/platform and programming language. Furthermore, as services are not dependent on their underlying platform, solutions can be designed with greater flexibility.

### 2.4.5 FASTER, MORE COST-EFFECTIVE APPLICATION DEVELOPMENT

Services are standardized which allows developers to make them as reusable services that can be orchestrated into composite services and applications to fulfill business change requests. This reduces the cost of solution, development and testing time and speeds up the process of implementing new requirements within the business system. Moreover, the use of one standardized development model increases efficiency and simplifies application development, maintenance and testing.

### 2.4.6 MORE MANAGEABLE AND SECURE APPLICATIONS

SOA solutions provide a standardized infrastructure for developing applications which are secure, can be monitored and have services with standardized contracts. SOA makes it very easy to add and map service capabilities to new and existing business processes. Services can be added independently within a service-oriented solution, which helps in removing the need of developing a new application. Existing IT applications just need to include new services to start providing the new functionality. This saves cost for the enterprise, of developing new applications. Moreover, SOA provides a strong authentication and authorization model for all services, and as services are loosely coupled, the overall security is increased.

### 2.5 SOA FROM A TECHNICAL PERSPECTIVE

Apart from the entities and characteristics discussed above, from a more technical and implementation perspective SOA requires the following elements to fully follow the above mentioned characteristics:

#### 2.5.1 SERVICES

“A service is a software component that is described by meta-data, which can be understood by a program” [8]

There are different ways of defining services, as different services consist of different attributes. In a more business context, services can differ a lot depending on the different business entities they provide. Services can be used to read data, write data or as a composition of services perform some workflow/process. Therefore, all services cannot be looked at and treated in the same manner, it is better to categorize them according to their attributes and functionalities.
Technically services can be categorized in the following three categories [9]:

1. Basic Services
2. Composed Services
3. Process Services

**Basic Services**

Basic Service is the simplest service and it only provides basic business functionality. In other words, they only provide a service for reading or writing to one specific backend. Their lifespan is short and are usually stateless services. Basic Services are further divided into two types:

i) **Basic data services:**

   This service is associated with a specific backend and they only perform reading and writing operations to that backend. Moreover, as these services perform reading and writing operations they must incorporate ACID (Atomic, Consistency, Isolation, Durability) [63] properties.

ii) **Basic logic services:**

   These are similar to basic data services; the difference is that these services incorporate only basic business rules and logic.

**Composed Services**

Composed services are composition of either basic services or other composed services. Composed services in the context of SOA are also referred as orchestrations. These can be considered at a higher level then basic service as their lifetime is also short and usually are stateless. These run usually within a business process.

**Process Services**

Process services usually represent workflows or business process. These are activities/services which run for a long term within one workflow or business process. As these are long term running services they need to maintain a state (e.g. shopping cart). These states might need to be maintained over multiple sessions. These states help in failover and recovery activities i.e. resume the process from the same state as it was before the error/failure was encountered.

**Working of Services in SOA:**

In SOA there are usually three roles which come into play: requester, provider, and broker.
- **Service Provider:**

  The service provider is the entity which has access to other services, and is also responsible for creating services and publishing them to the service broker.

- **Service Requester:**

  The service requester is the entity which basically requires some operational work performed by another service and works by searching through the list of service descriptions provided by the service broker. It is also responsible for binding the services after their discovery.

- **Service Broker:**

  Service broker contains all the information about services which are registered with it and is also responsible for redirecting requests to the corresponding service requester and provider.

### 2.5.2 ENTERPRISE SERVICE BUS

In computing, an enterprise service bus (ESB) consists of a software architecture construct which provides fundamental services for complex architectures via an event-driven and standards-based messaging-engine (the bus) [12].

The enterprise service bus pattern defines a model for integrating enterprise applications using a common messaging bus. ESB builds upon Enterprise Application Integration (EAI) making it more flexible, facilitating it with standard based integration and not limiting it to web service based integration (i.e. communication is not restricted to http protocol only). One of the neat features of ESB is that a consumer can send his message using http protocol and the provider can responds using a JMS, and they do not have to care about any transformation or communication protocols. The ESB would take care of all these infrastructural issues. In short, an ESB provides message based transportation similar to an EAI, in addition to this it also provides registry service which provides describe and discover facility for consumers. Different mechanisms are also provided for service level agreements (SLA’s) /contracts, security, routing of messages and transformation of messages. These capabilities do vary from vendor to vendor [14].

![Figure 7: Typical architecture of an Enterprise Service Bus](image-url)
The ESB pattern contains a single identity and access control mechanism, service naming mechanism, service registry, and common messaging bus which is responsible for communications among different protocols. ESB provides communication among services registered on the bus and applications consuming those services, without knowledge about their physical location. It also helps in standardizing identity management, naming conventions, message formats and communication protocols. The best feature which ESB provides is once a service gets registered on the bus, everything else registered on the bus can connect and communicate with that service, this leads to a very loosely coupled system [15].

ESB was introduced to overcome Enterprise Application Integration (EAI) issues. The main issues of point to point connection and single point failure were resolved by enterprise service bus.

Enterprise service bus can be considered the backbone of service oriented architecture. The features provided by an ESB are essential for the implementation of SOA. There are many vendors which provide open source and proprietary ESB for different technologies.

### CAPABILITIES SUPPORTED BY ESB

Following are some of the capabilities and functionalities which an enterprise service bus supports [16]:

- **Communication:** Communication and routing are two most important features of an enterprise service bus. It supports different routing and messaging styles e.g. request/response, publish/subscribe, and for communication it supports different transport protocols that can be made widely available. This helps in location transparency which in turn leads to loose coupling between the consumer and the service location.
- **Integration:** For integration ESB provides several adapters and integration techniques. These techniques help in creating services which hide technical details of services, exposing only relevant contract information for the consumer.
- **Service Interaction:** Interface formats and messaging models are very important for interoperable communication among applications and services. ESB supports different models and formats based on WSDL and SOAP, which standardize interfaces and allow services to communicate with only details specified within the contract.
- **Management:** ESB also provides administration facility to monitor and manage routing, addressing, transformation and control over naming capabilities. This helps in managing services within the enterprise.

### ADDITIONAL FEATURES IN ENTERPRISE SERVICE BUS

- Service orchestration engines can be added to an ESB to support long-running (stateful) and short-running (stateless) processes.
- Monitor and manage quality of service and service-level capabilities.
- Presentation services, to help create custom portals that combine services from multiple sources.

### 2.5.3 PORTALS

Portals are optional add-on for service oriented architecture. Their primary purpose is to formulate either administration console or application front-end integration into the SOA paradigm. In any software design interfaces are developed initially so that a picture can be made of the final product (this can be desktop based application or web based application). Similarly, in SOA portals can also act as the first logical step in the process of design [17].
The first step in SOA is to define required services and applications which would be consuming those services. Portals usually perform the same activity of identifying services which are required and the applications who would be consuming those services. So, portals can be considered as the first step toward SOA implementation as its primary nature lends itself to SOA approaches. Portals use service-oriented concepts in; leveraging web services and portlets, service consumers or applications which communicate to provide orchestrated flows and on-the-glass composite applications. In the concept of SOA, portals can play an important role in taking decision which technology should be used to implement the required functionalities and capabilities. To enter service oriented architecture market it is important for an enterprise to take a “ramp-up” approach. Portals become good candidates to start off with, even from the standpoint of making the SOA concept less abstract [17].

In short, portals provide user interfaces to interact with services; they depend on the service provider and service consumer for their design and importance.

![Portals in SOA](image)

As shown in Figure 8 portals reside at the top most layer of SOA. An ESB is accessible to all these layers.

### 2.5.4 REGISTRY

The implementation artifacts that drive from SOA should be registered within a repository to maximize reuse and provide management of enterprise assets [18].

Registries main purpose is to register services and equip them with the capability of being dynamically discovered. The service gets registered within the registry by providing its address and Meta information. The ultimate goal of registry is to allow diverse applications to communicate reliably, without any interoperability problem and with minimal human intervention.

In other words, a SOA registry can be seen as a resource that provides controlled access to data essential for governance of service-oriented architecture based systems. It’s a catalog which contains information about the available services. This catalog is constantly updated as information about services gets added, updated and deleted frequently. A SOA registry allows business services to get dynamically discovered to be used by the consumer.

SOA registry can be implemented using different approaches. The most common approach is the UDDI (Universal Description, Discovery and Integration) specification, which is a XML based registry that was
developed for the purpose of making systems interoperable for e-commerce and interaction among web services. SOA registry builds upon UDDI by adding feature of continuous revision of content. This sort of content can exist in the form of XML documents, workflows or process information and about any potential business partner. As SOA builds upon UDDI it facilitates other enterprise applications to discover and use current and relevant information in a more efficient manner then it can be done using UDDI [18, 19].

Figure 9: Working of Registry
Chapter 3
SERVICE ORIENTED ARCHITECTURE AT VATTENFALL

“Vattenfall Strategy for Service Orientation” is to define the concept, clarify a Group governance structure, identify key requirements for business and IT initiatives and projects to follow when applying service oriented concepts. This strategy will also be the foundation for conducting pilot projects in the area with the purpose of demonstrating business values in reality, and thus verifying the strategy. This would pave a way for IT to actively support and drive the business by providing a value proven concept [47].

3.1 DEFINITION OF SOA AT VATTENFALL NORDIC

Vattenfall Nordic’s definition of SOA is not much different from the conventional definition of SOA. They also refer to the same principles, guideline, technologies and standards. In addition, the most important requirement for Vattenfall Nordic is the integration of SOA with their current running system.

3.1.1 OBJECTIVES OF “VATTENFALL STRATEGY FOR SERVICE ORIENTATION”

The objective of this “Vattenfall Strategy for Service Orientation” is to define a clear concept of how SOA would be implemented within Vattenfall Nordic, it would cover all aspects from identifying a group governance structure to identifying key requirements for business and IT initiatives and projects to follow when applying service oriented concepts [47].

- The short-term target is to establish a common basis within the IT community for how Vattenfall Nordic will work with service orientation and SOA.

- The mid-term target is to have verified this strategy, thus having anchored it within the IT community, and demonstrated the benefits with SOA for some parts of the business via a couple of SOA pilots.

- The desired end state is that Vattenfall Nordic shall have one well understood and known concept for how to make value out of service orientation and SOA. All roles, responsibilities and governance structures are implemented. Key competences will being in place and available for business and IT.

3.2 VATTENFALL NORDIC REQUIREMENTS AND SOA

Service Oriented Architecture is a huge paradigm, this make it difficult to standardize according to certain technological standards. Being a concept, different SOA solution vendor provide different capabilities which map to the concept of Service Oriented Architecture. Apart from this, different enterprises have different requirements and it is difficult to pin point or list down a generic set of requirements for the implementation of SOA that fulfill requirements for all enterprises.

We have tried to identify a number of function blocks which are generic in nature and should be considered important for the implementation of SOA within an enterprise. These function blocks comprise of parameters from the Technical Reference Architecture (TRA) and specific needs of any enterprise. As Vattenfall Nordic has a huge infrastructure and integration platform spread across Europe, we believe that these function blocks would be more-or-less same for any other enterprise which is thinking of opting for a Service Oriented system.

3.2.1 TECHNICAL REFERENCE ARCHITECTURE

A set of generic building blocks which together delivers the capabilities necessary to create business functionality based on service orientation.

Vattenfall Nordic has devised a Technical Reference Architecture (TRA) which contains most of the functionality that maps to Service Oriented Architecture principles. For our evaluation scheme, we are going to
reuse the concepts within TRA and would be defining different capabilities within each function block of the TRA.

3.3 MODULES OF TECHNICAL REFERENCE ARCHITECTURE

The different modules within TRA are described below:

3.3.1 SERVICE CREATION AND ABSTRACTION

This module should enable applications to get registered in a SOA based solution environment. Encapsulate existing applications and data sources into standardized services, utilizing a standardized interface technology (e.g. web services or JMS). The capabilities which this module should provide are:

- SOA Connection
- Transformations
- Routing

3.3.2 RUN-TIME SERVICE MANAGEMENT

This module is responsible for executing discrete services in a controlled way according to defined policies. The capabilities this module should support are:

- Service throttling
- Runtime statistics and governance
- Dynamic service ramp-up
- Security policy fulfillment

3.3.3 SERVICE ORCHESTRATION

This module is responsible for orchestrating basic services to form composed services or business processes services. Composed services can be made on several levels; from short-term technical processes (e.g. managing asynchronous behavior or aggregation of data) to the execution of complete long-running business processes, including compensation mechanisms. The capabilities which this module should provide are:
• Process control
• Workflows
• Business rules execution

### 3.3.4 PROCESS CONTROL AND OPTIMIZATION

This module is responsible for creating feed-back to business stakeholders on how executed processes perform and provide event logging against each failure. The capabilities this module should support are:

• Business activity monitoring
• Reporting tools

### 3.3.5 PRESENTATION

The primary purpose of this module is to provide a graphical user interface to the business logic created in different layers. The capabilities which this module should provide are:

• Personalization
• Syndication
• Rich graphical control

### 3.3.6 IDENTITY MANAGEMENT

This module would be responsible for providing a structured mean to safeguard the resources in Service Oriented Architecture from unauthorized usage. The capabilities this module should support are:

• Authorization
• Authentication
• Role management & Auditing

### 3.3.7 DESIGN-TIME SERVICE MANAGEMENT

This module controls those design-time activities that are performed in accordance with governance rules. The capabilities which this module should provide are:

• Service repository/ service registry
• Life cycle management

### 3.4 SPECIFIC REQUIREMENTS

Apart from the capabilities covered in the technical reference architecture, an enterprise (in our case Vattenfall Nordic) has some specific needs which are vital for the business to function properly and which need to be considered as well. The specific needs are as follows:

1. **Rapid Application Development:** Rapid application development (RAD) increases developer efficiency in developing application without having to configure the environment. This need is specific to development platform’s support for rapid application development.
2. **Product Documentation:** Proper documentation should be available, for referencing environmental issues and functionality information. This is important as during development, it’s very important to have access to documentation of the framework and its provided functionality.
3. **Product Support:** Support is must for an enterprise, which is thinking of implementing SOA. SOA is a paradigm which means there could be issues never encountered earlier, for the resolution of such issues a proper vendor support platform should exist.

4. **Development Platform:** There should be a proper development platform available, which integrates all the functionality required for the implementation of SOA components. A development platform saves time for configuring the environment, which could be a real time consuming job.

5. **Cost and Licensing:** This is the most important requirement for an enterprise, i.e. cost of SOA suite and the yearly cost required to implement SOA. This function block would be one of the main decisive factors for our evaluation scheme for the implementation of SOA.

6. **Connectivity Adapters:** Connectivity across systems is the main concern for an enterprise. The easy availability of connectivity adapters for different systems is essential in making a decision for which SOA solution vendor to choose.

7. **Open Source Software:** Open Source is pretty much free of cost and that’s what an enterprise wants. How cost can be saved by not paying large amount of money to vendors. This is the second most important function block for this thesis.

8. **Business to Business Integration:** Business to business integration is becoming quite common, as everything is becoming available online. Integration with systems like Health Care requires specialized formats of data to be processed and SOA should support integration of these sorts of systems and formats.

9. **Integration with Existing Applications:** Integration with existing systems is essential for an enterprise to keep its current infrastructure running in a smooth manner. This is also an important requirement that needs to be taken into account.

10. **Technology Deployment Effort:** This requirement focuses on how difficult would it be to deploy the SOA solution on different servers which could be located at different locations on the globe.

11. **Product Scalability:** Easy and fast scalability is must for SOA i.e. a system should be able to scale without affecting any running processes.

12. **Compliance with Standards:** Interoperability requires standardization among vendors, so this also is a key factor for integrating or exposing current systems as services within the SOA implementation.

13. **Governance Platform:** Having a platform which is spread across different servers, becomes quite difficult to govern. A proper governance platform should be there to manage all activities across all platforms and within the SOA platform.

14. **End to End Business Process Monitoring:** Business process monitoring is the key to retrieve right information at the right time. There should be a mechanism in place which can be used to monitor states of business processes and should provide capability to correct errors or faults which can be encountered during a business process life cycle.
Chapter 4  
SOA SOLUTION BY VARIOUS VENDORS

There are a number of vendors who provide solutions which map to Service oriented architecture concept, the main solution providers are

1) RedHat JBoss [61]

2) Microsoft [62]

4.1 REDHAT AND SOA

For an enterprise to address the day to day business advancements and challenges, it requires a system which is flexible enough to incorporate change, which is easily scalable and most importantly affordable. Service Oriented Architecture provides all these capabilities which cater all business needs and this is what Red Hat delivers. Red Hat provides solutions for operating systems, middleware products and rich and user friendly graphical user interfaces. These solutions can be deployed either on cloud, hybrid systems or on-premise with minimal change. In short, Red Hat provides solutions which are easily scalable, which have proper support and are affordable as compared to other proprietary SOA solutions.

SOA requires different parts to be combined to provide a scalable and flexible infrastructure. As the business requirements and high level IT application designs have been finalized, software engineers can start developing applications, SOA services, user interfaces and integration models. Red Hat provides a robust and full fledge solution comprising of all the components required to implement SOA infrastructure.

RedHat JBoss Enterprise provides a full middleware solution comprising of different enterprise platforms. This middleware enables the development, deployment and management of SOA based applications which can be easily scaled up to different platforms. This suite includes JBoss Enterprise SOA Platform, JBoss Enterprise Portal Platform, JBoss Developer Studio, JBoss Enterprise Application Platform and JBoss Operations Netwrok. All these different platforms help in creating, deploying and managing applications in a more structured and cost effective way. Moreover, all these platforms are optimized to work together.

![Figure 11: RedHat JBoss Enterprise Middleware](20)
Enterprises are usually dependent on each other for providing different services to their customers. They operate under a collaborative umbrella to execute different business processes, which consists of many stakeholders and IT assets. The problem which arises within this umbrella is the connectivity of these assets with stakeholders and the alignment and orchestration of execution businesses processes. Not having a proper aligned business system leads to addition of cost, errors and lost opportunities in the business domain. These issues can be solved using SOA in a standardized way which would help increase ROI for businesses and also alleviate business collaborations.

JBoss Enterprise SOA Platform combines all the latest capabilities required for implementing SOA infrastructure. It provides a next generation enterprise service bus, infrastructure for crating automated business workflows and business rules technology to resolve integration and collaboration issues. JBoss Enterprise SOA Platform facilitates IT to integrate existing systems based on message oriented middleware and enterprise application integration, and also supports the integration of next generation of these systems including SOA based systems and Electronic data interchange (EDI) based systems. Red Hat offers an open source subscription for JBoss Enterprise SOA Platform which helps an enterprise to achieve an affordable, open and simpler platform.

JBoss Enterprise SOA Platform supports the following core functionalities for constructing a SOA based infrastructure:

**JBoss Enterprise Service Bus**

JBossESB fulfills the criteria of being the next generation of EAI. It consists of all the functionality which a typical EAI provides i.e. Transaction Manager, Integrated Development Environment, Business Process Management, Connectors, Business Process Monitoring, Security, Messaging Service, Human Workflow User Interface, Metadata Repository, Naming and Directory Service, Distributed Computing Architecture and Application Container. Moreover, JBossESB is part of a SOI (Service Oriented Infrastructure).

**Rosetta and JBoss Enterprise Service Bus**

At the core of JBossESB is Rosetta. Rosetta is an ESB that has been commercially developed for more than three years. It was developed to overcome user connectivity issues. It provides easily configurable infrastructure and toolset for interoperability of legacy systems, message interaction in a synchronous or asynchronous way, general purpose repository and also supports different transport protocols and logging of interactions. This
The project was extremely successful as Rosetta is used in critical systems based on Oracle Financials on multiple platform environments such as IBM mainframe operating systems and Oracle databases.

As JBossESB is based on Rosetta, so the main focus was to develop a methodology and toolset that would help in isolation of business logic from transport and other triggering mechanisms. It should provide a flexible way to plug in ad hoc business logic and data transformation. Finally, it should be easily customizable for future users to mold it according to their own needs and should be able to incorporate their own custom action classes [22].

FEATURES OF JBOSS ESB

JBossESB provides the following core functionality which is inherited from Rosetta [22]:

1. **Message Listener and Message Filtering code**: Message Listeners pick up incoming messages which can be JMS messages or any file type messages and forward these messages to a message processing pipeline. The pipeline processes and filters out the messages and routes them to another message endpoint.

2. **Data transformation**: JBossESB provides data transformation for different formats using the SmooksAction action processor.

3. **Content Based Routing Service**: JBossESB also provides content-based routing, which routes messages based on their content and not according to the specified endpoint destination.

4. **Message Repository**: JBossESB has a repository for saving messages/events exchanged within the ESB. The repository has the ability to store messages, process states, business process definitions or WSDL and service level agreements.

The above mentioned features are provided in the ESB through a set of business classes, adapters and processes. Clients and services interact with each other using messages which can be of a wide range including JMS, flat file system and email. JBossESB supports all these formats.

JBOSSES ESB AND ITS RELATIONSHIP WITH SOA

SOA is a paradigm or architectural style, and is not technology specific. To implement SOA, a change is required on how people work and interact with the underlying infrastructure. Red Hat JBossESB provides a base for SOA based systems, and SOA applications can easily be deployed upon these systems using this bus as the base integrator. JBossESB is continuously evolving to accommodate and provide user friendly GUI’s for integrating with different tools, runtime management and service cycle.

JBoss BUSINESS PROCESS MANAGEMENT

Business Process Management (BPM) gives the developer a way to design and create workflows, business process and sequence flows in an automated way. BPM solutions have three main components: a process execution engine, services which facilitate the engine to interact with other systems and tools that provide business process development and monitoring [23].

JBoss provides a Business Process Management (JBPM) Suite. JBoss BMP is a standard java application and does not require an application server to run on. This makes it easy to be used by enterprises, as it can be deployed as a web application or can be used as a standalone Java application. Moreover, it is not only focused for developers to design and create business processes, but also supports process management features which can be used by non-technical people i.e. business analysts also.
JBoss BPM works as an orchestration engine. The BPM engine and workflows enable the creation of business processes that can coordinate among people, applications and services. It also provides a mechanism for developing workflow which use the process engine to run and these workflows can be combined into other workflows to form an orchestration.

**FEATURES OF JBOSS BPM**

JBoss BPM provides the following core features to fully automate business processes [23, 24]:

1. **Process Engine:** At the core of jBPM is the process engine which can run on any Java environment. This engine executes different processes and also keeps track of the process state. This engine can be used as remote service or can be integrated inside an application. It also supports integration with JBoss BRMS engine.

2. **Process Monitor:** jBPM also provides the functionality to monitor the current end-to-end states of processes. It automatically records all history of each business process execution. Using this recorded history, reports can be generated which can provide an in-depth analysis of both the business process execution and any errors which were encountered.

3. **Process Languages:** jBPM is based on the Process Virtual Machine (PVM) [55]. This allows jBPM to support multiple languages. Currently jBPM supports BPMN 2.0 [56] and jPDL [57]. As being based on PVM, jBPM becomes quite flexible to incorporate new process languages.

**4.1.2 JBOSS ENTERPRISE BRMS**

JBoss Enterprise Business Rules Management System (BRMS) is an open source business rules management system. It provides a fast and easy way to develop and change business policies and rules. JBoss Enterprise BRMS contains a highly efficient rules engine and a user friendly rules deployment and management system. It also provides a repository for storing rules. This rules system facilitates business analyst and auditors to manage and view the business rules available within the application infrastructure. It also provides business analysts and SOA rule developers the flexibility to verify whether the rules implemented are in accordance with the business policies or not [25].

JBoss Enterprise BRMS provides the ability to reduce development time for application updates, SOA deployments and business processes. These all reductions are done with the help of latest business rules and policies. This allows enterprises to quickly respond to business change requests and update their IT applications to accommodate the change, which leads to business agility required to respond to competition and new business challenges.

**FEATURES OF JBOSS ENTERPRISE BRMS**

JBoss Enterprise BRMS provides the following features which help in overcoming business challenges:

1. **Business Rules Engine:** The Business Rules Engine (BRE) is the core of JBoss Enterprise BRMS. This engine is based on the Rete algorithm [26]. BRE implements this algorithm fully and also provides high performance indexing and optimization. It also supports dynamic addition and removal of rules. In addition, it supports temporal rules which get fired at a specified time period or when some specified constraints are encountered. The rules engine also consists of a model which logs rules execution time and sequence, this information is used for audit logging, business event tracking and management.

2. **Rules Authoring:** JBoss Enterprise BRMS provides a rich internet application user interface for process owners, administrators and business analysts to develop and manage rules in a fast and easy way.
It also facilitates Java developers by providing the Drools Rule Language (DRL). This supports all the above mentioned features and as it is based on Java. It also provides field constraints and functions for developing rules and applying constraints to them. The Drools Rule Language is much like a natural language i.e. rules can be specified using human readable language; this is achieved by mapping a property file which lists all the mapping for the natural language. JBoss Enterprise BRMS supports both natural language extension and Drools Rule Language.

JBoss Enterprise BRMS provides a Workbench IDE which is also integrated with JBoss Developer Studio. This IDE provides features like basic rule validation, syntax coloring, outline view, error reporting, RETE viewer, code completion and an audit viewer which exposes all the internal functionality of the rule engine.

3. **Rules Management**: JBoss Enterprise BRMS also includes a business rules management repository and web-based administration console. The rules repository enables storing, managing and version controlling of rule artifacts like fact models, domain specific language definitions, rules and tests. The web-based administration console helps developers, administrators and other users to easily manage the rules within an application and also facilitates in SOA deployments. The combination of repository and web-based administration greatly boosts IT administrative productivity and business agility.

### 4.1.3 JBOSS ENTERPRISE APPLICATION PLATFORM

JBoss Enterprise Application Platform (EAP) is a platform specifically designed to host enterprise Java applications and services. Its main focus is on developing, managing and hosting highly transactional applications built using Java EE programming technology, by providing an application platform which is built using different open source technologies [27].

JBoss Enterprise Application Platform provides a secure and reliable foundation that supports a number of popular Java programming models, which can use different underlying operating environments. It also provides different deployment models which spans from traditional application deployment to internal and external cloud-based deployments. JBoss EAP is built on open standards i.e. it integrates JBoss Application Server, JBoss Hibernate, JBoss Seam and other Java technologies to form a complete enterprise application development and deployment platform.

### FEATURES OF JBOSS ENTERPRISE APPLICATION PLATFORM

JBoss EAP provides the following features which cover all the requirements of an enterprise [28]:

1. **Enterprise-class performance & scalability**: JBoss EAP provides clustering and high-availability for applications by supporting features for failover, intelligent load balancing, caching and distributed deployments. These features add to the credibility of JBoss EAP for hosting highly scalable Java applications.

2. **Second-generation, service-based architecture**: JBoss EAP is based on JBoss Microcontainer architecture [64]. JBoss Microcontainer is an improved version of JBoss's JMX Microkernel, it supports faster class-loading, lifecycle management, performance, virtual deployment framework and flexibility for a number of programming and component models including Java EE, POJOs, OSGi and Spring. In addition, it allows the separation of enterprise services from the core application server runtime engine, which helps in developing a highly configurable Java application platform.

3. **Integrated frameworks**: JBoss EAP is a combination of different frameworks, which help in building web 2.0 applications and allows easy integration of technologies for developing Java applications from simple desktop and web based applications to highly transactional J2EE applications. All the integrated frameworks are certified and kept up-to-date, making sure that development and deployment are operationally consistent.
4. **Enhanced application security**: JBoss EAP follows Common Criteria [58] for its security model and implements all the necessary Java security requirements and standards. It also includes security features like security negotiation, audit and password masking, instance-based access control and integration support with single-on solutions.

5. **Simplify application management and configuration**: JBoss EAP provides an administrative console which allows users to configure, manage, execute and see performance metrics of running applications. To manage enterprise-wide applications it can integrate with JBoss Operations Networks.

**COMPONENTS OF JBOSS ENTERPRISE APPLICATION PLATFORM**

JBoss EAP integrates different frameworks into one single distribution combining all the features required to build and host highly transactional Java EE applications. The integrated frameworks are [28]:

1. **JBoss Application Server**: This is the core of JBoss EAP and is certified to host Java EE 5 based applications. In addition, it supports other programming frameworks and component models, like Spring and POJO’s. It also supports features for improving performance & increasing scalability.

2. **Hibernate**: JBoss EAP supports integration of Hibernate, which is an object/relational mapping and persistence (ORM) framework. Hibernate provides applications to model their objects in accordance with the relational database schema. In addition, hibernate also provides features to query data without any need to do manual connection using JDBC or write SQL to query data.

3. **Seam**: This application framework facilitates developers to develop rich web applications. The programming model used within seam helps in integrating different programming frameworks. Using different standards seam can easily combine technologies like business process management, Asynchronous JavaScript and XML (AJAX), Enterprise Java Beans (EJB), Java Server faces (JSF) and Java Portlets.


**4.1.4 JBOSS ENTERPRISE PORTAL PLATFORM**

Enterprises are moving towards the next generation of integration platforms such as SOA. This shift brings a change in the way front-end portals are developed. It is difficult to provide end users with an easy to use front-end that benefit from the underlying SOA infrastructure. JBoss Enterprise Portal Platform (EPP) facilitates enterprises to quickly build and deliver secure and dynamic web-based user interfaces.

JBoss EPP is deployed on JBoss Enterprise Application Platform as part of JBoss open choice. JBoss open choice lets Java application developers to choose among different frameworks, API’s and runtimes according to their needs, without having any impact on scalability, reliability and management of applications. In addition, JBoss open choice provides modular based deployment for enterprise applications regardless of the programming model or deployment style.

**FEATURES OF JBOSS ENTERPRISE PORTAL PLATFORM**

JBoss EPP is a reliable and high performance web portal platform. It provides the following features [29]:

1. **Maximize Existing Investments**: Return of investment (ROI) is critical for every enterprise and it requires services to be developed and delivered in a short period of time. JBoss EPP enables a fast return on investment by providing a platform which can be installed and integrated with JBoss Developer Studio in minutes and developers can start creating protlets applications instantly.
2. **Easy To Manage Site Tools:** Enterprises require fast and efficient ways to develop and manage web sites. JBoss EPP solves this by providing functionality which includes easy to manage site controls, contextual menus, site maps and the capability to integrate rich Internet applications (RIA) directly into portals.

3. **Easily Create and Manage Web Content:** JBoss EPP provides a Site Published *(powered by eXo)* feature, which allows fast and easy creation and management of web content on portals. This functionality is similarly to a content management system, which removes the need for a separate content management system for many portals.

4. **Simple, Robust and Reliable:** JBoss EPP provides a simple, robust and reliable platform for creating frontends. It works best with JBoss Enterprise Application Server by utilizing its capabilities and features of high performance and scalability for application.

### 4.1.5 JBOSS OPERATION NETWORKS PLATFORM

JBoss Operations Network (ON) provides a management platform for JBoss Middleware suite. There is a plug-in-pack for each middleware product (SOA-P, EWS and EAP), which gets integrated with the required platform. JBoss ON provides a single view for managing and analyzing applications running using JBoss middleware infrastructure, JBoss middleware applications, operating systems and web tiers. Moreover, JBoss ON Server and its modules also provide a centralized, end-to-end application management solution [30].

### JBOSS OPERATION NETWORKS SERVER

The JBoss ON Server is the main managing console for JBoss middleware environment. It provides a rich web based user interface for monitoring and controlling applications distributed across different environments. In addition, it provides a custom dashboard, authentication using LDAP, user roles for administrative access and control, API for remote accessibility and secure command line interface (CLI) support.

### MODULES OF JBOSS OPERATION NETWORKS

JBoss ON has different modules which provide different capabilities. Following are the modules which are provided by JBoss ON [30]:

1. **Inventory module:** This module keeps track of IT assets ranging from operating systems (AIX, Solaris, Linux, and Windows), application servers (JBoss SOA-P, EAP, EWS), and services (ESB services, Message Driven Beans, EJB) by maintaining a centralized repository. Using this repository a better view of running business applications can be generated. The core features which inventory module provides are auto/manual discovery by server agents, dynamic resource grouping based on rules and resource organization.

2. **Administration module:** This module provides a centralized location for executing control functions. It supports sequential execution, on-demand execution or scheduled execution of functions. It also keeps a record of the executions and the users who initiated the executions for audit purposes. The core features which administration module provides are ability to start/stop/restart applications across JBoss middleware platform, full or limited access to all configuration files, event based function execution and full versioning control of configuration files.

3. **Content module:** This module is responsible for notifying different environments of new patches and software updates. This really organizes and speeds up the update process of the applications, plus it supports versioning control, which makes management of updates quite easy to perform. The core functionality provided by content module is accepting/rejecting and executing of new patches/updates, applying updates on individual components, and version control and roll-back option for application deployment.
4. **Monitoring module:** This module is one of the most important modules within JBoss ON. It provides advanced monitoring capability for the entire JBoss middleware suite, operating systems and other web resources. This module equips administrators with the capability of predicting potential issues before they can even occur. In addition, it provides historical view, graphing and charting capabilities, defining fault threshold and automatic fault correction. The monitoring module is fully integrated within JBoss ON platform and it constantly collects information through agents to build a behavior model. The core features provided by monitoring module are configurable/custom statistics, email alerts, configurable data storage, control actions on alerts, problem detection and full alert audit report.

### 4.1.6 JBOSS DEVELOPER STUDIO

JBoss Developer Studio Portfolio Edition (DS) is an open source development environment. It comprises of all the JBoss Enterprise Platforms (EAP, SOA-P, EPP) and Tools (Eclipse) available. It also includes JBoss ON and provides access to OpenJDK. JBoss DS facilitates developers in every aspect of building rich web application, transactional enterprise application and even SOA-based applications. Developers have a wide range of technologies to choose from e.g. Hibernate, JBoss Seam, RichFaces, Drools, jBPM and many others.

**FEATURES OF JBOSS DEVELOPER STUDIO**

JBoss DS provides and supports the following functionality and features [31]:

1. **Fully integrated JBoss development environment:** JBoss DS is a fully featured development environment which contains both tooling (Eclipse, Eclipse Tooling) and runtime components (JBoss Enterprise Platforms).

2. **Based on Eclipse 3.5.2 (Galileo) and WTP 3.1.2:** For Java, Java EE and web development JBoss DS includes the latest Eclipse 3.5 (Galileo) and Web Tools Project (WTP 3.1.2) releases.

3. **Integrated with JBoss Enterprise Platforms:** JBoss DS integrates with different versions of JBoss Enterprise Platforms (EAP, SOA-P, BRMS, EPP) which are supported by Red Hat for five to seven years allowing developers to develop their application on a platform which has long term support available.

4. **Updated Seam support:** JBoss DS includes latest tools for Seam 2.2 [59], which is a state of the art application framework.

5. **Visual Page Editor:** JBoss DS supports a Visual Page Editor for editing web pages ranging from AJAX-based RichFaces to Struts components.

6. **AJAX capabilities with 120 RichFaces components:** It also provides tooling support for RichFaces, which is a component library for integrating AJAX capabilities into web pages using JSF.

7. **Business process modeling tools:** JBoss DS provides a powerful tool for business process modeling and workflow creation. Moreover, it also allows to manage workflows processes and interaction of these processes with human tasks.

8. **Business rules management tools:** JBoss DS supports integration of Drools which is a rule engine. It provides graphical user interface for the creation and editing of business rules, decision tables and rule resources. These rules can also easily be imported to process modeling section to be used within different business processes.

9. **SOA tools:** JBoss DS supports a wide set of tools for creation and deployment of SOA-based applications. The SOA tools include all the required components like ESB for interoperable integration, Smooks for data transformation etc.
10. **Portal tools:** JBoss DS has tools to create and deploy simple SEAM/JSF portlets on the same JBoss Enterprise Portal Platform.

11. **Spring tools:** JBoss DS also includes the Spring IDE for developing Spring applications.

12. **Optimized JBoss Application Server adapter:** JBoss DS also supports features for incremental deployment on different environments where JBoss Application Server is running.

13. **Red Hat Enterprise Linux and RHN access:** JBoss DS can be used to develop Linux kernel, it also supports Linux development tools.

### 4.1.7 JBOSS SOA GOVERNANCE

JBoss provided SOA Governance using a combination of three products as listed below:

<table>
<thead>
<tr>
<th>RedHat JBoss Products</th>
<th>SOA Software Added Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBoss Enterprise Application Platform</td>
<td>JBoss EAP provides Governance, federation, Metadata management, Mediation for interoperability, Policy enforcement, implementation, and monitoring Security policy implementation and enforcement of Automated PKI key distribution.</td>
</tr>
<tr>
<td>JBoss Enterprise SOA Platform (including ESB)</td>
<td>JBoss Enterprise SOA also supports the same features as provided by JBoss Enterprise Application Platform and feature of an Enterprise Service Bus (ESB).</td>
</tr>
<tr>
<td>JBoss jBPM</td>
<td>JBoss jBPM provides governance, federation, dynamic policy implementation and Abstraction of applications from complex policy operations.</td>
</tr>
</tbody>
</table>

### 4.2 MICROSOFT AND SOA

Microsoft has long been associated with providing enterprise application integration (EAI) tools and platforms. It has concentrated on the “middle out” approach for integration of different systems, as for SOA they do not provide any dedicated platform or product. SOA can be achieved in Microsoft by a combination of different products. Microsoft has been working with the concept of service orientation since 1999, and has been involved in the development of the Web services model, which had great impact on the way application architectures were designed. Later on, with the release of .NET Framework and SOA tools, Microsoft products became SOA compliant. Since then, Microsoft has concentrated on providing enterprises of all sizes with solutions which are SOA based and which aim to optimize business processes and strive for greater business agility and quicker time to value.

Microsoft has been concentrating on web services for the service part of SOA. For this it has been working with vendors such as IBM and BEA to formulate a collective and standardized WS-* architecture. Web Services Interoperability Organization (WS-I) [32] was formulized with the collaboration of Microsoft and IBM to support interoperability across platforms, operating systems and programming languages.

Microsoft has been using service orientation across its entire technology stack, ranging from developers tools integrated with .NET framework for the creation of Web Services, to server products such as BizTalk Server, Windows Server AppFabric and Microsoft Share Point which use these web services by connecting and orchestrating them to process business flows. Moreover, these web services can be consumed by different composite applications running on Intranet, Internet or by rich client applications.

### MICROSOFT APPLICATION PLATFORM FOR SOA

Microsoft has two main products which facilitate the implementation of SOA. The main products involved in implementing SOA are:
1. Microsoft BizTalk Server
2. Windows Server AppFabric

4.2.1 MICROSOFT BIZTALK SERVER

BizTalk Server is Microsoft’s main integration platform which supports capabilities of connectivity, native messaging, broker, service orchestrations and business process executions. BizTalk Server was originally designed to cater enterprise application integration issues, but has evolved from being an integrator to a product which now supports comprehensive set of technologies that facilitate in creation of SOA based systems. It provides support for connectivity to platforms, applications, and devices to consume, expose and compose new services. In addition, it also provides service inventory architectures which are built using orchestration and enterprise service bus style platforms. Moreover, it includes tools which enable enterprises to expose their business processes as services, and gives them the capability to connect to both proprietary and standard based systems. This makes BizTalk Server a central part of SOA strategy provided by Microsoft [5, 34].

Figure 13: Microsoft Application Platform for SOA

Figure 14: Microsoft BizTalk Server and SOA
BizTalk Server provides the following features either independently or by simple addition of components which help in implementation of SOA:

1. **Enterprise Service Bus Toolkit**: Microsoft ESB Toolkit is an add-on component for BizTalk Server, which helps in achieving functionality of an enterprise service bus (ESB) and allows BizTalk to support a loosely coupled messaging architecture for service-oriented architecture (SOA) based applications.

2. **Business Rules Framework**: Microsoft BizTalk Server includes the Business Rules Framework as a stand-alone .NET compliant library, which has different modules and components.

3. **BizTalk Orchestration Engine**: The BizTalk Orchestration Engine is responsible for coordinating and scheduling message processing and performing complex logic on the message as it passes through an orchestrated business process.

4. **Business Activity Monitoring**: Business Activity Monitoring (BAM) is a fundamental part of Microsoft BizTalk Server, giving it the capability to track and monitor information generated by different business processes.

5. **BizTalk Messaging Engine**: BizTalk Messaging Engine is responsible for receiving inbound messages, parsing and identifying message format, evaluate routing criteria, delivery to destination and status tracking.

6. **Business to Business Integration**: BizTalk provides different capabilities which simplify the integration of business processes with external enterprises. BizTalk Server natively provides support for Electronic Data Interchange (EDI) data protocols and Applicability Statement 2 (AS2) protocol.

---

**MICROSOFT AND ENTERPRISE SERVICE BUS**

For sometime enterprise service bus (ESB) was an architectural pattern for Microsoft and not a separate product. In Microsoft’s perspective the function provided by an enterprise service bus could be achieved by different Microsoft products including BizTalk Server and Windows Communication Foundation (WCF). As the concept of ESB got popular, Microsoft released its first ESB Guideline in 2006. This ESB Guideline provided architectural guidance describing how to achieve ESB functionality using Microsoft Platform. This guideline shaped up into a component for BizTalk Server known as the ESB Toolkit [5].

Microsoft doesn’t offer a standalone enterprise service bus; it comes as an add-on for BizTalk Server and is known as the BizTalk ESB ToolKit.

**MICROSOFT BIZTALK ESB TOOLKIT 2.0**

Microsoft ESB Toolkit is a set of components, services and architectural guidance which extends the capability of BizTalk Server to start acting as an Enterprise Service Bus. These new capabilities allow creation of service-oriented applications that incorporate itinerary-based service invocation, dynamic resolution of endpoints, WS-* integration, exception management and auditing, and SOA governance using third party tools [5, 35].
Microsoft ESB ToolKit provides the following core services [5, 35, 38]:

1. **Resolver service**: The resolver service is responsible for providing capabilities of endpoint resolution at runtime and dynamic routing. It acts as a service registry making services available in a heterogeneous environment. ESB ToolKit provides endpoint resolvers for the following: XPath, UDDI 3.0, UDDI 2.0, Static, WS-MEX and Business Rules Engine (BRE). In addition, it also provides resolvers for other artifacts such as, Static (for itineraries), BRE (for itineraries), Composite, BRE (for maps) and SQL Server.

2. **On-ramp service**: This service promotes loose coupling by providing dynamic message transformation and translation. The web service consumer only sends messages (SOAP, WCF, JMS, WMQ, FTP, HTTP, Flat File, XML, or any other custom formats) to the ESB and they automatically get transformed to BizTalk compliant messages.

3. **Itinerary services**: These include orchestration-based and messaging-based services for performing itinerary-based routing. It also allows creation of custom services for itinerary-based routing.

4. **Transformation service**: The transformation service helps in protocol transformation i.e. the services hosted on different platforms can interact with each other using different protocols including WS-* standards for Web Services. This transformation also helps non-BizTalk applications to access and utilize BizTalk capabilities.

5. **Exception Management service**: ESB ToolKit provides an exception management service allowing non BizTalk applications to publish faults and errors using the default BizTalk exception management mechanism.

6. **BizTalk Operations service**: This service provides information about the current state of BizTalk artifacts.

---

**MICROSOFT BizTalk Business Rules Framework**

Business rules are conditions on which decision are made in business processes, they can be considered as conditions that determine which business action needs to be performed on the basis of values in factual variables. In conventional application design, business rules are implemented using methods. This makes it difficult to use them in other applications and business processes.
Service Oriented Architecture (SOA) has changed the way applications are designed and created. SOA changes the way applications interact by introducing messaging as the way of communication and every decision is made on the basis of message content. Applications that are prone to constant change in their business really benefit from SOA. The separation of business rules from code makes them easily accessible by other applications and processes. Moreover, it increases business agility and business processes become more flexible to change and adaptive to new requirements.

Microsoft provides the Business Rules Framework as a stand-alone .NET compliant library, which is also by default included into BizTalk Server. Business rules can also be integrated in orchestrations to extend their functionality e.g. changing business logic can be encapsulated into rules, rules can be used to determine variable delays and can also be used to determine execution path of a business process [36, 37].

**MODULES OF BizTALK BUSINESS RULES FRAMEWORK**

Microsoft BizTalk Rules Framework consists of the following modules [36, 37]:

1. **Business Rules Composer**: The Business Rule Composer is used to create different rule vocabularies. These vocabularies are user-defined names for the facts that are used in rule conditions and actions. These vocabularies can be bound to different data sources e.g. .NET class members or XML schema or database elements. Moreover, it allows to version vocabularies which makes it easy to audit and manage. Composer can be used by developers, administrators and information workers to publish vocabularies.

2. **Run-Time Rule Engine**: The Run-Time Rule Engine is responsible for processing rules created as declarative statements, grouping rules together that combine to form a policy without rearranging them or redefining them and resolving conflicts among rules and allowing support for forward chaining of rules.

3. **Business Rule Policy**: Business rule policy is a logical grouping of different business rules. Every rule in this group corresponds to a condition, which can be evaluated to true or false and if it results in true the corresponding rule actions are performed. This allows redeployment of the rule policy only if a change occurs in any of the business rules used within it and no need to redeploy the whole orchestration.


**MICROSOFT BizTALK BUSINESS ACTIVITY MONITORING**

Monitoring of business activities and generating information regarding business processes can prove to be vital for the success of a business. The demand for right information at the right time has increased and businesses require quick and accurate information from their underlying integration solutions. This information enables enterprises to make timely decisions for the improvement of their business processes and services provided to customers.

Microsoft caters this business need by providing a business activity monitoring tool. Business Activity Monitoring (BAM) is part of Microsoft BizTalk Server, giving it the capability to track and monitor information generated by different business processes. BAM provides information about business processes state, usage trends and critical conditions within different system of processes. In addition to this, BAM also provides an application programming interface (API) that enables developers to monitor information which is not in the scope of BizTalk processes [39, 40].

**FEATURES OF BizTALK BUSINESS ACTIVITY MONITORING**

BizTalk Business Activity Monitoring gives the following features [39, 40]:

---

32
1. **BAM Activities**: BAM activities are units of work which are used to gather business data. These activities can also be used to see the overall progress of a business process. As these activities are for monitoring purpose they do not affect the BizTalk solutions. In addition, activities can be created for multiple systems and also be related with each other. BAM facilitates to generate one view of such an orchestrated business process.

2. **BAM Views**: BAM views helps in organizing and displaying information collected through BAM activities. BAM views can differ depending on context, as one business process can be initiated by multiple users. There would be only one BAM activity against each process, to see logs of each user there could be multiple BAM views displaying information according to context defined.

3. **Aggregating and Filtering Data**: BAM provides different interceptors for collection of information from pipelines, messages and business processes orchestrations. Using its aggregation capability it can predict usage trends within a business. These trends can be viewed using BAM views, which can help enterprises in making critical decisions. It also provides the capability of filtering data i.e. users can request for data according to their requirements.

**BIZTALK BUSINESS ACTIVITY MONITORING AND SOA**

As SOA is all about distributed processes and services running on disparate systems, full process execution visibility becomes a vital component in managing this infrastructure. Microsoft BAM facilitates developers to create a single activity for monitoring huge business processes. This helps in implementing a single view of information about the process execution rather than using different tools to generate a view [40].

**MICROSOFT BIZTALK ORCHESTRATION ENGINE**

A business process is a combination of rules and actions, which function together to meet some specific business goals. Business processes coordinate with other business processes which can be on other systems such as Enterprise Resource Planning (ERP), Customer Management System (CRM), or exposed as web services. This processes execution can span over hours, days, weeks or even months to complete.

Microsoft BizTalk includes such a feature to support orchestration of different processes. It also provides different tools and services which makes it easy to design, automate and manage business processes. BizTalk orchestration supports two types of transactional programming models; the first is Atomic transactions, these transactions would be rolled back to previous state if they do not successfully complete, the other model is Long running transactions, these can stay running for days or longer and they require custom exception handling to recover from an exception [41, 42].

**FEATURES OF BIZTALK ORCHESTRATION ENGINE**

BizTalk Orchestration Engine provides the following features [41, 42]:

1. **Orchestration Designer**: This designer gives a drag and drop graphical user interface to construct orchestrated business processes. The different shapes just need to be configured accordingly for the orchestration to work properly.

2. **BizTalk Orchestration Engine**: The BizTalk orchestration engine executes the orchestrated processes created using BizTalk Orchestration Designer and provides monitoring services for these processes. In addition, it maintains states of orchestrated instances, performs optimization of resources, provides reliable shutdown-and-recovery system and supports routing pattern called correlation.

3. **Calling External Assemblies**: BizTalk Orchestration also provides support to call external assemblies, i.e. if functionality is available in another .NET application, it can pass parameters to those applications through method calls, making it possible to integrate external applications into BizTalk process.
4. **Service Integration Scenarios:** BizTalk Server fully supports calling Web services and Windows Communication Foundation (WCF) services even from within an orchestration. This enables developers to create WCF services which can be used within disparate business processes and also publish locally created business processes as services on the internet. BizTalk provides a Services Publishing Wizard to publish WCF services that run in Internet Information Server (IIS).

5. **BizTalk Service Integration Capabilities:** In addition, to the WCF adapters, BizTalk Server also supports consuming WCF services, publishing orchestration as WCF services, publishing schema as a WCF service, publishing receive location metadata as WSDL and integration with ASMX Web services.

**BIZTALK ORCHESTRATION IN SOA DESIGNS**

SOA has a number of principles and business process automation is one of the important ones. Microsoft BizTalk Server supports business process automation in an extensive way. Service aggregation SOA pattern is the most used SOA pattern for business process automation and usually it is designed and implemented using BizTalk Server.

**MICROSOFT BIZTALK MESSAGING ENGINE**

At the heart of BizTalk Server are the Orchestration and Messaging Engines, which are responsible for integrating and exchanging messages among services based on different protocols and standards. BizTalk Messaging Engine is responsible for receiving inbound messages, parsing and identifying message formats, evaluate routing criteria, delivery to destination and status tracking [43, 44].

**FEATURES OF BIZTALK MESSAGING ENGINE**

BizTalk Messaging Engine provides the following features [43, 44]:

1. **Publish/Subscribe Model:** BizTalk Server uses the publish/subscribe model to route messages. The publish/subscribe model allows services to be published to a central store (in BizTalk it’s the MessageBox database). The consumers/subscribers subscribe to the published services, when a message is received the central store forwards it to all the subscribers. This publish/subscribe model allows services to be added/modified without having any impact on the application design.

   BizTalk Server routes messages on the basis of matching the message context properties of messages with expressions. The message context is persisted across the message life cycle and if needed also stored in the MessageBox database.

2. **Building Schemas:** BizTalk Server supports a number of schemas that it can process. A schema can be either predefined or might require to be created using a XML, flat file document structure or standard EDI document. BizTalk by default supports schema types for XML, Flat-File and more than 8000 schemas for EDIFACT and X12 message standards, which are used for integration with EDI applications.

3. **Mapping Data:** BizTalk transforms inbound messages from one schema format to another using BizTalk map convert capability. These transformations can be simple or complex. Maps are usually defined at design time as they link fields within the two schemas which are being mapped.

4. **Connecting Through Adapters:** BizTalk needs adaptors to interoperate with systems which are not direct BizTalk message format compliant. Adapters are add-ons to BizTalk Server which enable BizTalk to communicate with different types of systems through standard-based protocols and with applications that correspond to proprietary communication standards. Most adapters support send and receive operations, other support only one directional communication. BizTalk supports native BizTalk
Line-of-Business (LOB), WCF Adapters, BizTalk partner adapters and Microsoft WCF Line of Business Adapter SDK.

5. Processing Messages through a Pipeline: Pipelines can be thought of message buffers with extended functionality. The purpose of pipelines is to prepare a message (by verifying it against schema, encoding and decoding, encrypting and decrypting and other processing that might be required) for processing by the server when it is received and similarly prepare it for sending when it has been processed. These can also be called within an orchestration.

MICROSOFT BIZTALK BUSINESS TO BUSINESS INTEGRATION

BizTalk Server supports integration of external companies or departments within a business process. BizTalk provides different capabilities which simplify the integration of business processes with external companies. BizTalk Server natively provides support for Electronic Data Interchange (EDI) data protocols and Applicability Statement 2 (AS2) protocol. It also provides storage facility for management of external partner’s information and BizTalk Server Accelerators [45, 46].

FEATURES OF BUSINESS TO BUSINESS INTEGRATION

BizTalk Business To Business integration provides the following features [45, 46]:

1. Electronic Data Interchange (EDI) and Applicability Statement 2 (AS2): Electronic Data Interchange (EDI) is the most used format for data exchange within businesses. EDI uses standard message format based on ANSI X12, UN/EDIFACT and messaging protocols. BizTalk EDI and AS2 receive processing capabilities are, parsing EDI interchange, performing HIPAA document splitting, validating messages, serializing EDI interchange, EDI document exchange and AS2 document transport etc.

2. EDI Parties: BizTalk can incorporate external partners by having separate configuration for that external partner with its own unique communication parameters. Both should agree on the message format which would be exchanged among BizTalk and the external partner. BizTalk Server includes several services and tools for management of external partner relationships.

3. BizTalk Server Accelerators: BizTalk Server accelerators are used to support and add functionality within BizTalk to support business integration from sectors ranging from high tech to healthcare. These help in reducing time, cost and effort in development and management of applications from these sectors.
4.2.2 MICROSOFT WINDOWS SERVER APPFABRIC

Services and service-oriented architectures is a concept and exists in many types of applications. One type is of having transactional components which are data driven in a highly distributed business logic and which get managed on a middle tier. As the demand of highly responsive and scalable applications (web applications or applications which provide services) increases, a solid architecture is needed to cater limitation associated with application performance and scalability. Microsoft’s AppFabric technology is developed to cater this middle tear application scaling and performance limitation problem.

Windows Server AppFabric is a set of integrated technologies that is used to manage, host and scale web services giving it a touch of distributed environment or service orientated architectural style. Windows Server AppFabric adds capabilities to Internet Information Server (IIS) by adding functionality like enhanced hosting, management, caching of web applications and middle-tier services. AppFabric hosting features add service management extensions to Internet Information Services (IIS), Windows Process Activation Service (WAS), and the .NET Framework 4.0. Hosting services included Hosting Administration tools which make it easier to manage, configure and deploy Windows Workflow Foundation (WF) based workflow services and Windows Communication Foundation based services (WCF), which in turn help improve performance and scalability of .NET applications especially ASP.NET applications [48, 49].
COMPONENTS OF WINDOWS SERVER APPFABRIC

Windows Server AppFabric provides the following components [49, 50, 51, 52]:

1. **Service Hosting**: Windows Server AppFabric was designed specifically to host WCF and WF services. AppFabric incorporates hosting capabilities of Windows Process Activation service (WAS), which enables it to host .NET framework services. Management of workflows developed using Windows Workflow Foundation is done by the Workflow Management Service, which has features of command queue functionality, durable timers and auto-start. In addition, it allows for the monitoring of applications and management of configuration files, security, performance and service endpoints.

2. **AppFabric Windows Services**: There are two services which Windows Server AppFabric installs for event tracing and collection, and workflow management. The Event Tracing and Collection service is responsible for collecting all the events raised by WCF and WF services hosted on AppFabric and store them into the monitoring store database of AppFabric. The Workflow management service, manages workflows by activating workflow instances, restarting abandoned workflows etc.

3. **Data Storage**: Windows Server AppFabric uses SQL Server express edition to store its data entities. SQL Server is the recommended store unit; otherwise they can be stored in memory or flat files also. It has two storage entities, the persistence store and the monitoring store. The persistence store is responsible for storing states of workflows and any metadata information associated with the workflows. The monitoring store keeps track of all the .NET framework events associated with WCF and WF, for monitoring purpose.

4. **Security Model**: Windows Server AppFabric provides security model which integrates with Microsoft security technologies to manage .NET 4.0 applications and services. It can integrated and use existing Windows Active Directory, .NET 4.0 security, IIS and SQL Server security model. While designing AppFabric security model, three roles can be created i.e. Application Server Observer, Application Server Administrator and Application Server Users. These roles can be assigned different permissions and can also correspond to Windows NT group accounts, IIS application pools and SQL server logins and database roles.
For analysis purpose, the function blocks identified in Chapter 3 would be used as the evaluation and comparison criteria for the vendor products. The main focus would be on the capabilities which are provided by each vendor product in accordance to the function blocks.

The comparison would be done for the following three areas:

1) SOA capabilities provided by vendors
2) Cost Effectiveness
3) SOA Competence

5.1 WEIGHTING CRITERIA

As these identified function blocks were generic, to compare and evaluate which vendor is better the importance of each individual function block was calculated in accordance with the need within Vattenfall Nordic. The following weighting criteria was devised to assign a weight to each function block, which would keep things simple and easily understandable.

The following would be our weighting criteria:

There are in total 3 points and the importance level increases with each point as follows:

- Highly Important = 3 points
- Partially Important = 2 points
- Least Important = 1 point

*Importance percentage is calculated = Importance Point(s)/ Total Point(s) * 100 = 2/3 * 100 = 67%

5.1.1 PARAMETER WEIGHTINGS ACCORDING TO VATTENFALL NORDIC IMPORTANCE

We asked a team of solution architects to assign weight to each individual function block in accordance to its importance within Vattenfall Nordic. The following table shows the importance of each individual function block and also the importance percentage for that function block:

<table>
<thead>
<tr>
<th>#</th>
<th>Parameters</th>
<th>Importance Weight</th>
<th>Importance Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Service Creation and Abstraction</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>2.</td>
<td>Runtime Service Management</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>3.</td>
<td>Service Orchestration</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>4.</td>
<td>Process Control and Optimization</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>5.</td>
<td>Presentation</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>6.</td>
<td>Identity Management</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>7.</td>
<td>Designtime Service Management</td>
<td>1</td>
<td>33%</td>
</tr>
<tr>
<td>8.</td>
<td>RAD (Rapid Application Development)</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>9.</td>
<td>Documentation</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>10.</td>
<td>Support</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>Function Block</td>
<td>Weight</td>
<td>Importance Level</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------</td>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td>11</td>
<td>Development Platform</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>12</td>
<td>Cost and licensing</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>13</td>
<td>Adaptors</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>14</td>
<td>Integration with Existing Applications</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>15</td>
<td>Open Source</td>
<td>1</td>
<td>33%</td>
</tr>
<tr>
<td>16</td>
<td>B2B transactions</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>17</td>
<td>Deploy/Un-deploy Efforts</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>18</td>
<td>Scalability</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>19</td>
<td>Standards</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>20</td>
<td>Governance Platform</td>
<td>1</td>
<td>33%</td>
</tr>
<tr>
<td>21</td>
<td>End to End Business Process Monitoring</td>
<td>2</td>
<td>67%</td>
</tr>
</tbody>
</table>

**Table 1: Function Block Importance Weightings**

**GRAPHICAL REPRESENTATION OF VATTENFALL NORDIC IMPORTANCE LEVEL**

For a more visual representation of these weightings we charted them out with the help of the following graph:

![Importance Level Graph](image)

The next section would describe the technical capabilities of the two vendors under consideration in accordance with identified function blocks. Furthermore, these vendors would be assigned a weight on basis of the capabilities they provide/satisfy.
### 5.2.1 Fulfillment of Requirements by Vendors

Following table illustrates the details of vendor capability against each function block for both JBoss and Microsoft. Microsoft provides two solutions for SOA; it can be implemented either using BizTalk Server or Windows Server AppFabric, this analysis would consider both these products for comparison.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SOA Vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red Hat JBoss</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Service Creation and Abstraction</td>
<td></td>
</tr>
<tr>
<td>SOA Connection</td>
<td>JBoss ESB and Web Services</td>
</tr>
<tr>
<td>Transformations</td>
<td>JBoss Smooks Transformer Adaptor</td>
</tr>
<tr>
<td>Routing</td>
<td>Achieved using JBoss ESB</td>
</tr>
<tr>
<td>2. Runtime Service Management</td>
<td></td>
</tr>
<tr>
<td>Service Throttling</td>
<td>Intel SOA expresses way, Service gateway (Third Party component).</td>
</tr>
<tr>
<td>Runtime Statistics and Governance</td>
<td>*JBoss SOA Governance (jBPM)</td>
</tr>
<tr>
<td>Dynamic Service Ramp-up</td>
<td>Not Supported</td>
</tr>
<tr>
<td>3. Service Orchestration</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Workflows</td>
<td>JBoss jBPM workflow and BPM (Business Process Management) engine</td>
</tr>
<tr>
<td>4. Process Control and Optimization</td>
<td></td>
</tr>
<tr>
<td>Reporting Tools</td>
<td>JBoss Developer Studio included Business Intelligence Report Tool (BIRT)</td>
</tr>
<tr>
<td>5. Presentation</td>
<td>SOA Portal Platform</td>
</tr>
<tr>
<td>Personalization</td>
<td>✔</td>
</tr>
<tr>
<td>Syndication</td>
<td>✔</td>
</tr>
<tr>
<td>Rich graphical control</td>
<td>✔</td>
</tr>
<tr>
<td>Authentication</td>
<td>An extensible, JAAS-based authentication layer that allows users to authenticate against any security provider</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Role Management &amp; Auditing</td>
<td>Permission Management - a set of built-in Seam components to allow easy management of an application's security policy.</td>
</tr>
<tr>
<td>7. Design-time Service Management</td>
<td>Service Repository/Registry</td>
</tr>
<tr>
<td></td>
<td>Life Cycle Management and Versioning</td>
</tr>
<tr>
<td>8. RAD (Rapid Application Development)</td>
<td>jBOSS Developer Studio + Seam-gen is the RAD tool packaged within jBOSS Seam.</td>
</tr>
<tr>
<td>12. Cost and licensing</td>
<td>Detailed comparison in next section</td>
</tr>
<tr>
<td>13. Adaptors</td>
<td>JBOSS ESB gateways adapters</td>
</tr>
</tbody>
</table>
SOA Governance is mentioned in chapter 3 where vendor product details are listed.

Table 2: Function Blocks and Vendor Product Capabilities

<table>
<thead>
<tr>
<th>Function Blocks</th>
<th>Vendor Product Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration with Existing Applications</td>
<td>JBoss ESB Adapters (Exposing application as Web Service)</td>
</tr>
<tr>
<td></td>
<td>WCF Custom Adapters (Exposing application as Web Service)</td>
</tr>
<tr>
<td></td>
<td>WCF Custom Adapters (Exposing application as Web Service)</td>
</tr>
<tr>
<td>Open source</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>B2B Transactions</td>
<td>SOA Smooks and UN/EDIFACT</td>
</tr>
<tr>
<td></td>
<td>BizTalk Business to Business Integration framework.</td>
</tr>
<tr>
<td></td>
<td>Not supported</td>
</tr>
<tr>
<td>Deploy/Un-deploy Efforts</td>
<td>Apache ant, Eclipse, XML configuration (Manually installations)</td>
</tr>
<tr>
<td></td>
<td>Deploy BizTalk Server as a routing and transformation hub for Web Services.</td>
</tr>
<tr>
<td></td>
<td>Services Deployed using Visual Studio or Manually in IIS</td>
</tr>
<tr>
<td>Scalability</td>
<td>JBoss Application Server</td>
</tr>
<tr>
<td></td>
<td>BizTalk Provides Horizontal and Vertical Scalability options</td>
</tr>
<tr>
<td></td>
<td>WF state persistence allows easy scalability.</td>
</tr>
<tr>
<td>Standards</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Governance Platform</td>
<td>*JBoss SOA Governance</td>
</tr>
<tr>
<td></td>
<td>Not Directly supported Achieved using Third Party Tools</td>
</tr>
<tr>
<td></td>
<td>Not Directly supported Achieved using Third Party Tools</td>
</tr>
<tr>
<td>End to End BPM</td>
<td>JBoss jBPM provides visibility into the current end-to-end state of processes in which users and applications are interacting.</td>
</tr>
<tr>
<td></td>
<td>BizTalk Server Business Activity Monitoring service</td>
</tr>
<tr>
<td></td>
<td>AppFabric Dashboard for monitoring of Workflows</td>
</tr>
</tbody>
</table>

*SOA Governance is mentioned in chapter 3 where vendor product details are listed

5.2.2 VENDOR WEIGHTING ON BASIS OF CAPABILITIES PROVIDED

After mapping vendor capabilities with the identified function blocks, weights are assigned to each vendor product according to the capabilities being fulfilled under each function block. This weighting is important as it would be used to compare the functional capabilities of each vendor product.

The comparison would be done after calculating percentage of fulfillment of each capability against each vendor.

Weighting criteria is based on the following rules

- The function blocks with no sub function blocks would be evaluated as one function block and the score it gets would be either “1” or “0” which would result in percentage as 100% or 0%.
If there is one or more than one sub function blocks then the percentage would be calculated on basis of the number of sub function blocks being satisfied. For example: as Service creation and abstraction has 3 sub function blocks, percentage would be calculated by looking at the satisfied sub function blocks and dividing them by the total sub function blocks i.e. (satisfiedFB/TotalFB) * 100.

<table>
<thead>
<tr>
<th>#</th>
<th>Criteria</th>
<th>Red Hat JBoss</th>
<th>BizTalk Server</th>
<th>Microsoft</th>
<th>Windows Server</th>
<th>AppFabric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Parameters</td>
<td>Status</td>
<td>Fulfillment %</td>
<td>Status</td>
<td>Fulfillment %</td>
<td>Status</td>
</tr>
<tr>
<td>1.</td>
<td>Service Creation and Abstraction</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
</tr>
<tr>
<td></td>
<td>SOA Connection</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Transformations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Routing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2.</td>
<td>Runtime Service Management</td>
<td>3 / 4</td>
<td>75%</td>
<td>3 / 4</td>
<td>75%</td>
<td>3 / 4</td>
</tr>
<tr>
<td></td>
<td>Service throttling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Runtime statistics and governance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Dynamic service ramp-up</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td></td>
<td>Security policy fulfillment</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3.</td>
<td>Service Orchestration</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
</tr>
<tr>
<td></td>
<td>Process control</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Workflows</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Business rules execution</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4.</td>
<td>Process Control and Optimization</td>
<td>2 / 2</td>
<td>100%</td>
<td>2 / 2</td>
<td>100%</td>
<td>1 / 2</td>
</tr>
<tr>
<td></td>
<td>Business Activity Monitoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Reporting Tools</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5.</td>
<td>Presentation</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
</tr>
<tr>
<td></td>
<td>Personalization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Syndication</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Rich graphical control</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6.</td>
<td>Identity Management</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
<td>100%</td>
<td>3 / 3</td>
</tr>
<tr>
<td></td>
<td>Authorization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Component</td>
<td>Vendor Evaluation</td>
<td>Pass</td>
<td>Fail</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------</td>
<td>-------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Role Management &amp; Auditing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Design Time Service Management</td>
<td>2/2</td>
<td>100%</td>
<td>2/2</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Repository/Registry</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Life Cycle Management</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Rapid Application Development</td>
<td>1/1</td>
<td>100%</td>
<td>1/1</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Documentation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>10. Support</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>11. Development Platform</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>12. Adaptors</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>13. Integration with Existing Applications</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>14. Open Source</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>15. B2B Transactions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>16. Scalability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>17. Deploy/Undeploy Effort</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>18. Standards</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>19. Governance Platform</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>20. End to End BPM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1/1</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Vendor evaluation of basis of capabilities provided
FUNCTION BLOCKS SATISFIED BY VENDOR

These graphs show the percentage of fulfillment of the evaluation criteria for each vendor.

RED HAT JBOSS CAPABILITIES AND FUNCTION BLOCKS

Figure 19: Red Hat JBoss Capabilities and Function Blocks

MICROSOFT BIZTALK SERVER CAPABILITIES AND FUNCTION BLOCKS

Figure 20: Microsoft BizTalk Server Capabilities and Function Blocks
COMPARISON OF VENDOR PRODUCTS ACCORDING TO CAPABILITIES

Figure 21: Microsoft Windows Server AppFabric Capabilities and Function Blocks

Figure 22: Capabilities Comparison of JBoss, BizTalk Server and Windows Server AppFabric
Cost consideration is vital for an enterprise for the selection of the right vendor. Cost comparison would also be done using two scenarios:

1) Implementation cost for building a new SOA platform
2) Implementation cost for building SOA system using already running platform

HARDWARE AND SOFTWARE SPECIFICATION USED FOR CALCULATING COST

These specifications were chosen to have an accurate approximation of each vendor product price.

<table>
<thead>
<tr>
<th>#</th>
<th>Products</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operating System</td>
<td>Price is calculated for one copy of Operating System + Support per year.</td>
</tr>
<tr>
<td>2</td>
<td>SOA Enterprise Platform</td>
<td>Application Servers + Support are sold by per processors (unlimited client access) based or by the number of client access license (CAL’s). The comparison looks at the number of processors which is 4 processors for each server for this comparison per year.</td>
</tr>
<tr>
<td>3</td>
<td>Support</td>
<td>Support price is calculated on yearly basis per year.</td>
</tr>
<tr>
<td>4</td>
<td>Data Base</td>
<td>Database Servers + Support are also sold by per processors (unlimited client access) based or by the number of client access license (CAL’s). The comparison looks at the number of processors which is 1 processor for each server for this comparison per year.</td>
</tr>
<tr>
<td>5</td>
<td>Development Platform</td>
<td>Price is calculated for one copy of Development Platform + Support per year.</td>
</tr>
<tr>
<td>6</td>
<td>Enterprise Middleware</td>
<td>It is included in the application server cost.</td>
</tr>
</tbody>
</table>

Table 4: H/W & S/W specification for price calculation

5.3.1 IMPLEMENTATION COST FOR BUILDING A NEW SOA SYSTEM

The following table shows a cost comparison for implementing service oriented architectural system from scratch using either one of the vendors. This comparison is done using List prices available from each vendor site. The specification for each product category is also mentioned:

<table>
<thead>
<tr>
<th>#</th>
<th>Products</th>
<th>Red Hat Price</th>
<th>Microsoft Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JBoss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Operating System</td>
<td>$ 2499</td>
<td>Windows Server 2008 R2 + Software Assurance $ 4101</td>
</tr>
<tr>
<td></td>
<td>Red Hat Enterprise Linux Premium Subscription (4 processors)</td>
<td></td>
<td>Windows Server 2008 R2 + Software Assurance $ 4101</td>
</tr>
<tr>
<td>2</td>
<td>SOA Enterprise Platform</td>
<td>$ 27000</td>
<td>BizTalk Server (4 processors) + ESB Toolkit + Software Assurance $ 246,132</td>
</tr>
<tr>
<td></td>
<td>JBoss SOA Enterprise Platform+JBoss Operation Network</td>
<td></td>
<td>WCF Custom Adapter Pack $ 5000</td>
</tr>
<tr>
<td>3</td>
<td>Support</td>
<td>$ 16000</td>
<td>Software Assurance is included in product price $ 0</td>
</tr>
<tr>
<td></td>
<td>JBoss Developers Subscription,(20 Suites of JDeveloper Studio)</td>
<td></td>
<td>Software Assurance is included in product price $ 0</td>
</tr>
<tr>
<td>4</td>
<td>Data Base</td>
<td>$ 4999</td>
<td>MS SQL Server $ 47817</td>
</tr>
<tr>
<td></td>
<td>My SQL Enterprise platinum Subscription</td>
<td></td>
<td>SQL Express Edition $ 0</td>
</tr>
<tr>
<td>5</td>
<td>Development Platform</td>
<td>$0</td>
<td>Visual Studio 2010 Ultimate + MSDN $ 11000</td>
</tr>
<tr>
<td></td>
<td>JDeveloper Studio (Eclipse Based).</td>
<td></td>
<td>Visual Studio 2010 $ 11000</td>
</tr>
</tbody>
</table>
Cost comparison is done for building a new system for SOA for both vendors. There would be two comparisons for Microsoft as it provides two products for the implementation of SOA: The comparison is as follows:

**COST COMPARISON FOR RED HAT JBOSS AND MICROSOFT BIZTALK SERVER**

![Cost Comparison Chart](image)

Figure 23: Cost Comparison for Red Hat JBoss and Microsoft BizTalk Server for new SOA system
This scenario is very much enterprise specific as in large enterprises a system for integration already exists and they usually have a contract of at least 3-5 years with the integration platform vendor. This thesis is exploring options for Vattenfall Nordic and they are currently Microsoft products for integration purpose. The approximated cost which they are paying for their integration solution is shown in table 6.

The comparison in this scenario would be done without considering the existing integration platform cost. The only cost considered in this scenario would be any addition cost required from the current cost for implementation of SOA.

Figure 24: Cost Comparison for Red Hat JBoss and Microsoft Windows Server AppFabric for new SOA system

5.3.2 IMPLEMENTATION COST FOR BUILDING SOA USING CURRENT SYSTEM(S)
VATTOFALL NORDIC APPROXIMATION COST FOR EXISTING INTEGRATION PLATFORM

Vattenfall Nordic is running all its systems on Microsoft products and their main integration product is BizTalk Server. The following table shows Vattenfall Nordic’s approximation integration platform cost per year:

<table>
<thead>
<tr>
<th>#</th>
<th>Products</th>
<th>Microsoft Products</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operating System</td>
<td>Windows Server 2008 R2 + Windows Server AppFabric + Software Assurance</td>
<td>$3500</td>
</tr>
<tr>
<td>2</td>
<td>SOA Enterprise Platform</td>
<td>BizTalk Server (4 processors) + ESB Toolkit + Software Assurance</td>
<td>$80,000</td>
</tr>
<tr>
<td>3</td>
<td>Support</td>
<td>Software Assurance is included in product price</td>
<td>$0</td>
</tr>
<tr>
<td>4</td>
<td>Data Base</td>
<td>MS SQL Server 2005</td>
<td>$3500</td>
</tr>
<tr>
<td>5</td>
<td>Development Platform</td>
<td>Visual Studio 2010 Ultimate + MSDN Subscription</td>
<td>$4000</td>
</tr>
<tr>
<td>6</td>
<td>Enterprise Middleware</td>
<td>N/A</td>
<td>$0</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td></td>
<td>$91,000</td>
</tr>
</tbody>
</table>

Table 6: VattenFall Nordic Current Integration Platform Cost

VATTOFALL NORDIC EXIST INTEGRATION PLATFORM COST

Vattenfall Nordic is running its current system on Microsoft technologies and the graph shows the approximate cost being paid for integration platform.

Figure 25: VattenFall Nordic's Existing System Cost
RED HAT JBOSS AND VATTENFALL NORDIC CURRENT INTEGRATION PLATFORM COST COMPARISON

This graph shows a comparison of the current integration platform cost of Vattenfall Nordic and Red Hat JBoss. This graph is important in the aspect when Vattenfall Nordic decides to fully move their integration platform to Red Hat JBoss SOA platform and they want to see a comparison/difference of their current integration system cost and the new system cost.

---

**Figure 26: VattenFall Nordic Existing System Cost and Red Hat JBoss**

SOA IMPLEMENTATION COST COMPARISON USING VATTENFALL NORDIC CURRENT INTEGRATION PLATFORM AND RED HAT JBOSS

This graph shows SOA implementation costs using the current existing system and if we move towards Red Hat JBoss. There would be no additional cost associated apart from the current cost being paid if the company uses its current integration platform for a SOA implementation, whereas if the company opts for JBoss SOA Solution there would be an overhead cost as shown in the graph.

---

**Figure 27: SOA Implementation cost using VattenFall Existing System and Red Hat JBoss**
5.4 SOA COMPETENCE IN THE ORGANIZATION

SOA is a concept which needs to be understood to a level to fully implement and utilize its potential. Therefore, SOA competence within the enterprise becomes one of deciding factor for moving towards SOA implementation. This also in turn effects the vendor selection, as people would be much more competent in the currently running integration platform vendor products.

Vattenfall Nordic is running mostly Microsoft products and employees are much more acclimatized with Microsoft products rather than Red Hat’s Enterprise Middleware suite. The following table shows the competence level of employees within Vattenfall Nordic with respect to different Vendors:

<table>
<thead>
<tr>
<th>#</th>
<th>Tools</th>
<th>JBoss</th>
<th>VF competence</th>
<th>Microsoft</th>
<th>VF competence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Development Platform</td>
<td>Eclipse Based</td>
<td>3</td>
<td>Visual Studio, Team Studio Foundation</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Data Base</td>
<td>MySQL</td>
<td>0</td>
<td>SQL Server (Dev &amp; Management)</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>SOA Tools</td>
<td>JBoss SOA Platform</td>
<td>0</td>
<td>BizTalk Server</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Operating System</td>
<td>Red Hat Linux</td>
<td>0</td>
<td>Windows 2008 R2 + VS</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 7: Comparison of SOA competence within Vattenfall Nordic

SOA COMPETENCE COMPARISON FOR MICROSOFT AND RED HAT PLATFORMS

Figure 28: SOA Competence of employees within Vattenfall Nordic
5.5 SUMMARY OF RECOMMENDATIONS

There are different factors that influence the selection of vendor for SOA Implementation by an enterprise. These can vary from organization to organization, depending on their requirements and constraints. Based on the above analysis the following three factors are identified which could help different organizations based on their requirements to reach a decision regarding SOA vendor selection:

1) Technological Capabilities
2) Cost Effectiveness
3) SOA Competence

5.5.1 TECHNOLOGICAL CAPABILITIES

Some organizations require an integration platform which fulfills their requirements irrespective of cost. The following graph shows an average fulfillment of the identified function blocks by different vendor products:

![Function Blocks Graph]

**Figure 29: Percentage of Function Blocks Satisfied by Vendor Products**

RECOMMENDATIONS

From a technological perspective all the products are equally good in providing SOA Implementation, the following are some recommendations:

1. **Open Source**: Open source has not been considered as a function block in calculating technological capabilities, if it gets added Red Hat JBoss takes the lead. The recommendation here is if a company is looking for Open Source vendor product for the implementation of SOA, Red Hat JBoss would be the ideal choice.

2. **Rapid Development**: Microsoft has always been ahead in providing easy to configure and user friendly products. If the company is not inclined towards open source or cost factors and wants a quick SOA solution, Microsoft’s products would be best for such kind of development and implementation of SOA.
5.5.2 COST EFFECTIVENESS

Implementation cost is a big factor which every enterprise looks at before deciding which vendor to choose. The following graph shows a cost comparison of different vendor products for SOA implementation:

![SOA Product Cost Graph]

**Figure 30: SOA Implementation Cost by different Vendor Products**

RECOMMENDATIONS

The cost difference between open source and proprietary products can be clearly seen in Figure 30 and this difference becomes a major deciding factor for an enterprise, the following are some recommendations which can be helpful for an organization for choosing cost effective vendor:

1. **Low Cost More Features**: If an organization is looking for a SOA solution which is low in cost and supports adequate features for SOA implementation. The best choice would be Red Hat JBoss Middleware platform, which provides all the technological capabilities required (can be seen in technological capabilities section).

2. **Low Cost Less Features**: If the organization wants low cost with proprietary products for the implementation of SOA and can compromise on some capabilities (e.g. B2B Integration, No Reporting feature), then the viable option would be to go for Microsoft Windows Server AppFabric. This product is free of cost and it gets integrated with Internet Information Server (IIS) 7.0 running on any Windows 2008 Server or Windows 7.

3. **Existing Platform Utilization**: If an organization is running its integration platform on Microsoft products especially Microsoft BizTalk Server and would continue to do so in the future. The best option for this scenario is to just upgrade the current running system to start supporting a SOA. Microsoft provides free of cost products (ESB Toolkit, WCF Custom Adapters) which can upgrade BizTalk to start functioning in accordance with SOA principles.
5.5.3 SOA COMPETENCE

To implement SOA according to its design principles, it is important that people who are implementing SOA should have conceptual as well as product specific implementation knowledge. This is also important consideration for an organization with huge integration platform. The following graph shows the SOA competence level in VattenFall Nordic with respect to different vendors:

![SOA Competence graph](image)

**Figure 31: SOA Competence in Vattenfall Nordic**

**RECOMMENDATIONS**

SOA and Product competence is important as enterprises having huge integration platforms cannot just switch to another vendor for implementation of any new system or concept. The following are some recommendations:

1. **SOA Competence**: Employees should be provided with proper training on the concept of SOA irrespective of the technology used to implement it. Until there is not a mind set of SOA, it would be difficult for people to understand and implement SOA to its full potential.

2. **Product Competence**: Product competence is also important, and organizations should consider the product competence level of their employees before moving towards SOA. For the above comparison Vattenfall Nordic has mostly competence in Microsoft products, and for a fast implementation of SOA, Microsoft would be the best choice. Moreover, if they move towards implementing SOA using Red Hat JBoss, they would need to hire some new resources who are expert in managing Red Hat JBoss SOA platform.
Chapter 6
ROADMAP AND CONCLUSION

6.1 VENDOR PRODUCTS ROADMAP

Vendor Roadmap is also important factor for an enterprise, as they need long term support and product upgrades for the integration platform they choose for the implementation of Service Oriented Architecture.

Microsoft and Red Hat both are very aggressive in making their products as much SOA compliant as possible. Both the vendors have started supporting cloud platform which is the next generation for service oriented architecture. Microsoft launched its Windows Azure cloud platform in start of 2010 and Red Hat announced its PaaS (Platform as a Service) in August 2010. The table below shows some glimpse of functionality which can be expected from both vendor products.

<table>
<thead>
<tr>
<th>Area</th>
<th>Microsoft</th>
<th>Red Hat</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enterprise Connectivity</strong></td>
<td>● Microsoft AppFabric integration with Business to Business applications using LOB adapters.</td>
<td>● Apache CXF and Apache Camel Integration for connectivity.</td>
</tr>
<tr>
<td><strong>Application Platform Alignment</strong></td>
<td>● Enhanced development, scalability, deployment and management capabilities to be integrated into AppFabric. ● AppFabric and BizTalk integration</td>
<td>● Enhanced UDDI v3 registry ● BPEL Support ● HornetQ support ● SOA Repository ● ESB 5</td>
</tr>
<tr>
<td><strong>Cloud Support</strong></td>
<td>● Direct application deployment from development platform to cloud ● Integration of private clouds with Windows Azure Platform</td>
<td>● Support for deployment across private and public clouds. ● Extended support for programming models, languages, frameworks and APIs.</td>
</tr>
</tbody>
</table>

Table 8: Roadmap for Vendors Products

6.2 CONCLUSION

SOA has evolved at a tremendous rate in the last couple of years. It was originally a technical concept but because of its business integration nature made its way towards enterprises. Businesses gain from SOA implementation, as services in SOA not only help in resolving technical interoperability, but also help in solving specialized business functionality. Services can also be combined to act as a business process for the fulfillment of business activities. From an implementation perspective, Web service technology is becoming the standard implementation technique for SOA solutions and most of the enterprises are building their SOA computing environments using web services.

SOA implementation in enterprises makes it easy to integrate applications and enables building of sophisticated applications which fulfill business and enterprise needs. To make a choice for the right vendor to implement SOA is a daunting task. Every enterprise has its own specific requirement and constraints. These could be financial constraints or business constraints. It is difficult to pin point a specific criteria for the selection of SOA implementation within an enterprise. This thesis tries to generalize a list of technical and non technical functional blocks that influence implementation of SOA and Vendor selection.

There are a number of tools that provide specialized functionalities for construction of SOA based systems. Both proprietary and open source vendors are providing product suites which include integration platforms,
management and monitoring suites, collection of service design techniques and creation and modeling tools for SOA.

SOA is the next big thing for future integration of enterprises. It is constantly evolving as it gets adopted more and more by enterprises and organizations. In future, SOA would have a significant impact on the development of enterprise application integration infrastructure as more and more technologies and vendors start providing solutions.
REFERENCES


