

UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Mariana Kukhtyn

Refactoring of ESB-based systems on the clouds

Master's thesis (30 ECTS)

Supervisor: Satish Srirama, Ph.D.

Author: "....." June 2011

Supervisor: "....." June 2011

Professor: "....." June 2011

TARTU 2011

‘The one thing that matters is the effort. It continues, whereas the end to be attained is but an illusion of the climber, as he fares on and on from crest to crest; and once the goal is reached it has no meaning.’

Antoine de Saint-Exupry, *The Wisdom of the Sands*

UNIVERSITY OF TARTU

Abstract

Faculty of Mathematics and Computer Science
Institute of Computer Science

Master of Science

Refactoring of ESB-based systems on the cloud

by Mariana Kukhtyn

Nowadays enterprises are facing new, unprecedented press for competitiveness: the speed of service providing, price of inaccuracy, possibility to use applications from desktops, notes and other mobile devices. These new conditions force business to search for additional resources and configurations for providing better services. Thus, we are arguing to find general hints for uniting enterprises, cloud computing and distributed middleware software in one system in order to fulfill this goal.

ESB-based systems and Service Oriented Architecture taken as one of its cases are assessed in concern to their possible application and profit-usage on the remote servers. In this work, for the first time different vision of distributed ESB-systems (Federated, Internet and proper Distributed) are described, proving that idea is not new but different research groups focus on different features. Statistics data on cloud application is used to object widely-spread statements on cloud security and finance-efficiency. Moreover, skeptical attitude is addressed by reference to Hype cycles of technology, which indicate the possible blossom of technologies in nearest future even though at the current moment these technologies are not mature.

Also simple guide aiming to help in alignment ESB-based system integration and cloud computing usage is presented. The thesis work is concluded with comparison of ideas and results, experience description and description of outcomes.

Acknowledgements

I want to thank my parents and my whole family for their support and faith in me. Also I would like to thank my thesis advisor Satish Srirama for making me feel trusted and free to learn on my own.

Contents

Abstract	iii
Acknowledgements	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Goals	2
1.2 Outline	3
2 Theoretical overview of ESB-based systems and their usage on the cloud	4
2.1 What is ESB?	4
2.1.1 Explication	4
2.1.2 Prerequisites for ESB usage	7
2.2 ESB modifications	8
2.2.1 Internet Service Bus	8
2.2.2 Distributed Service Bus	9
2.2.3 Federated Service Bus	10
2.3 Cloud aspects	11
2.3.1 Choosing cloud	13
2.3.2 Security	15
2.3.3 The ROI of the Cloud	16
2.3.4 Prerequisites for Cloud Computing	18
3 Practical utility of ESB-based systems on the cloud.	20
3.1 Oracle SOA Suite features	20
3.1.1 Oracle Enterprise Service Bus vs. Oracle Service Bus	23
3.1.2 Oracle SOA on Amazon cloud	24
3.2 Oracle implementation in companies	24
4 Testing ESB-based system on the cloud	27
4.1 Test case description	29
4.2 Test environment requisites	31

5	Results and analysis of tests	33
5.1	Refactoring idea	33
5.2	Experience conclusion for ESBs migrating to the cloud	37
6	Conclusion	39
7	Sisukokkuvte	41
A	Loan application request to Normal Loan Processor	42
B	Response From Normal Loan Processor	43
	Bibliography	44

List of Figures

2.1	Enterprise Service Bus	6
2.2	An Internet Service Bus	8
2.3	Distributed Service Bus	9
2.4	ESB deployment patterns	10
2.5	Hype Cycle by Gartner	11
2.6	Hype Cycle by Gartner. Refined	12
2.7	Hype Cycle by Gartner, August 2010	13
2.8	Main reasons for using clouds	14
2.9	Preferences on deployment models	15
2.10	ROI expectations of Cloud Computing	17
3.1	An Architectural Perspective of a SOA Platform	21
3.2	History of the Oracle SOA Platform	22
3.3	Oracle SOA Suite 11g Mediator vs Oracle Service Bus	23
4.1	Weblogic Domain	28
4.2	Loan Application Request Web Service via Oracle Service Bus	30
4.3	Test case architecture	31
5.1	First configuration of test case	34
5.2	Second configuration of test case	34
5.3	Proxy service statistics	35
5.4	Normal Loan service statistics	36

List of Tables

3.1	Mediator vs. Oracle Service Bus comparison	23
3.2	SOA Design Patterns in the Cloud	25
4.1	Machine specification	32
5.1	First configuration of test services on the cloud	36
5.2	Second configuration of test services on the cloud	37

Chapter 1

Introduction

Enterprise Service Bus (ESB) and cloud computing are two notions that are united by the lack of clear and concise definition. But not only is this fact common for both of them. ESB and cloud computing entered CIOs(Chief information officers) lives last decade bringing confusion, interest and overexcitement. Few of companies ventured to try ESB and cloud computing in enterprise systems but these concepts required solid reshuffling of the whole system what became a reason for others to wait.

Till now commentators cannot agree whether to perceive ESB as an architectural style, a software product, or a group of software products. [21] Community gets even more puzzled with middleware vendors who re-badged their products as ESB. There are BizTalk Server from Microsoft, Fuse ESB (Enterprise Service Mix), JBoss Enterprise Service bus, Mule ESB, Open ESB, PEtALS ESB, WSO2 Enterprise Service Bus, Oracle Enterprise Service bus and Oracle Service bus by Oracle and bunch of other Messaging Middleware on the market. Meanwhile there is no global standard for ESB, some expert prune down the whole concept - “Indeed, it’s often said that if you are using enterprise messaging products, you have an ESB.” [20].

At its simplest an ESB is the mechanism for transportation messages between services in complex architectures, which offers additional facilities as message queuing capability, message routing and transformation, mediation capabilities, support for WS- protocols and support for monitoring and logging message activity.

In a homogeneous environment it is in fact unnecessary to have a service bus at all. Routing between services and requesters may be achieved on a point to point basis. Support for security and reliability may also be implemented on a point-to-point basis.

Though for more comprehensive environments one can tell that ESB is an essential part of SOA architecture and as more services will participate in business choreography the bigger will be the need for centralised coordination of multiple implementation services.

The benefit of an ESB is that it eases the process of creating an SOA. Within the boundaries of an ESB, support for multiple protocols and data transformation enables heterogeneous services to behave as if they were homogeneous.[28] Adapters allow to expose legacy systems as services without programming. The support for reliable and secure messaging and queuing is also available through straight-forward configuration rather than coding. Take into account the availability of logging and access control for governance and ESB can be a very useful tool indeed.

Cloud computing is the notion to describe provisioning of computational resources on demand via a computer network. The main characteristics of cloud computing are scalability and pay-as-you-go payment model. Combining different deployments models of cloud computing and implementing ESB requisite functionality allow enterprises to rethink its IT structure thus to become more mobile, scalable, agile and cost-efficient.

In this thesis I will try to describe the present state of two architectures – Enterprise Service Bus and Cloud computing, their intersection presented as distributed enterprise system (ESB distributed on two clouds – private and(or) public), outcomes of this merging, advantages and drawbacks, come out with theoretical recommendations for enterprise to go on with these two architectures and test example to reveal message and performance issues.

1.1 Goals

At the start of the work the next goals were set:

- Reveal the necessary and sufficient condition for the enterprise systems to become ESB-based systems
- Reveal the necessary and sufficient condition for the enterprise systems to move to the cloud
- Observe performance of ESB-based system on the cloud
- Show efficient refactoring of the enterprise system based on the performance and communication activities between separate services
- Give practical recommendations for the best software distribution.

1.2 Outline

Throughout the work we will go through detailed definition of ESB, its modifications, cloud computing evolution, security issues concern to cloud computing, practical aspects of using distributed enterprise solutions and we will look to practical part and possible issues on that end.

- Chapter 2 shows theoretical overview of ESB-based systems, its usage, modifications and enterprise usage of the cloud.
- Chapter 3 discusses the main approaches for modeling and usage of ESB-based system on example of Oracle SOA Suite which include Oracle Enterprise Service Bus (Mediator) and can be deployed with Oracle Service Bus and Amazon EC2 possibilities for ESB deployment.
- Chapter 4 describes test case specification and test environment configuration.
- Chapter 5 outlines refactoring idea and experience description.

Chapter 2

Theoretical overview of ESB-based systems and their usage on the cloud

2.1 What is ESB?

Nowadays enterprises are facing new, unprecedented press for competitiveness: the speed of service providing, price of inaccuracy, possibility to use applications from desktops, notes and other mobile devices. These new conditions force business to search for additional resources and configurations for providing better services. Software discussed in this work can be used to meet actual demands, though not all enterprises will find using this software suitable. Thus, we are arguing to find general hints for uniting enterprises, cloud computing and distributed middleware systems in one system.

ESB-based system and taken as one of its cases Service Oriented Architecture are assessed in concern to their possible application and profit-usage on the remote servers.

2.1.1 Explication

Service Oriented Architecture (SOA) notion became very popular lately. It is architecture for distributed systems which are composed of interoperable services. The main characteristic of use of services is that it ignores the details of actual implementation platform, technologies, programming languages, allowing interoperability across platforms and organizational units. Service Oriented Architecture also can be defined in terms of business (set of services that a business wants to expose to their customers

or other organizations) and technical perspective (an architectural style which requires a service provider, requestor and a service description). Main SOA characteristics and principles are:

- Loose-coupling
- Location transparency
- Protocol independence.

Service oriented architecture may be realized with different technologies. At first CORBA was used, but currently interest moved to Web Services standards, which are XML-based languages for accessing and describing services. Service Oriented Architecture (SOA) enables flexible integration of applications and resources by:

- representing every application or resource as a service with a standardized interface
- enabling the services to exchange structured information (messages, documents, business objects)
- coordinating and mediating between the services to ensure they can be invoked, used and changed effectively.

When number of services, which have to be interconnected, raised dramatically, the need for routing of service requests, transformation of data protocols and formats, service and business process choreography execution resulted into emergence of middleware for these purposes. Eventually, messaging middleware extended its possibilities and Enterprise Service Bus (ESB) was introduced.

ESBs have been instrumental in the evolution of integrated middleware infrastructure technology by combining features from previous technologies with new services, such as message validation, transformation, content-based routing, security and load balancing. [27]

The ESB technology is grown out of the enterprise application integration (EAI) and the Web services infrastructure:

- EAI technology : Integration based on message- oriented middleware (MOM) was adding Web Services support in response to SOA opportunities
- Web services technology: Segmented WS infrastructure providing security, management, registries and more.[29]

In its simplest form, an ESB is really just the architectural layer that connects a bunch of software capabilities (referred to as services) to high level controlling entities (referred to as choreographers) so the capabilities can work in concert to accomplish tasks.

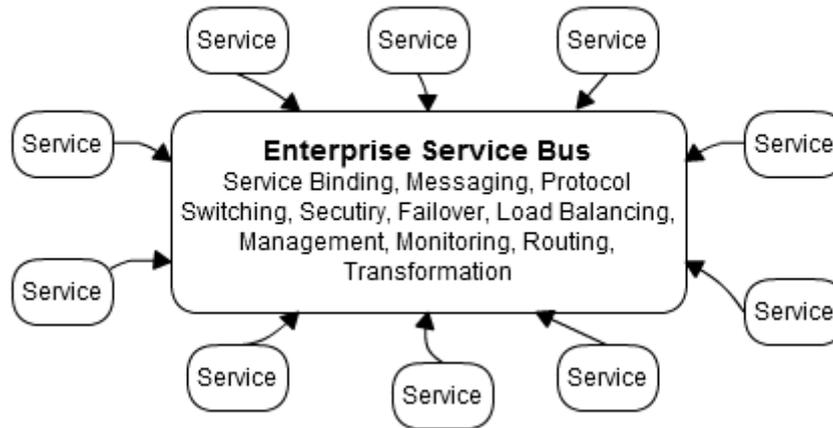


FIGURE 2.1: Enterprise Service Bus

ESB incorporates the features required to implement complex service-oriented architectures transformation and mapping. It can meet challenges of integrating applications and provide a single, unified architecture that can:

- Distribute information across an enterprise quickly and easily.
- Mask differences among underlying platforms, software architectures, and network protocols.
- Ensure information delivery even when some systems or networks may go off-line from time to time.
- Re-route, log, and enrich information without requiring applications to be rewritten.
- Provide incremental solution implementations so all enterprise services and applications need not change immediately or all at once.[19]

The main advantage of ESB is possibility to connect different application and to perform data exchange between them either within enterprise Intranet or across the Internet, or even combining these two options.

Properties of ESB:

- light weight
- loosely coupled
- event-driven
- transactional
- abstract endpoints
- intelligent routing
- message transformation(inbound/outbound)
- reliable messaging
- multi-protocol messaging.

2.1.2 Prerequisites for ESB usage

The key to ESB's appeal, and possibly also its future success, lies in its ability to support incremental service and application integration as driven by business requirements, not as governed by available technology. According to some sources, "ESB is not a new software product - it's a new way of looking at how to integrate applications, coordinate resources, and manipulate information." [19] Considering Enterprise Service Bus is worth when there are next conditions [30]:

- There are 3 or more application to be integrated;
- These services use more than one messaging protocol to communicate;
- In future there should be plugged-in more applications/services;
- There is need in message routing capabilities;
- Scalability is critical;
- The applications are loose-coupled, scalable and require robustness.

It is important to keep in mind that there is no one recipe for fit-all solution and one need to know architecture substantially before making a technology decision.

2.2 ESB modifications

As notion of ESB is closely connected to Web Services next evolutionary step after its rise was to deliver this entire ESB-based infrastructure to the Internet. It hasn't evolved into one clear definition but different sources name it as Internet Service Bus or Distributed Service Bus or Federated Service Bus. We should underline that all aforementioned Buses embody ESB facilities but in more detailed manner. In this section we will just introduce the different sights to ESB.

2.2.1 Internet Service Bus

The core Internet Service Bus concept is built around a Uniform Resource Identifier (URI) space. One application registers and owns some particular URI. URIs below this root represent application integration points, and are similar to destinations in the Java Messaging Service(JMS), queues in message-oriented middleware, or topics in publish/subscribe systems. Further development of ISB application is performed by associating policy and function with URIs. The composite application is a set of URIs, policies and functions. The ISB provides an identity and access function for controlling which messages can be sent to a URI, and by whom.[22] The identity and access function is an example of associating a policy with a URI.

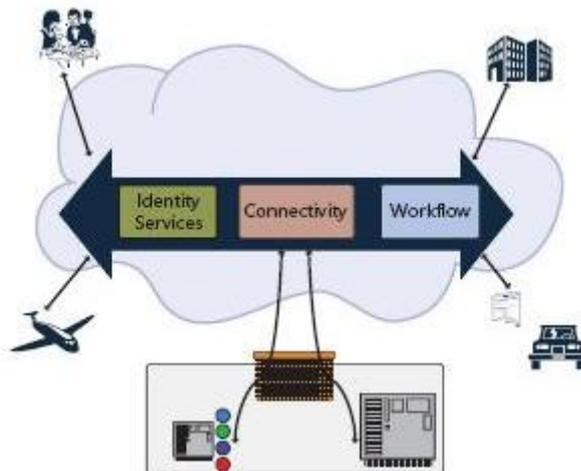


FIGURE 2.2: An Internet Service Bus [22]

Specifically the ISBs connectivity component offers three core functions:

1. Relay enables communication between the ISB and applications behind firewalls. The relay function eliminates the need to set up systematic cross-enterprise connections for simple scenarios.

2. Protocol provides a set of common protocols for exchanging messages, such as WS-* or REST. The ISB also provides protocol mapping for automatically connecting endpoints using different protocols.
3. Functions provide support for associating simple ESB-like functions to the URL. Examples could include multicast, WS-Eventing, persistent messaging, etc. [22]

2.2.2 Distributed Service Bus

Distributed Service Bus was presented in global service delivery platform (GSDP), an open platform through which domain independent services can be used to build problem-specific service solutions. Then this (named SOA4All) platform focuses on automating the management of services that are traditionally driven by technologies such as WSDL and SOAP, and on empowering RESTful services.[31] Automation is advocated through the application of semantics and hence by means of Semantic Web services. In other words, services are annotated by means of Semantic Web languages, and the platform services operate on the semantic descriptions of services and processes, rather than the actual software implementation or the physical endpoints.

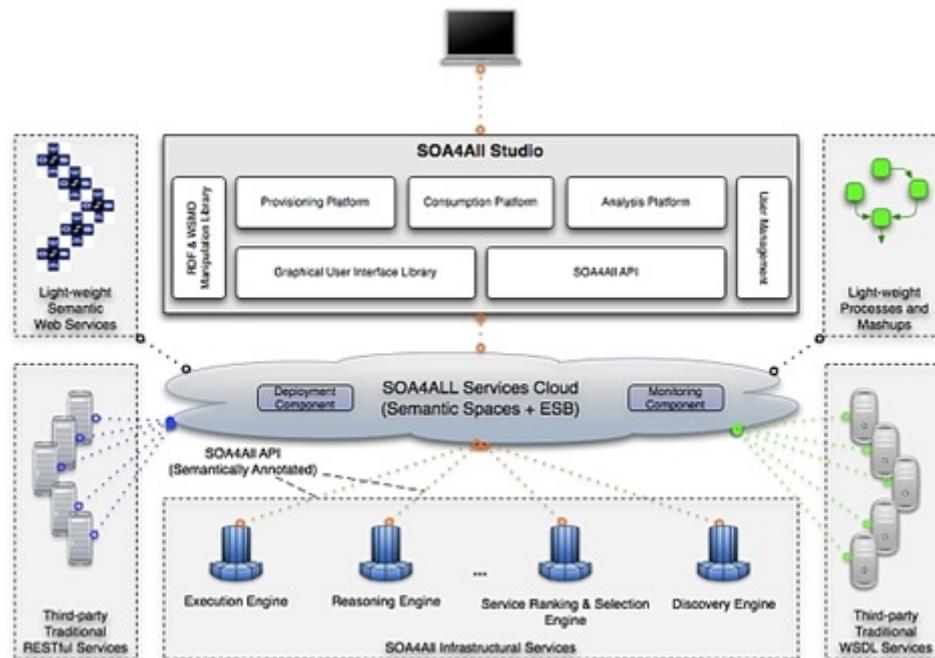


FIGURE 2.3: Distributed Service Bus [31]

In fact, the services turn into utilities, which disappear in the Web that becomes the platform and a public, open and distributed alternative to private legacy systems. The service capabilities and the offered quality of service become the decisive characteristics, rather than the endpoint location or provider.

The Distributed Service Bus enables Web-style communication and collaboration via semantic spaces and service bus technology, and yields the core runtime infrastructure. The DSB augments enterprise service bus technology with distributed service registries, the layering of the service bus on top of established Internet-enabled middleware, and enhancement of the communication and coordination protocols by means of semantic spaces.[31]

2.2.3 Federated Service Bus

By Federated ESB I mean more complicated structure of IT achitecture which doesn't stop on single ESB for the whole orranization. It can be several interconnected ESBs, each of them deployed in some specific organizational unit or department.

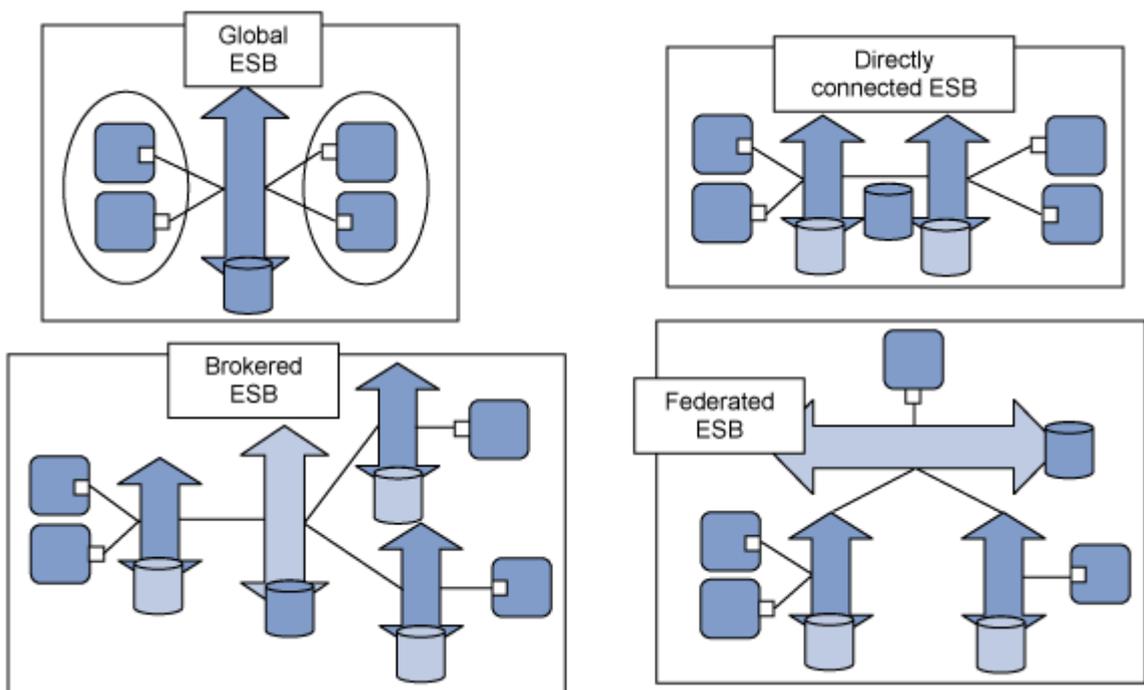


FIGURE 2.4: ESB deployment patterns [12]

There are several subtypes one can distinguish [12]:

- Directly connected ESB. A common service registry makes all of the services in several independent ESB installations visible. Used where services are provided and managed by a line of business but made available enterprise-wide.
- Brokered ESB. Bridge services that selectively expose requesters or providers to partners in other domains regulate sharing among multiple ESB installations that each manages its own namespace. Service interactions between ESBs are facilitated through a common broker that implements the bridge services. Used by

departments that develop and manage their own services, but share a few of them, or selectively access services provided across the enterprise.

- **Federated ESB.** One master ESB to which several dependent ESBs are federated. Service consumers and providers connect to the master or to a dependent ESB to access services throughout the network. Used by organizations that want to federate a set of moderately autonomous departments under the umbrella of a supervising department.

2.3 Cloud aspects

Industrial usage of cloud computing not only for storage is still doubted by some people. For this reason I decided to include next chapter to my work and to discern closely actual progress and market position of cloud computing on Hype Cycle by Gartner.

A hype cycle is graphical representation of the maturity, adoption and social application of specific technologies.[32] It was first introduced in 1995 for depicting the over-enthusiasm and subsequent disappointment that typically happens with the introduction of the new technologies [34] and their further practical application and acceptance.



FIGURE 2.5: Hype Cycle by Gartner [32]

It has five phases which can be describe next:

1. **Technology trigger** – new technology is introduced or product is launched or other event that generates significant interest and buzz takes place.
2. **Peak of Inflated Expectations** – the visibility of technology increases as the buzz increases, surrounding over-enthusiasm over the technology leads to unrealistic expectations. Over-enthusiasm is caused by some successful applications of the technology, but as this technology is not runned-in there are typically more failures.

3. **Thought of Disillusionment** – technologies fail to meet expectations and quickly become unfashionable. The press loses its interest to topic and technology.
4. **Slope of Enlightenment** – though some companies persevere and continue experiments to understand the benefits and practical uses of the technology.
5. **Plateau of Productivity** – benefits of technology are widely demonstrated and accepted. Technology becomes mature and stable and evolves in second and third generations.

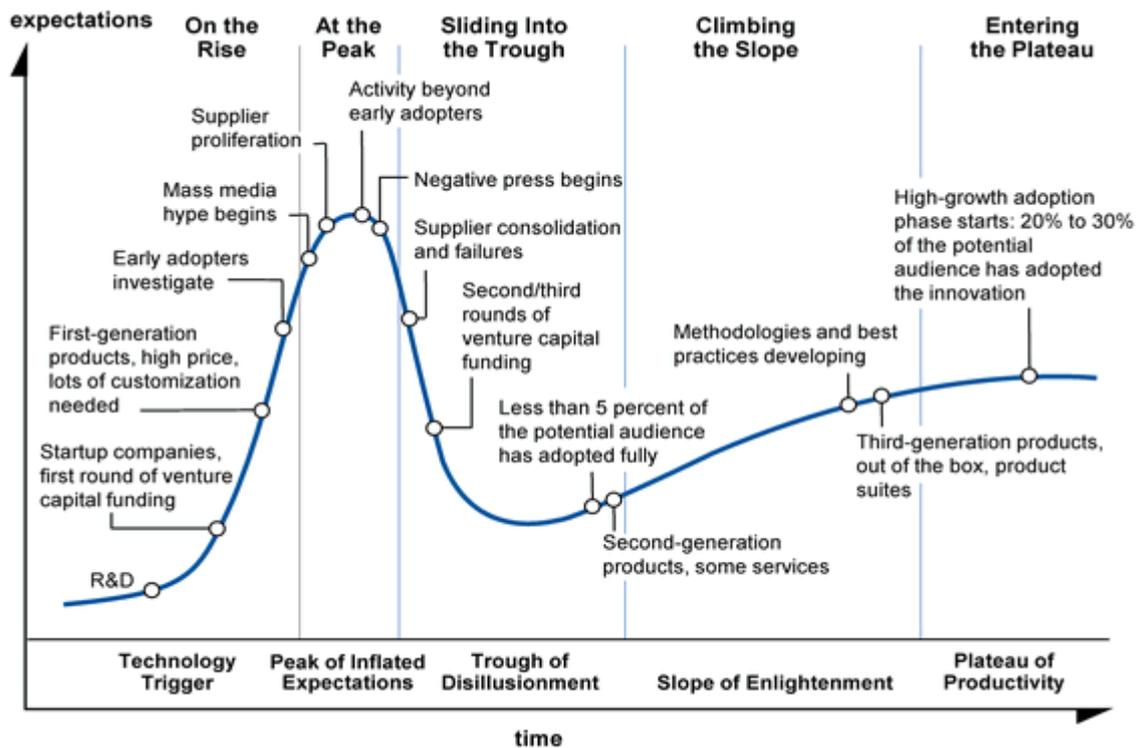


FIGURE 2.6: Hype Cycle by Gartner. Refined [32]

It should be mentioned that some products/technologies fail on the Thought of Disillusionment stage and never reach next stages.

I will use this hype cycle to analyze maturity of product/technologies that will be considered in this work. The latest Hype cycle by Gartner was presented in August 2010 (Figure 2.7) and in this version (as distinguished from previous ones) the cloud refers to three entries – Cloud Computing, Cloud/Web Platforms and Private Cloud Computing. All three categories are projected to be 2-5 years from mainstream adoption. In 2010, Cloud Computing and Cloud/Web Platforms have made it over the peak with Cloud/Web Platforms down the slope and are due to negative press, supplier consolidation and failures. Meanwhile, Private Cloud Computing is making its way to the peak facing suppliers proliferation.[33]

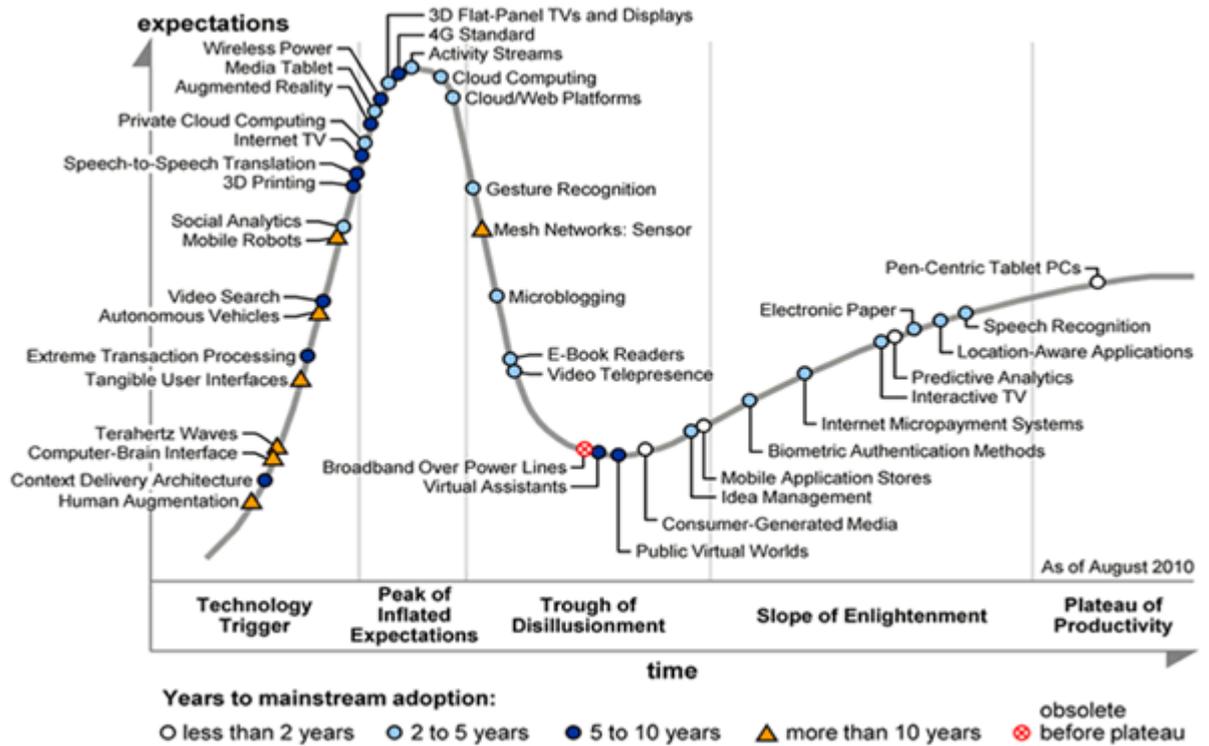


FIGURE 2.7: Hype Cycle by Gartner, August 2010 [33]

ESB was last presented in Gartner Hype Cycle for 2007 year with 5 to 10 years of mainstream adoption. As we see for both technologies the Thought of Disillusionment is upcoming and of course it is a question if these technologies will continue successfully with Slope of Enlightenment or they will mutate into something new. Though we can't see the possibility for these technologies to disappear, we can say that the hardest times are ahead and of course both of them need some good stabilizing stage in order to move towards mass adoption but nevertheless this fact can't erase significance of demonstrated IT development so far.

2.3.1 Choosing cloud

Next I will consider some practical aspects of cloud computing usage on real-life examples. The most active adaptor of cloud computing was state sector in United States of America. In fact, the Office of Management and Budget in December 2010 declared that government now operates under a cloud-first policy, meaning that agencies must first try to incorporate some type of cloud computing into projects. And if they choose not to use a cloud scenario, they must justify their decision. This means that government encourage solutions where a secure, reliable and cost-effective cloud options exist.

In December 2010 Government Information Group presented survey among state agencies and it will be used for analyzing preferable behaviour towards cloud usage when the fact of its usage is predetermined. Survey was conducted among 460 respondents, roughly half of them work for civilian agency and rest for military agencies.

According to a survey by the 1105 Government Information Group, cost reduction, fast access to data and applications, and simplifying IT infrastructure and management are the top three reasons why federal agencies are moving to the cloud. More reasons are presented in Figure 2.8.

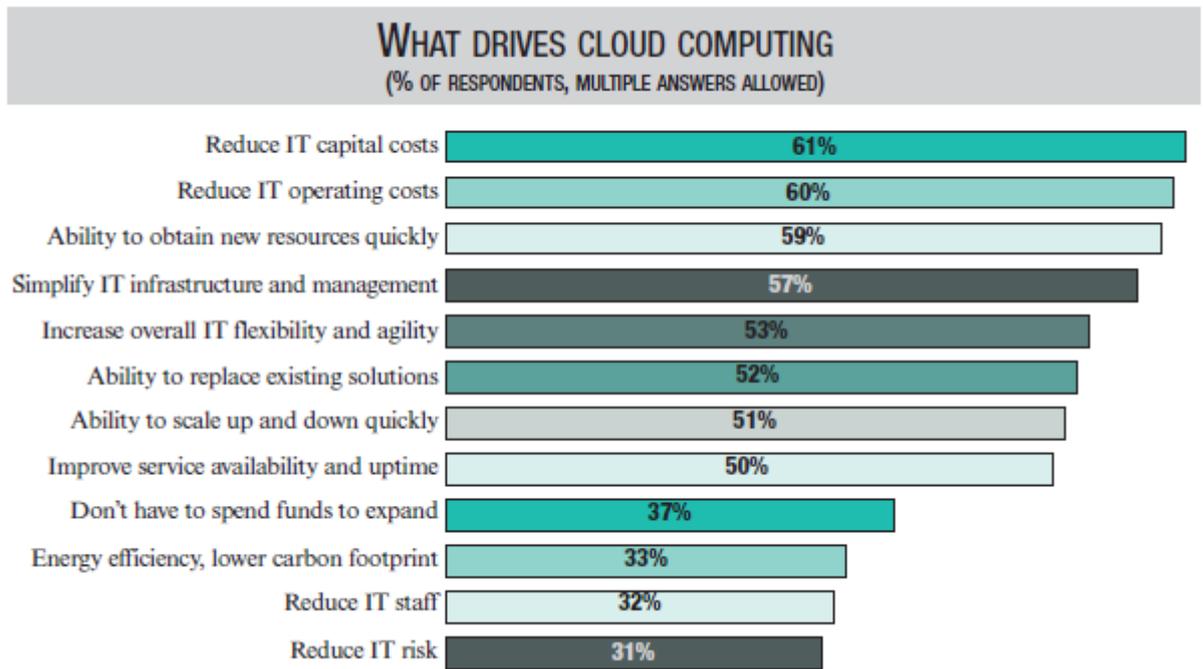


FIGURE 2.8: Main reasons for using clouds [35]

Agencies were free to choose type of cloud to use. Majority of respondents prefer to tight to Private cloud. Among examples of private clouds are research use of servers on demand and testing and developing network solutions for command, control, intelligence, and surveillance and reconnaissance projects.

Overall public cloud got the least percentage of use, which was about 12-15%. Hybrid and community clouds were a bit more popular in range 16-23%. And the most attractive variant was private clouds - they calm down security cowards but still let benefit from cloud advantages. The full information of deployment models is presented in Figure 2.9.

Private clouds are operated solely for one organization, can be managed either by the organization itself or by a third party. They are used for sensitive or mission-critical information transfer.

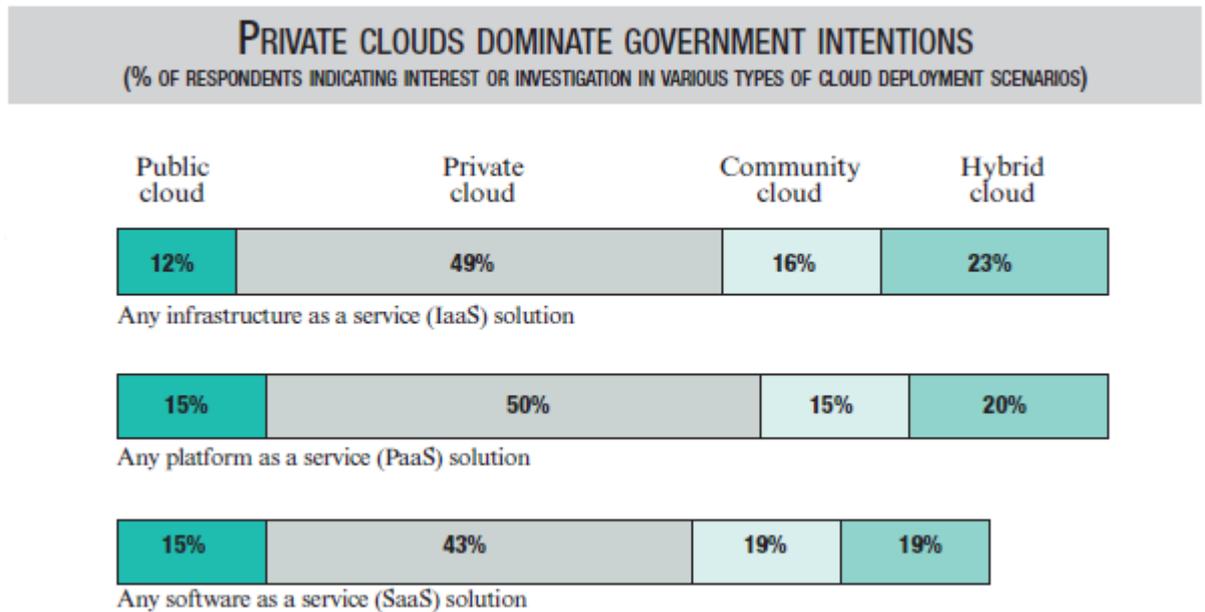


FIGURE 2.9: Preferences on deployment models [35]

Public clouds are more secure than accessing information via the Internet and tend to cost less than private clouds because services are more commoditized. Most popular functions for public clouds are:

- Collaboration;
- Social networks;
- CRM;
- Storage.

Hybrid cloud is a combination of both public and private clouds. In this variant the private cloud can contain sensitive information and applications and the public cloud can contain less sensitive information and platforms.

Community cloud is private cloud, shared by several organizations - e.g. community dedicated to compliance considerations or a community focused on security requirements policy.

2.3.2 Security

Security issues are among top-three issues discussed in parallel with cloud computing. The critical approach is provided by the fact that a lot of applications dispose clients

information and in terms of state agencies this information can be extremely secret and confident. Though there is no one common attitude, consumers remain really sceptical on data protection in the clouds. According to survey by the 1105 Government Information Group, the most critical cloud computing security worries are potential data loss or leakage, robust identity authentication and credential management, and secure and timely identity provisioning. Other concerns include effective data encryption; physical security; insecure application programming interfaces; and account, service and traffic hijacking.

In general, survey showed that only small part (about 15% of respondents) is familiar with any of major organizations focused on cloud security. Respondents that fully adopted at least one application on the cloud were more aware of the various security initiatives. So fear is provoked by ignorance.

Security on the cloud continues improving due to evolution of the technology and education of customers. But next recommendations for the cloud applications will allow more confidence about security. Recommendations are

- defining and enforcing strong password policies
- considering federated authentication to delegate authentication to the organization using the cloud service
- implementing user-centric authentication (systems where users, rather than service providers, control their identity credentials). [35]

2.3.3 The ROI of the Cloud

Important factor of moving to the cloud is undoubtedly economic feasibility. While the Brookings Institution estimates that federal agencies are experiencing up to a 50% savings overall by moving to the cloud, in fact, some types of federal cloud deployments save more, while others save somewhat less.

Primary areas for savings are:

- Direct labour. Automated provisioning significantly reduce IT management costs. Automated upgrading altered procedure for maintaining servers and troubleshooting save time for administration needs resulting in reduced needs in administrators and economy on stuff salary. IBM Research found that 81% of public cloud infrastructure adoption payback is due to decreased labour.

- **Hardware.** On-site hardware has to be replaced less often and less new hardware has to be purchased. That, in turn, leads to much-reduced power and cooling costs, as well as less space needed in the data centre. IBM estimates that medium-sized cloud deployments lead to a 62% savings in a year compared with an on-premise system. For larger cloud deployments, the annual savings approaches 50% compared with an on-premise implementation.
- **Software.** Avoiding shelfware (software which is never used and so ends up on the shelf), cheaper software costs due to data centres discounts.
- **End-user productivity.** The most difficult and doubtful area of savings. Depends a lot on cloud provider. Might be received for example by obtaining services directly from the cloud providers.

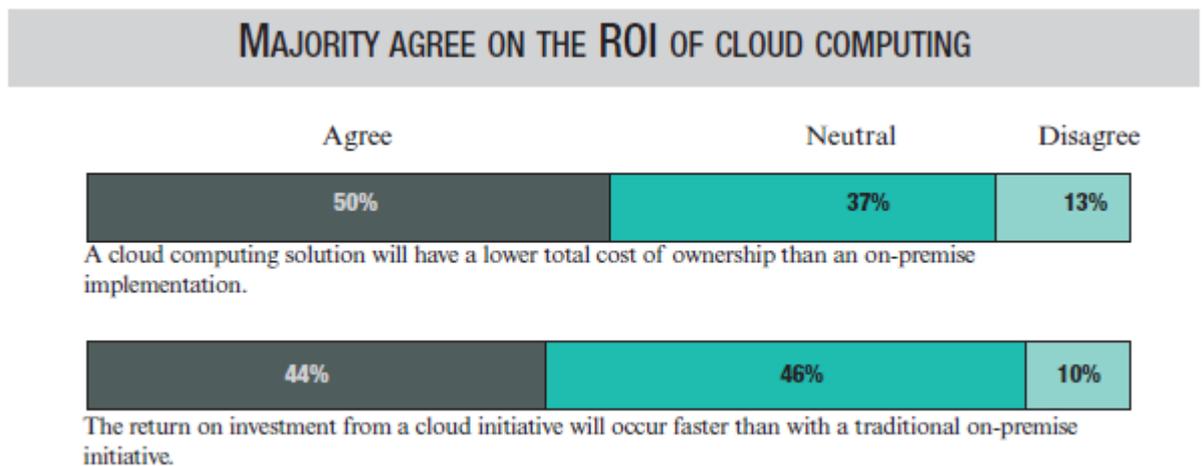


FIGURE 2.10: ROI expectations of Cloud Computing [35]

In this survey, majority answered that cloud computing solution will have a lower total cost of ownership than an on-premise implementation (50%) and return investment from a cloud initiative will occur faster than with a traditional on-premise initiative(44%). Minority disagree with these statements 13% and 10% respectively.

Counterpoise expenditures for moving to the cloud, training staff, reshuffling infrastructure and applications should be considered too.

To conclude, on example of USA agencies we can tell that customers generally are optimistic about cost reduction and ROI but ignorant about cloud security possibilities. It simply demonstrates the third stage of Ganter Hype Cycle and what is needed - to learn more about possible cloud computing applications in due course to judge fair.

2.3.4 Prerequisites for Cloud Computing

Cloud Computing has become a trendy innovation in software operation for many businesses. The software is usually hosted on multiple servers in various locations. The costs of cloud computing are normally based on a usage model, with payments being charged on a time usage basis or an occurrence basis. The most attractive cloud computing advantage for enterprise people is cost savings. But along with this doubtless advantage and several more there are some significant risks concerning cloud computing.

So the main advantages of cloud computing are:

1. Pay-as-you-go model. Reduced costs due to computer expenses (one doesn't have to process power or hard disk space that is required by desktop applications) and hardware expenses (one have no need to buy additional server for peak loads).
2. Improved performance. Providers are trying to serve the requests during (milli-) seconds to prevent bouncing from service, while desktop application performance can take more time depending on memory size and number of programs and processes loaded to the memory.
3. Scalability.
4. Increased storage capacity. Cloud computing is concerned to virtually unlimited phenomenon and it is applicable to storage volumes. Any amount of information can be stored on the cloud. It is not tied to PC's hard drive any more.
5. Highly automated. Web-based applications are updated automatically and are ready for use on next log in. No longer company's IT personnel should take care of updates, downloads and upgrades.
6. Increased flexibility. All documents created via web-based application can be read by everyone else who accesses application. There are no format incompatibilities.
7. More mobility. All the documents can be accessed whenever one has a computer (or mobile phone) and Internet connection.
8. Increased data reliability. The most cloud providers report about less than 1% of inaccessibility of the clouds. It is less than pc's practice. Also in case one's computer crushes or something happens to hard disk there is risk to lost information forever or covering this information can take quite a long time while data on cloud can be accessed from another computer and is replicated in different locations for reliability reasons. [3]

The main disadvantages of cloud computing are:

1. Lost control over information and data
2. Increased dependency on third party, decreased business flexibility
3. Security
4. Unstable cost structure
5. Integration problems and complexity of moving process.[5]

Therefore, according to cloud computing may be a solution for next cases, when:

- Processes, applications and data are largely independent;
- Cost is an issue;
- Web is desired platform;
- High level of security is not required;
- Application is new.

Thus, not all computing resources should exist in the clouds. Examples of the cases where computing resources are the most likely to be cost-effective and reasonable for performing into clouds are:

1. R&D projects. By it here means the projects that need some resources for computing and estimating some values for particular task and won't be needed after.
2. Low-priority business applications (Web-based collaboration, applications that are used for mobile workers or on different machines)
3. Web-based collaboration services. [1]

Chapter 3

Practical utility of ESB-based systems on the cloud.

In this chapter we will discuss the main approaches for modeling and usage of ESB-based system on example of Oracle SOA Suite which include Oracle Enterprise Service Bus (Mediator) and can be deployed with Oracle Service Bus. Next Amazon prospectives and enterprise deployment prospectives are discussed in regards to Oracle products.

3.1 Oracle SOA Suite features

Oracle SOA Suite is a part of the Oracle Fusion Middleware family of software products. It is a comprehensive, hot-pluggable software suite to build, deploy and manage Service-Oriented Architectures. As of May 2011 Oracle Corporation markets Oracle SOA Suite version 11g Release 1 Patchset 3 (11.1.1.5).

The key components within Oracle SOA platform include:

- Oracle SOA Composite Platform
- BPEL Process Manager (BPEL PM)
- Human Workflow (HWF)
- Business Rules
- Mediator
- Oracle Service Bus (OSB)

- Oracle Business Activity Monitoring (BAM)
- Oracle Complex Event Processing (CEP)
- Oracle Enterprise Manager (EM)
- Oracle Web Services Manager (OWSM)
- Oracle Enterprise Repository and Registry (OER / OSR)
- Oracle Data Integrator (ODI) [27]

An architectural model for SOA can be divided into two high level categories:

- Business Service Layers
- SOA Infrastructure

This categorization provides a separation of IT concerns and business logic concerns and that is critical to the success of a SOA implementation. Separating concerns into these layers allows greater agility and flexibility in the architecture.

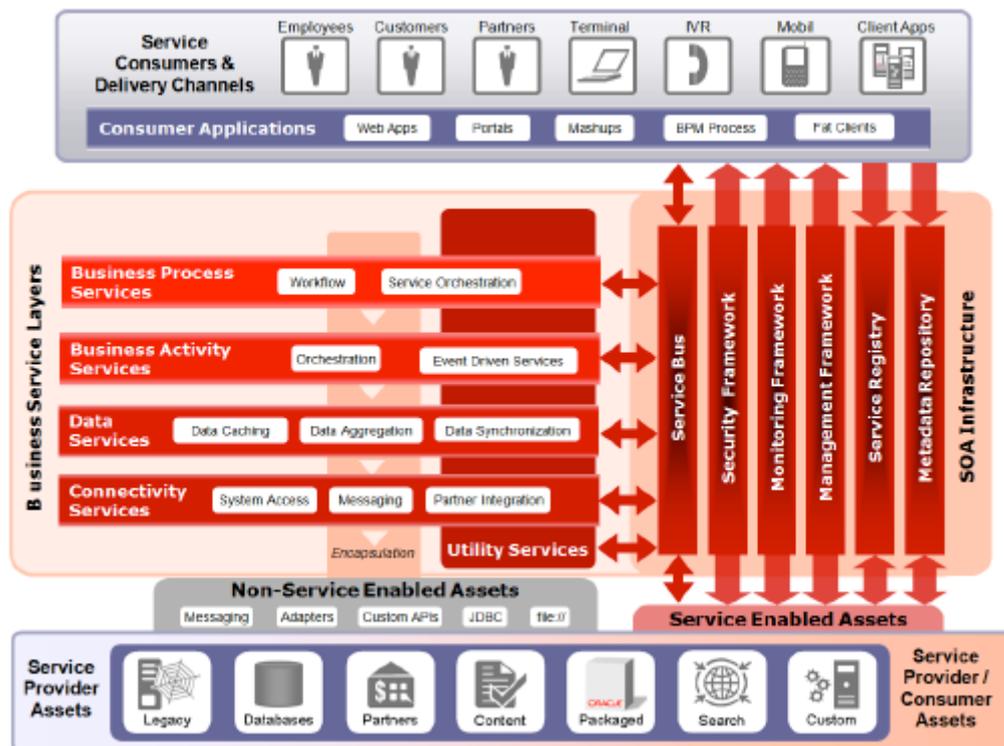


FIGURE 3.1: An Architectural Perspective of a SOA Platform [27]

The Business Service Layers deals with concerns associated with process automation and business logic development. It does not address IT concerns such as abstracting

the physical location of a service endpoint or ensuring end-to-end resilience to system failures. This layer also addresses business level visibility issues by integrating with business events for event driven SOA.

The SOA Infrastructure provides capabilities to publish, discover, share, monitor and manage services. Services utilize the connection management and recovery functions provided by the SOA infrastructure. It also decouples service providers from consumers. Decoupling enables location transparency and ability to introduce new versions of a service without impacting all clients of the service.

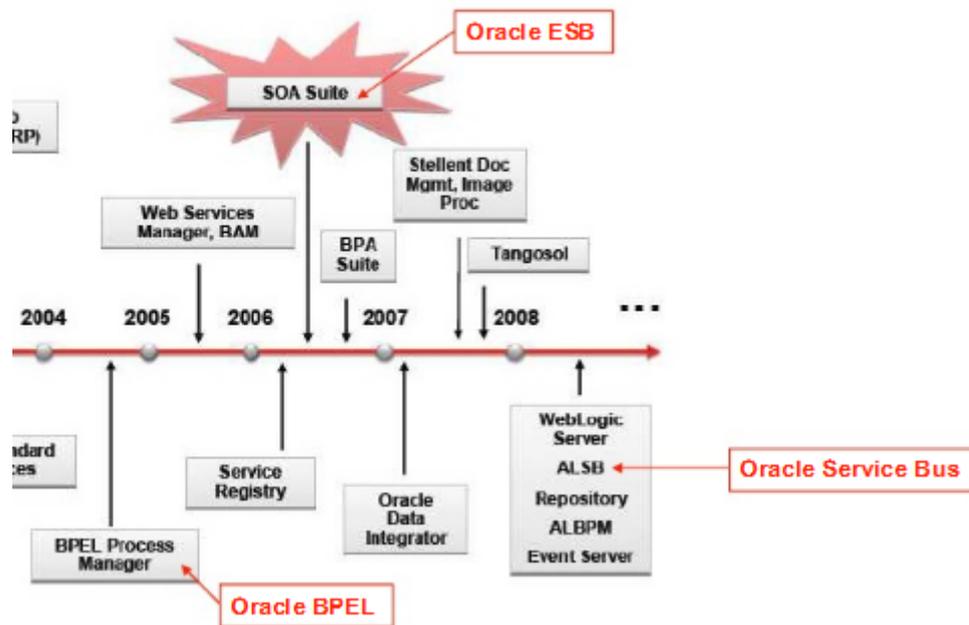


FIGURE 3.2: History of the Oracle SOA Platform [13]

BPEL Process Manager is the primary composition, orchestration and process engine is the SOA Suite. Oracle Enterprise Service Bus (OESB) was the primordial ESB, but after acquisition of BEA it is set to provide mediation services between SOA Suite Components and in 11g Release it is known as Mediator. Now OESB and BPEL Process Manager are essential parts of Oracle SOA Suite.

Oracle Service Bus (OSB) is a former BEA Agualogic Service Bus (ALSB) that was acquired by Oracle in April 2008. Now it is Oracle's primary service bus and it is preferred platform for interactions external to the SOA Suite. Though it also can be used independently, without SOA Suite.

Thus, Oracle Corporation owns 2 ESBs, one can be used as standalone and another is embedded part of SOA Suite. More about their functionality is discussed in next subsection.

3.1.1 Oracle Enterprise Service Bus vs. Oracle Service Bus

Mediator (Oracle Enterprise Service Bus) is the tiny, light-weight service bus, Oracle Service Bus is the large, powerful bus. Though some functionality is common between OESB and OSB, OSB has much more functions. Oracle Service Bus was built on the base of AquaLogic Service Bus 3.x with adding some features as X-reference, domain-value maps, JCA adapters, global policy management and XSLT tooling. [27] Functionality and crossfunctionality is illustrated in Figure 3.3 below.

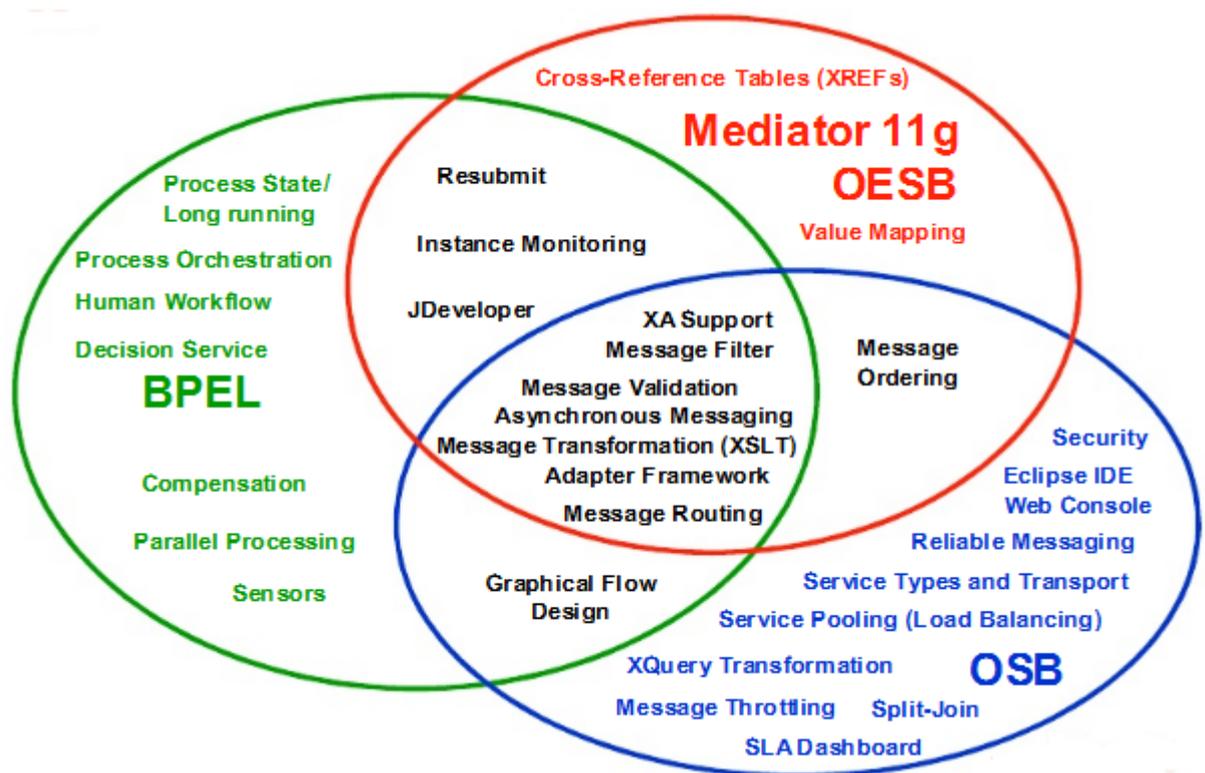


FIGURE 3.3: Oracle SOA Suite 11g Mediator vs Oracle Service Bus [27]

Briefly distinctions between two Oracle ESB's are presented in the next table:

Feature	Mediator	Oracle Service Bus
Functionality	Limited, VETRO* pattern	Extended
Development	JDeveloper IDE	Eclipse IDE or Web Console
Message Transformation	with XSLT	over XQuery and XSLT
Integration with SCA	Yes	Not yet

TABLE 3.1: Mediator vs. Oracle Service Bus comparison

Mediator is limited to simple functionality for the implementation of the VETRO pattern (Validate, Enrich, Transform, Route, Orchestrate), meanwhile Oracle service Bus has

extended functionality important for enterprise-wide Integration, like - Message Throttling, Service Pooling and Reliable Messaging. Mediator unlike OSB can be used and deployed as a SCA component.

3.1.2 Oracle SOA on Amazon cloud

As it was beforementioned there are different types of clouds available for enterprise usage. Amazon is the leader of public cloud suppliers and its Amazon EC2 product is the most commonly-used decision for ones decided to use public clouds. Despite of all disadvantages and skeptic attitude, Amazon provide some features that can be combined with Oracle SOA Suite for profit. They are presented in Table 3.2 on page 25. [36]

Amazon had an Amazon Machine Image (AMI) with Oracle BPM 11gR1, which included Oracle SOA Suite 11gR1 Patchset-2. It was a fully configured image which required no installation and was ready-to-use right after starting the instance. In the process of writing this work (May, 2011), AMI disappeared. But it was a nice try to get hands on experience with Oracle SOA Suite within few minutes and good start for combining enterprise applications with remote cloud servers. We really hope that the next addition of Oracle AMI's on Amazon will be emitted soon as they were available (different configurations) during last 2 years.

3.2 Oracle implementation in companies

Many companies reported about using or planning to try ESB-based systems. Beijing Mobile, Hana Bank, Texas Government, Utah Department on transportation and a lot of others are Oracle clients of SOA Suite and Oracle Service Bus Middleware products. Nevertheless, there is no explicit information on cost-efficiency of such decisions. Moreover, unfortunately there are no studies that were aimed to prove cost-efficiency of using cloud computing for enterprise purposes. It can be explained with quite solid initial efforts to start with new architectures (ESB-based systems and cloud computing) and protracted payback.

Though these materials can be hardly used for reporting on all lines of business, Oracle technical reposts are the only available sources. The general patterns to be seen after building ESB-based system are next:

- Accelerated development cycles by 40%–50%
- Cut the total cost of ownership by approximately 50%

SOA Pattern	Applicability to Cloud Platform
<p>Adapter Pattern: It allows to connect to multiple technologies like messaging, databases without knowing much of the internals of them. For example Database Adapter is basically to service enable databases without having to write SQL. Oracle 11g SOA Suite supports database adapters to multiple databases.</p>	<p>Amazon RDS (Relational Database Services) exposed as an SOAP based interface which can be connected as a Adapter pattern using the regular Services Interface as part of Oracle SOA Suite.</p> <p>Amazon Simple DB is a non relational database service, which can be connected using the regular SOAP interface and can satisfy the needs of Adapter pattern in SOA. Amazon EBS and S3 based instances can satisfy the needs of File Adapter in a SOA.</p>
<p>Mediator Pattern: Mediator is the component in charge of interconnecting the other components within a composite application. Oracle SOA suite uses mediator pattern for intra composite mediation.</p>	<p>With Multiple EC2 Instances can run different services like basic compute services, storage services like EBS, database services like RDS, Simple DB, it is very much possible to have a mediator that interconnects the multiple services towards a business process realization.</p>
<p>Binding Pattern: The binding can be used expose the SOA composite applications over SOAP .</p>	<p>With Elastic IP, Services can be exposed to outside Cloud Consumers that will act as a Binding.</p>
<p>Business Process Orchestration Pattern: This pattern aims at realizing long running business process through orchestration of multiple individual services. Oracle 11g Suite support BPEL Component to perform orchestration.</p>	<p>With various services exposed as Services as explained above, Business process can be orchestrated in a cloud platform much similar in a normal data center.</p>
<p>Proxy Pattern: A business service describes the end point being virtualized, its policies and interfaces. A proxy service is what the service bus exposes to service consumers and implements the virtualization logic.</p>	<p>With Elastic IP acting as a proxy interface to multiple virtual machines and abstracts from issues like Virtual machine failure or migration of Virtual machines, Cloud platform adapts the Proxy pattern to support virtualization.</p>
<p>Asynchronous Pattern: In a real world scenario Asynchronous pattern is helpful when</p> <ul style="list-style-type: none"> • response might not come back instantly from the service provider • given request needs to be send to multiple providers or vice versa • service consumer did not know who will provide the service • consumer and provider will not be online at the same due to geographical and time zone constraints 	<p>Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. Using Amazon SQS, you can decouple the components of an application so they run independently, with Amazon SQS easing message management between components. Any component of a distributed application can store any type of data in a fail-safe queue. Any other component can then later receive the data programmatically using the SQS API.</p>

TABLE 3.2: **SOA Design Patterns in the Cloud**

- Eliminated vendor lock-in
- Increased business agility
- Facilitated easy interoperability with third-party systems
- Reduced hardware requirements
- Simplified clustering and security
- Developed an SOA that can be used by other subsidiaries

Chapter 4

Testing ESB-based system on the cloud

So as to explain the ESB-based system on the cloud we have to present several notions for Oracle systems and distributed architectures that will be used further.

A domain is an interrelated set of WebLogic Server resources that are managed as a unit. A domain includes one or more WebLogic Server instances, which can be clustered, non-clustered, or a combination of clustered and non-clustered instances. A cluster is part of a particular WebLogic Server domain. A domain can include multiple clusters. A domain also contains the application components deployed in the domain, and the resources and services required by those application components and the server instances in the domain. Examples of the resources and services used by applications and server instances include machine definitions, optional network channels, connectors, and startup classes. [24]

Clustered WebLogic Server instances behave similarly to non-clustered instances, except that they provide failover and load balancing. The process and tools used to configure clustered WebLogic Server instances are the same as those used to configure non-clustered instances.

By and large WebLogic Domain consist of Administration Server, Managed Server and cluster. There can only be one administration Server in domain and zero to N Managed Server.

Administration Server is WebLogic Server instance that maintains configuration data for a domain. Any WebLogic Server instance apart from Administration Server is called as Managed Servers.

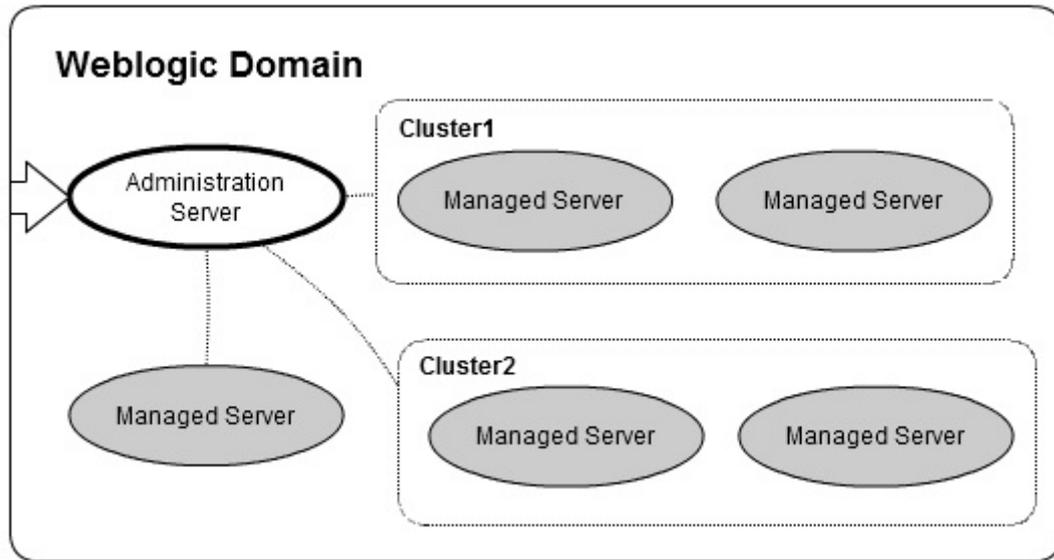


FIGURE 4.1: Weblogic Domain

Group of WebLogic Managed Server Instances that work together to provide high availability and scalability for applications is called cluster. WebLogic Servers with in cluster can run on same machine or different machines. These are also called as managed Server cluster. [25]

Weblogic domain can be realized in different configuration combinations, involving different Tier means. Tier represents logical divisions of an Applications service:

- Web Tier

The web tier provides static content (for example, simple HTML pages) to clients of a Web application. The web tier is generally the first point of contact between external clients and the Web application. A simple Web application may have a web tier that consists of one or more machines running WebLogic Express, Apache, Sun One Web Server, or Microsoft Internet Information Server.

- Presentation Tier

The presentation tier provides dynamic content (for example, servlets or Java Server Pages) to clients of a Web application. A cluster of WebLogic Server instances that hosts servlets and/or JSPs comprises the presentation tier of a web application. If the cluster also serves static HTML pages for your application, it encompasses both the web tier and the presentation tier.

- Object Tier

The object tier provides Java objects (for example, Enterprise JavaBeans or RMI classes) and their associated business logic to a Web application. A WebLogic Server cluster that hosts EJBs provides an object tier. [26]

Architecture of cluster defines how various tiers of an application (Web, Presentation, Object) are deployed on one or more clusters.

Type of Oracle WebLogic Server Architecture:

- **Basic Recommended Architecture**
All tier of Web Application (Web, Presentation, Object) are deployed on same WebLogic Server cluster.
- **Multi-Tier Recommended Architecture**
Different tier of application (Web, Presentation, Object) are deployed to different cluster, One WebLogic cluster for Web, Presentation tier and another WebLogic cluster for Object Tier (EJB, RMI)
- **Two-Tier Proxy Architecture**
The same as basic recommended architecture but Web Tier is hosted on existing Web Server (IIS, Apache, Netscape). Static HTML content is served via existing Web Server where as Presentation or Object Tier request are served from WebLogic Cluster.
- **Multi-Tier Proxy Architecture**
The same as Multi-Tier recommended Architecture but web tier is hosted on existing Web Server (IIS, Netscape, Apache) with WebLogic Proxy Plug-In on WebLogic Server. Proxy Plug-In- is WebLogic Server extension to HTTP Server (Apache, Netscape, IIS) to access clustered servlets provided by WebLogic cluster. [26]

4.1 Test case description

For test case purpose we will use mortgage broker scenario describing a typical loan application process for Oracle Enterprise Service Bus, which can be found in examples for this software on official website. Though many possible examples for testing were considered but they were denied due to issues concerning configuration and installation processes.

Oracle Service Bus provides intelligent message brokering between business services (such as enterprise services and databases) and service clients (such as presentation applications or other business services) through proxy services. [27] Both business services and proxy services can be configured in Oracle Service Bus Console.

Proxy services are Oracle Service Bus definitions of intermediary Web services that Oracle Service Bus implements locally on WebLogic Server. With Oracle Service Bus message brokering, service clients exchange messages with an intermediary proxy service rather than working directly with a business service.

Business services are Oracle Service Bus definitions of the enterprise services that exchange messages during business processes. To configure a business service one must specify its interface, type of transport it uses, its security requirements, and other characteristics. A business service definition is similar to that of a proxy service, but it does not have a pipeline.

A primary mortgage company uses Oracle Service Bus to route loan applications to appropriate business services based on the interest rate requested. An application containing a request for a rate less than 5% requires management approval and is routed to an appropriate business service for processing (Manager Approval Business Service). All other loan applications are routed to the appropriate business service for processing (Normal Loan Business Service).

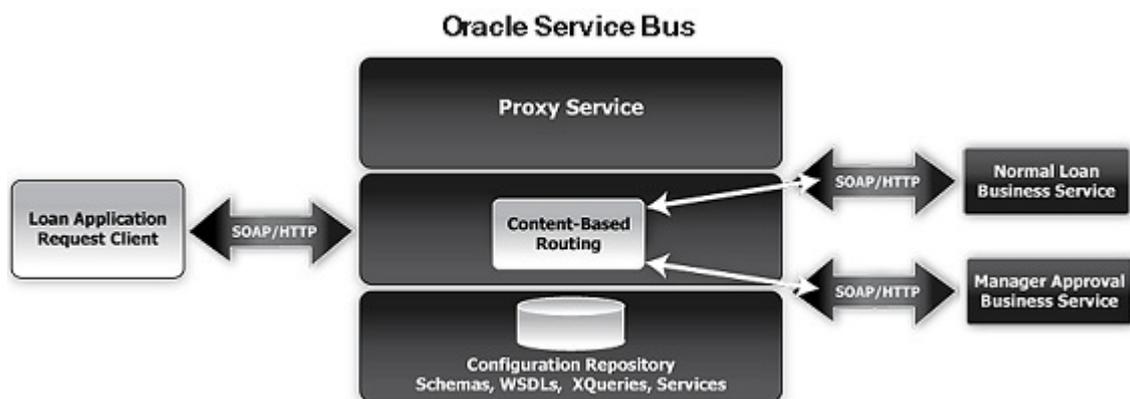


FIGURE 4.2: Loan Application Request Web Service via Oracle Service Bus [27]

A client sends a loan application (Appendix A) to a proxy service named LoanGateway1. The default proxy service has a conditional routing stage that checks the value of the requested interest rate in the loan application document. If the interest rate is less than 5%, the loan application is routed to the ManagerLoanReview business service; otherwise it is routed to the NormalLoan business service. The target business service returns a response (Appendix B).

We will assume that 70% of requests can be served by Normal Loan Business Service and only 30% of requests have to be processed by Manager Approval Business Service.

4.2 Test environment requisites

Test environment will be distributed on the clouds in form of Basic Recommended Architecture (All tier of Web Application (Web, Presentation, Object) are deployed on same WebLogic Server cluster.) Our domain will consist of one Admin Server and two Managed Server - micro and small - into one cluster - sample cluster, herewith Admin Server and small Managed server physically are running on small machine and micro Managed Server is assigned to micro Machine. For naturality refer to Figure 4.3 and Table 4.1

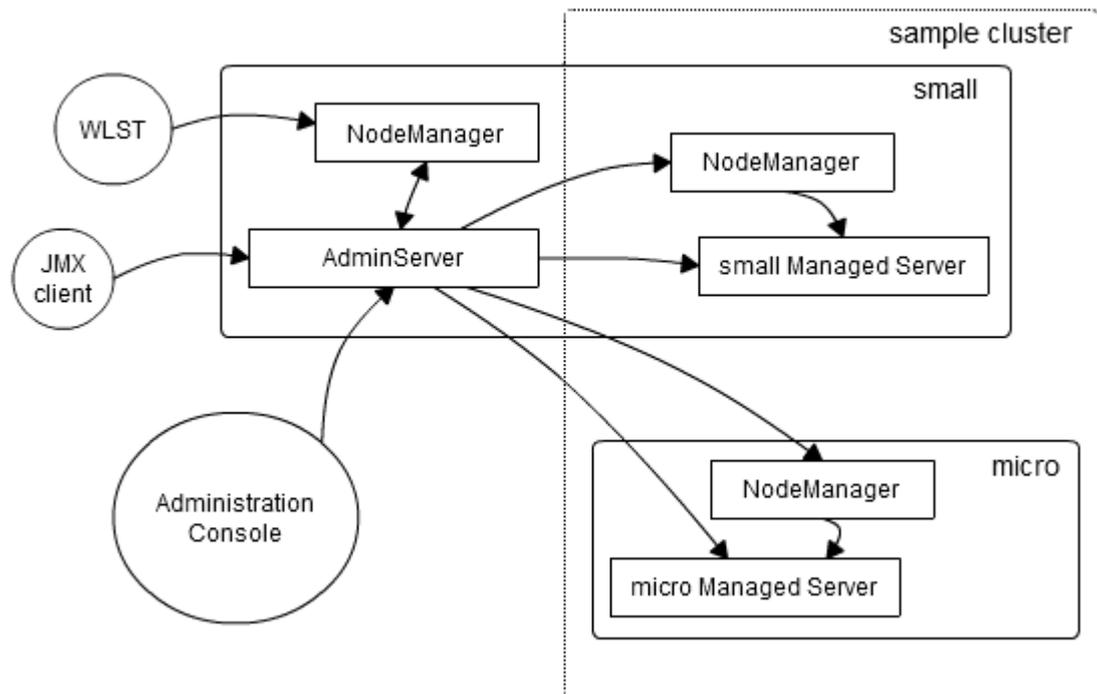


FIGURE 4.3: Test case architecture

Machine	Small Machine	Micro Machine
Amazon Image	m1.small	m1.micro
IP	10.32.145.115	10.194.33.139
OS	Ubuntu (x86-64)	Ubuntu (x86-64)
Servers	Admin, 1 Managed	1 Managed
OSB Software	Oracle JRockit JDK 28.1.3 WebLogic Server 11gR1 PS3 (10.3.4) Oracle Service Bus 11gR1 (11.1.1.4.0)	Oracle JRockit JDK 28.1.3 WebLogic Server 11gR1 PS3 (10.3.4) Oracle Service Bus 11gR1 (11.1.1.4.0)
Additional Software	Oracle Database 10g Release 2 (10.2.0.1) Express Edition for Linux x86 Repository Creation Utility 11.1.1.4.0	-

TABLE 4.1: **Machine specification**

Chapter 5

Results and analysis of tests

5.1 Refactoring idea

The initial idea of this work was remodeling distributed ESB-based systems on the cloud thus to provide the best capacity on the base of communication load. According to the observation of Distributed Systems group communication load in particular was the most cost-risky issue in the process of migrating applications to the cloud. [4] The services that communicate the most had to be pushed to one machine in that way to reduce load on communication channel and lessen overall communication time among services. Thus basically all services will be grouped according to communication rates among each other. The most tightly-bundled services should go to one cluster within one domain leaving for outcluster communication the least intensive channels.

For this purposes, I have chosen Oracle Service Bus and Oracle Fusion Middleware products. Though, there is no marketing researches on ESB usage and proper comparison of market share and user satisfaction, we decided to go on with Oracle products as its product functionality and possibility for integration with other software (as BPEL Process Management Engines, Databases, Database Repositories) is one of the best on the market. Additional motivation was active integration of Oracle SOA Suite and other Oracle products to the cloud environment (Amazon EC2 image, embedded Oracle SOA Suite features).

In Oracle Service Bus example described in previous chapter, there are Loan Application request client, one Proxy service and two Business Services. Business Services can be deployed on the clouds based on need to have additional resources to serve the requests, scalability need etc. Within initial architecture of two physical services there are 4 possible combinations of deploying two Business services. However it was decided to investigate how communication load influence request speed in two combinations:

1. Normal Loan Business Process on small machine (1 Admin Server, 1 Managed Server + Proxy service) + Manager Approval Loan Business Process (1 Managed Server)

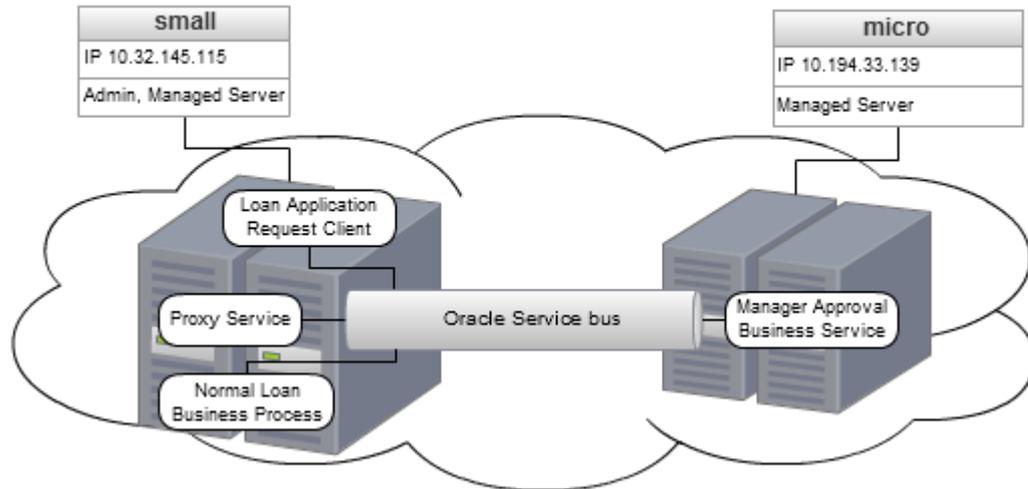


FIGURE 5.1: First configuration of test case

2. Manager Approval Loan Business Process (1 Admin Server, 1 Managed Server + Proxy service) + Normal Loan Business Process on small machine (1 Managed Server)

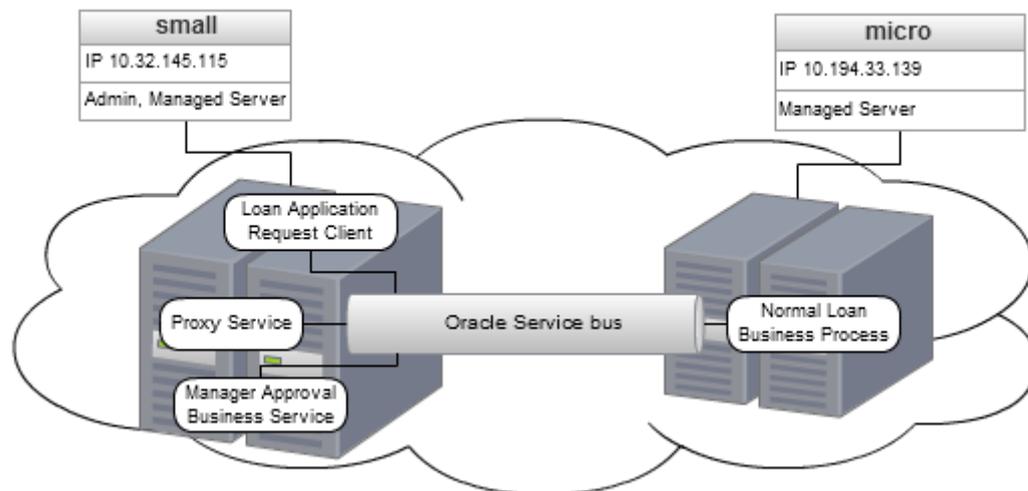


FIGURE 5.2: Second configuration of test case

The tricky part is that distribution of the load is 30:70 for Manager Approval Loan Business Process and Normal Loan Business Process respectively. Next I will monitor and discuss results.

For this purpose, one can use Oracle Web Console. Oracle Service Bus as well as other Oracle products can be managed from Web Console. Start/Stop servers, make diagnostics, see log files and a lot of other activities can be performed via Web Console. Also a bunch of monitoring features are embedded into Web Console. After enabling monitoring developer can trace next parameters:

- amount of requests to each service
- request errors
- minimum response time
- maximum response time
- average response time

Next figures illustrate monitoring of Proxy Service and Normal Loan processing service - Figures 5.3 and 5.4. Note that failure rates are not normal for test cases, they are due to unavailability of one of the servers.

Service Metrics	Pipeline Metrics
General	
SLA Alert Count	0
Pipeline Alert Count	0
Min Response Time	39 msec
Max Response Time	1 sec 657 msec
Overall Avg. Response Time	329 msec
Message Count	60
Error Count	19
Success Ratio	68.00%
Failure Ratio	31.00%
WS Security Errors	0
Validation Errors	0

FIGURE 5.3: Proxy service statistics

General	
SLA Alert Count	0
Min Response Time	50 msecs
Max Response Time	1 sec 359 msecs
Overall Avg. Response Time	361 msecs
Message Count	25
Error Count	11
Failover Count	0
Success Ratio	56,00%
Failure Ratio	44,00%
WS Security Errors	0
Validation Errors	N/A

FIGURE 5.4: Normal Loan service statistics

The performance results for the first configuration of services explained earlier on the cloud - Table 5.1

Characteristic	Proxy	Normal Loan	Manager Approval Loan
Min Response Time, msecs	39	35	35
Max Response Time, msecs	218	134	211
Overall Avg. Response Time, msecs	109	98	95
Message Count	40	28	12
Success Ratio, %	100	100	100

TABLE 5.1: First configuration of test services on the cloud

The performance results for the second configuration of services on the cloud - Table 5.2

The general amount of test is 40 for both configuration - 28 for Normal Loan Business Process (70%) and 12 for Manager Approval Business Process (30%). The success ratio is 100% for all cases. Expectedly, configuration when heavy-communication services

Characteristic	Proxy	Normal Loan	Manager Approval Loan
Min Response Time, msec	44	37	39
Max Response Time, msec	328	390	129
Overall Avg Response Time, msec	150	110	91
Message Count	40	28	12
Success Ratio, %	100	100	100

TABLE 5.2: Second configuration of test services on the cloud

are based on one machine demonstrate better minimum, maximum and overall average response time.

5.2 Experience conclusion for ESBs migrating to the cloud

Much bigger systems were also considered in this concern as well as enabling different types of clouds (public - Amazon and private - Eucalyptus), OS architectures and OS systems. But many restrictions concerning deployment and configuration were run into, which have consumed a lot of the time in the thesis execution. Here the thesis discusses experience of deploying ESBs on the cloud and the issues associated with the migration of enterprise application to the cloud.

Before considering Oracle Service bus test case, I had tried several other options with another middleware software. Servicemix 4.2 and Mule ESB 3.0 were considered. Loan Broker example of Mule ESB was successfully deployed on SciCloud [40] of Tartu university. It was nice start, but Community (free) Edition had less functionality and Enterprise Edition had only 30 days of trial. Taking into account aforementioned and possibility to go for a bigger systems I decided to try with Oracle. But it was not just shift to Oracle, I had to start with Amazon Cloud Services. During this time I had to go through fresh waters of working with public cloud computing, working with frameworks, firewalls and internal Oracle restrictions in the way to connect the components between SciCloud and Amazon. Lot of things are learned but the tools required lot of effort to run smoothly.

Thus, Oracle SOA Suite was deployed on Amazon EC2 and Eucalyptus Scicloud [40] of Tartu University, Oracle service Bus was deployed on Amazon EC2 and Eucalyptus and on two Amazon EC2 clouds. All this work required a lot of administration skills

not only in means of Ubuntu OS but also managing and fixing Oracle database, work with euca tools and amazon tools, rearrangement of system in concern with memory restrictions and so on. All this configurations were dismissed in regard of personal ideas and unwillingness to increase cloud resources again, real desire to use as small instances as possible. A lot of time was put to fit configuration to this beliefs. And finally I decided to try more straightforward examples to test my primary ideas and to colligate all my theoretical knowledge and observations gained so far.

However Oracle service bus also throws several challenges. Some of the issues with Oracle Service Bus that one should keep in mind while deploying the system:

1. Java heap memory problem. Only Jrockit java is applicable for configuring domains in Weblogic otherwise Java heap memory problem appears.
2. Incompatibility of different versions. It is the best solution to check all peaces of software for version information. One version more obsolete or newer can stop from running some application.
3. No possibility to get updates of the software with "apt-get update" standard command in Ubuntu.
4. Failure to satisfy prerequisites. It is advised to skip prerequisites during installation as none of the cloud (either Amazon or Eucalyptus cloud) is able to meet them, which leads to errors you never can identify correctly.
5. Ubuntu groups permission conflicts
6. Big community support - few good comments. Oracle Support forum probably is the best place to look for solutions to issues, but still users are the best to help each other. Not a lot of parental support.
7. No good documentation - redundant, obsolete, incomplete. Releases are faster to come than supporting documentation. This concerns errors description, use cases, on-time tutorials and so on. The most valuable contribution to Oracle community documentation is provided by Oracle blogs.

To conclude, I have to say that a great deal in going into ESBs is perceiving the differences that various vendors put under ESB brand. Unless one has clear understanding what is particular Service bus and why should he implement spme functionality exactly using this Service bus, using Service bus on the cloud is gamble.

Chapter 6

Conclusion

This work is dedicated to ESB-based systems on the cloud and auxiliary theoretical aspects and practical guidelines. It covers ample introduction and description of Enterprise service bus, describes practices of engaging cloud usage for organizations, look through integration patterns by the example of Oracle Middleware software and concludes with experience summing up.

Despite of my own motivation to start this work in order to try out cloud computing resources for enterprise systems benefit, I faced several issues I want to mention.

First issue is concerned with lack of unified definition of Enterprise Service bus. Being as Utopian idea or some vendors brand name, indeed it is hard to take, that "ESB is not a new software product - it's a new way of looking at how to integrate applications, coordinate resources, and manipulate information".

The work done during preparation of this thesis work has rather applied character, than scientific. Starting from basic knowledge of working with the remote services, some essential experience with work on the Eucalyptis (private) and Amazon (public) cloud was gained as well as administration skills for managing Windows- and Linux-OS machines, working with Oracle products on aforementioned systems, installation and configuration, setting domains and clusters, distributing applications among one and more clusters.

While working on this work two well-known winged words about cloud computing were experienced in full – first, ESB mantra "configuration rather than coding" and the second "don't dare moving to cloud unless facile and easy-deploying running on the premise". Moving any system to the cloud is not a trivial task as it is concerned to question of administration of remote servers, firewall settings, increased number of users and modification of their roles and much more you could not even think of. Hereof comes

another principle of using ESB-based systems (and eventually distributed systems) – “do not use it until you really need it”.

Chapter 7

Sisukokkuvte

Tanapaeval ettevotted konkurentsivoime pressi all: pakutava teenuse kiirus, ebatpuse hind, võimalus kasutada rakendus lauaarvutist, sulearvutist ning teistest mobiilseadmetest. Need uued tingimused suruvad ari taiendavaid ressursse ning konfiguratsioone otsida selleks, et pakkuda parimad teenused. Selle eesmärgiga me uritame anda uldised vihjed kuidas uhendada ettevotted, pilv-arutus ning hajus vahetarkvara uhes sisteemis.

ESB-pohised sisteemid ja nende kontekstis voetud Teenus Orienteeritud Arhitektuur on hinnatud, selleks et vaadelda nende voimaliku rakendust ja kasulikkust kaug-serverites. Selles toos on esmakordselt vaadeldud hajus ESB-sisteemid (Liitunud, Internet ja korralik Hajutatud) toestades, et see idee ei ole uus vaid erinevat uurimisrühmad on fokuseeritud erinevate aspektide peal. Statistilised andmed pilv-rakenduste kohta on kasutatud laialt levitatud pilve turvalisuse ning finants-efektiivsuse vaidete vastuvaitmiseks. Lisaks skeptiline suhtumine on kasitletud viitades vastava tehnoloogia Hype tsuklide peale.

Need naitavad nende tehnoloogiate voimalik kiirarendus lahimas tulevikus isegi siis kui hetkel need pole veel arenenud piisavalt. Peale selle on esitatud lihtne abi materjal ESB-pohiste sisteemide integreerimisest ning pilv-arvutuste kaustusest. Too kokkuvottes vorreldakse ideed ja tulemusi ning kirjeldatakse vastavat kogemust.

Appendix A

Loan application request to Normal Loan Processor

```
Request Document
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  </soap:Header>
  <soapenv:Body>
    <exam:processLoanApp xmlns:exam="http://example.org" >
      <loanRequest xsi:type="java:LoanStruct" xmlns:java="java:normal.client" xmlns:xsi="http://www.w3.org
        /2001/XMLSchema-instance" >
        <!--Optional-->
        <java:Name>Dumas</java:Name>
        <!--Optional-->
        <java:SSN>12345</java:SSN>
        <!--Optional-->
        <java:Rate>5.9</java:Rate>
        <!--Optional-->
        <java:Amount>10000000</java:Amount>
        <!--Optional-->
        <java:NumOfYear>5</java:NumOfYear>
        <!--Optional-->
        <java:Notes>Manager Loan</java:Notes>
      </loanRequest>
    </exam:processLoanApp>
  </soapenv:Body>
</soapenv:Envelope>
```

Appendix B

Response From Normal Loan Processor

```
Response Document
<env:Envelope env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >
  <env:Header/>
  <env:Body>
    <m:processLoanAppResponse xmlns:m="http://example.org">
      <return>
        <java:Name xmlns:java="java:normal.client">Dumas</java:Name>
        <java:SSN xmlns:java="java:normal.client">12345</java:SSN>
        <java:Rate xmlns:java="java:normal.client">5.9</java:Rate>
        <java:Amount xmlns:java="java:normal.client">10000000</java:Amount>
        <java:NumOfYear xmlns:java="java:normal.client">5</java:NumOfYear>
        <java:Notes xmlns:java="java:normal.client">
          APPROVED BY THE &lt;i>&lt;b>NORMAL&lt;/b>&lt;/i> LOAN APPLICATION PROCESSING
          SERVICE
        </java:Notes>
      </return>
    </m:processLoanAppResponse>
  </env:Body>
</env:Envelope>
```

Bibliography

- [1] Leavitt, N. Is Cloud Computing Really Ready for Prime Time?, *IEEE-2009*
- [2] Paul Hofmann. ERP is Dead, Long Live ERP *IEEE Internet Computing, vol. 12, no. 4, pp. 84-88*
- [3] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the Clouds: A Berkeley View of Cloud Computing *Technical Report EECS-2009-28, EECS Department, University of California, Berkeley.*
- [4] S. N. Srirama, O. Batrashev, P. Jakovits, E. Vainikko Scalability of Parallel Scientific Applications on the Cloud *Scientific Programming Journal, Special Issue on Science-driven Cloud Computing, IOS Press. DOI 10.3233/SPR-2011-0320 (In Print)*
- [5] Robert C. White, Jr. Cloud Computing: Advantages and Disadvantages, *August 24, 2010*. Available <http://boardroombrief.com/theblog/2010/08/24/cloud-computing-advantages-and-disadvantages/>
- [6] Presentation Cloud Computing and Enterprise Architecture. Available <http://www.slideshare.net/Linthicum/cloud-computing-and-enterprise-architecture>
- [7] Mike Maxey. Cloud Computing Public or Private? How to Choose Cloud Storage *October 14, 2008*. Available <http://www.sys-con.com/node/707840>
- [8] Rob Barry. ESBs in the cloud: Tricky in the early going, *June 09, 2010*. Available at <http://searchsoa.techtarget.com/news/1514427/ESBs-in-the-cloud-Tricky-in-the-early-going>
- [9] Greg Flurry. Exploring the Enterprise Service Bus, Part 1: Discover how an ESB can help you meet the requirements for your SOA solution *April, 2007* Available <http://www.ibm.com/developerworks/webservices/library/ar-esbpat1/index.html>
- [10] Rachel Reinitz and Greg Flurry. Exploring the Enterprise Service Bus, Part 2: Why the ESB is a fundamental part of SOA *September, 2007* Available <http://www.ibm.com/developerworks/webservices/library/ar-esbpat2/>

- [11] Greg Flurry and Mei Y. Selvage and Dr. Guenter Sauter and Eoin Lane. Exploring the Enterprise Service Bus, Part 3: Four approaches to implementing a canonical message model in an ESB *October, 2008* Available <http://www.ibm.com/developerworks/webservices/library/ar-esbpat3/index.html>
- [12] Waseem Roshen. Services-based enterprise integration patterns made easy, Part 4: Enterprise service bus *December, 2008* Available <http://www.ibm.com/developerworks/webservices/library/ws-intpatterns4/index.html?ca=drs->
- [13] Guido Schmutz. Oracle SOA Suite 11g Mediator vs Oracle Service Bus OSB *November, 2009* Available <http://www.scribd.com/doc/23622536/Oracle-SOA-Suite-11g-Mediator-vs-Oracle-Service-Bus-OSB>
- [14] Adrien Louis. ESB Topology Alternatives *May, 2008* Available at <http://www.infoq.com/articles/louis-esb-topologies>
- [15] <http://www.mulesoft.org/documentation/display/MULE3INTRO/Home>
- [16] <http://petals.ow2.org/demonstrations.html>
- [17] <http://wiki.open-esb.java.net/Wiki.jsp?page=Documentation>
- [18] <http://www.eaipatterns.com/SystemManagementExample.html>
- [19] Definition enterprise service bus ESB Available at <http://searchsoa.techtarget.com/definition/enterprise-service-bus>
- [20] Bobby Woolf. Why do developers need an Enterprise Service Bus? *August, 2005* Available <http://www.ibm.com/developerworks/webservices/library/ws-whyesh/>
- [21] Raul Lapeira. ESB is an architectural style, a software product, or a group of software products? *Artifact Consulting, April, 2005*
- [22] Dennis Pilarinos Donald F. Ferguson and John Shewchuk. The Internet Service Bus *October, 2007* Available <http://msdn.microsoft.com/en-us/library/bb906065.aspx>
- [23] SOA programming model for implementing Web services, Part 4: An introduction to the IBM Enterprise Service Bus. SOA programming model for implementing Web services, Part 4: An introduction to the IBM Enterprise Service Bus *July, 2005* Available <http://www.ibm.com/developerworks/library/ws-soa-progmodel4/>
- [24] Atul Kumar. Domain , Administration and Managed Server, Cluster in Oracle WebLogic *July, 2008* Available at <http://onlineappsdba.com/index.php/2008/07/24/domain-administration-managed-server-cluster-in-oracle-weblogic/>

- [25] Atul Kumar. Node Manager in Oracle WebLogic Server *June, 2009* Available at <http://onlineappsdba.com/index.php/2009/06/10/node-manager-in-oracle-weblogic-server/>
- [26] Atul Kumar. Cluster Architecture : Oracle WebLogic Server *August, 2008* Available at <http://onlineappsdba.com/index.php/2008/08/14/cluster-architecture-oracle-weblogic-server/>
- [27] Introduction to the Oracle Service Bus Tutorials Available at <http://download.oracle.com/docs/cd/E1315901/osb/docs10gr3/tutorial/tutIntro.html>
- [28] <http://www.webwrights.co.uk/service-oriented-architecture/>
- [29] Michael Wisler. JBI based ESB as backbone for SOI applications *Report at the International conference in java technology, June, 2007* Available at <http://docs.huihoo.com/soa/JBI-based-ESB-as-backbone-for-SOI-applications.pdf>
- [30] Ross Mason. To ESB or not to ESB *July, 2009* Available at <http://blogs.mulesoft.org/to-esb-or-not-to-esb/>
- [31] Guinard, D. Trifa, V. Karnouskos, S. Spiess, P. Savio, D. Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services *Services Computing, IEEE Transactions on, July-Sept. 2010*
- [32] Takeshi Eto The Cloud: Onward to the Trough of Disillusionment *December, 2010* Available at <http://blog.discountasp.net/the-cloud-onward-to-trough-of-disillusionment/>
- [33] Gartner Hype Cycle Available at <http://www.gartner.com/>
- [34] Fenn Jackie. Understanding hype cycles *When to Leap on the Hype Cycle. Gartner Group, May, 2008*
- [35] Cloud Computing Research Study *by An 1105 government information group, December, 2010* Available at <http://download.1105media.com/>
- [36] . SOA Design Patterns in the Cloud Available at <http://srinivasansundararajan.systems.com/node/1654420/mobile>
- [37] . Cloud Speed Test Results *February, 2010* Available at <http://blog.cloudharmony.com/2010/02/cloud-speed-test-results.html>
- [38] Marc Kelderman. Install Clustered Oracle SOA Suite 11g *February, 2010* Available at <http://www.namredlek.nl/orasoa/InstallSOASuite11gOnCluster-v1.pdf>

-
- [39] Jack Desai. Architecting OSB for High Availability and Whole Server Migration *October, 2009* Available <http://www.oracle.com/technetwork/middleware/service-bus/overview/osb-wp-ha-final-draft-134330.pdf>
- [40] S. N. Srirama, O. Batrashev, E. Vainikko. SciCloud: Scientific Computing on the Cloud *10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010), May 17-20, 2010, pp. 579. IEEE Computer Society* Available at <http://math.ut.ee/srirama/publications/ccgrid10.pdf>