**COIT20259**

**Enterprise Computing Architecture**

**HT1, 2021**

**Research Report on**

**Major Enterprise Computing Architecture**

**Unit Co-Ordinator / Lecturer**

**Wei Li**

**Tutor**

**Prabhu Singh**

**Submitted By**

**Bishal Budhathoki**

**12116421**

Table of Contents

[Introduction 3](#_Toc69332600)

[1. Enterprise computing and available technologies 3](#_Toc69332601)

[2. Layout and Coverage of Report Structure 3](#_Toc69332602)

[Layered Model 3](#_Toc69332603)

[1. Benefit of Layered Design 3](#_Toc69332604)

[2. Main Components of each layer and their Functions 3](#_Toc69332605)

[Service Orientation 4](#_Toc69332606)

[1. Infrastructure as a Service (IaaS) 4](#_Toc69332607)

[2. Platform as a Service (PaaS) 5](#_Toc69332608)

[3. Software as a Service (SaaS) 5](#_Toc69332609)

[4. Which service provides Java EE development and runtime environment? 5](#_Toc69332610)

[Runtime Framework 5](#_Toc69332611)

[1. Transaction 5](#_Toc69332612)

[2. Concurrent Control 5](#_Toc69332613)

[3. Security 5](#_Toc69332614)

[4. Component Packaging and Deployment 5](#_Toc69332615)

[5. Benefit of the above: 6](#_Toc69332616)

[Pros, Cons and Misconceptions 6](#_Toc69332617)

[1. Open-Source VS Proprietary Software 6](#_Toc69332618)

[2. Software Portability 6](#_Toc69332619)

[3. Misconceptions and clarification about cloud computing 6](#_Toc69332620)

[Conclusion 6](#_Toc69332621)

[1. Suitability 6](#_Toc69332622)

[2. Potential Impact 7](#_Toc69332623)

[References 7](#_Toc69332624)

# Introduction

## Enterprise computing and available technologies

Heterogeneous network of systems where the computers can range from big mainframes, supercomputers to PCs, mobile, laptops. All these computers were bought at various time from different vendors running more than one Operating Systems *(Crawford & Farley 2005).* It utilizes various protocols and standards on various server applications that runs on this heterogeneous hardware. Available technologies in this area are dominantly Java EE and .NET. But with these other technologies are: Java Cobra which wrap legacy systems as COBRA servers and encapsulate them to look like normal objects whose methods can be called, Java IDL, RMI.

## Layout and Coverage of Report Structure

In this report, layered model and its benefit and their main components and functions are discussed along with enterprise computing and their available technologies. Various service orientations are also discussed. Runtime framework along with their benefits, their components for packaging and deployment. There is a section before the conclusion talking open source, proprietary software, software portability, cloud computing’s advantage, disadvantage, and their misconceptions.

# Layered Model

## Benefit of Layered Design

There is various advantage of layered model. Some of them are given below:

1. Testability:

Every component is specific to the layer and since the layer can be mimicked, testing is quite easy.

1. Ease of development:

This pattern so popular with very least concerns for the implementation and the application developed by the companies are developed by splitting the skill sets by layers.

1. Separation of Concerns

One layer does not need to know what another layer got like presentation layer concerns is to deal with its logic and same goes to other layer.

1. Closed Layer:

This means that every layer in this architecture is closed with each other. This means that the layer must be passed from layer to layer and cannot jump over the layer *(Richards 2015).*

## Main Components of each layer and their Functions

There are five layers in Layered design model for .NET. They are

1. Presentation Layer

This layer function is to manage every user interface, browser communication logic.

1. Business Layer

This layer executes particular business rules related with the request. Business object and Data Access Layer are the components of this layer.

1. Services Layer

Services that are to be provided is sorted and provided. Most of the times business layer, service layer and persistence layer are combined as one.

1. Persistence Layer

It gets the data from the database layer.

1. Database Layer

In this layer data from the users from presentation layer are stored obtained through persistence layer and data is given to the layer above when requested *(Richards 2015)*. XSLT, XML files, Database tables are components of this layer.

There are three possible layers in Java EE. They are:

1. Presentation Layer

This layer works like the .NET but has different components. Its components are JSP-create dynamic web application, Servlets-get request from JSP and return the service, JavaBeans-performs encapsulation of objects, Swing-creates windows UI.

1. Business Layer

This has also same working mechanism to .NET. Its components are Entity Beans-data object representing real life object where session bean handle actions (*Enterprise JavaBeans (EJBs) (WebLogic Server Components)* 2021), Message Driven Beans-process messages asynchronously (*What Is a Message-Driven Bean? - The Java EE 6 Tutorial* 2021), Session Beans-business process objects that perform actions (*Enterprise JavaBeans (EJBs) (WebLogic Server Components)* 2021).

1. Persistence Layer

The components of this layer are Java Persistence API-handles java object that is mapped with relational data and stored and be accessible in future (*The Java Persistence API - A Simpler Programming Model for Entity Persistence* 2021), JPA Entity Manager-interface to interact with persistence context (*EntityManager (Java(TM) EE 7 Specification APIs)* 2021).

# Service Orientation

## Infrastructure as a Service (IaaS)

Here, this service is for only to manage low level resources and the user has control over what is running in this infrastructure facility. Amazon Web Service is one best example of this.

## Platform as a Service (PaaS)

This is where faster deployment and development platforms are provided without any concerns on either hardware or software requirements or both. Here the required APIs and services can be used. End product is reliable and developed quickly. Microsoft Azure is one example of this.

## Software as a Service (SaaS)

Rather than developing and maintaining every application, there is a use of software service operated through internet. SaaS provided software might not be running on the cloud. Google Mail, Slack are two examples (IaaS, PaaS and Saas – All Cloud, All different - Java Enterprise Performance | Dynatrace 2021).

## Which service provides Java EE development and runtime environment?

Platform as a service is the service provided by Java EE development and runtime environment. Java EE is developed on top of Java Standard Edition (SE). Large scale, reliable, scalable, secure network applications can be developed (*Differences between Java EE and Java SE - Your First Cup: An Introduction to the Java EE Platform* 2021).

# Runtime Framework

GlassFish is a server provided by Java EE as a service engine, runtime component that let Java EE web services communicate with JBI components.

## Transaction

It is a sequence of discreet actions in an application that must be successfully completed. This ensures data integrity and consistency and if these actions are not completed then the changes are turned back (*About Transactions - Oracle GlassFish Server 3.1 Administration Guide* 2021).

## Concurrent Control

There are concurrent resources that provides concurrency to Java EE applications. Concurrent resources are configured and made available to use by application components like servlets and EJBs in GlassFish server (*Administering Concurrent Resources* 2021).

## Security

GlassFish Server provides security like message security, single sign-on across all GlassFish Server applications, and programmatic login (*GlassFish Server Specific Security Features (Oracle GlassFish Server 3.0.1 Application Development Guide)* 2021).

## Component Packaging and Deployment

Component packaging and deployment is a method of integrating discrete components of an application to a single component and deploying to a known application server (*Overview of Assembly and Deployment (Sun GlassFish Enterprise Server 2.1 Application Deployment Guide)* 2021).

## Benefit of the above:

Advantages of above provided feature of GlassFish Server Runtime Framework is that the application developed makes sure the functions are reliable to execute and concurrently gets executed and resources made available with a security that can handles applications build with GlassFish making sure all application are packaged and deployed correctly.

# Pros, Cons and Misconceptions

## Open-Source VS Proprietary Software

Java EE application can work on any operating system but .NET only on limited version of Windows (Marinina 2017).

|  |  |
| --- | --- |
| Open-Source | Proprietary Software |
| 1. Source code of an application is available on the internet and anyone can modify it for any changes freely. | 1. Source code is available only with the parent company not to the public. |
| 1. Bug fixing are faster due to the available huge community. | b. Bug fixing is on vendors and can be fast or slow (*Difference between Open source Software and Proprietary Software - GeeksforGeeks* 2021). |

Misconceptions on these two is open-source are not enterprise-grade example Google’s Android (LAGUNA 2021).

## Software Portability

It is making software available for all possible environment without losing all features and obtaining same results (Eaton 2012) So, it seems Java EE applications are more portable.

## Misconceptions and clarification about cloud computing

Productivity: If the application is kept on the cloud, then all its supporting elements are set on the cloud and if cloud gets interrupted, productivity will be crushed (Clark 2015). If every component is on the internet and there is not a technology/tools to control onsite and when the cloud provider loose internet, productivity is highly affected.

# Conclusion

## Suitability

After doing above research, enterprise computing can be useful in both cloud and on-site. It depends on the type of business, but cloud is preferred for small business.

## Potential Impact

Business on both cloud and onsite are preferable. Cloud handles the request and services whereas web-based application handles the input data and displays output. This can save huge amount on infrastructure and some time staffs too mainly for small business.

# References

*About Transactions - Oracle GlassFish Server 3.1 Administration Guide* 2021. viewed 14 April 2021, https://docs.oracle.com/cd/E18930\_01/html/821-2416/ablso.html#:~:text=In%20the%20GlassFish%20Server%20environment%2C%20the%20transactional%20user,for%20enterprise%20beans%2C%20or%20explicit%20programmatic%20transaction%20demarcation..

*Administering Concurrent Resources* 2021. viewed 14 April 2021, <https://glassfish.org/docs/5.1.0/administration-guide/concurrent.html>.

Crawford, W & Farley, J 2005, *Java Enterprise in a Nutshell*, 3rd edn, O'Reilly Media, Inc.

Clark, A 2015, *10 Big Misconceptions About Cloud Computing*. viewed 14 April 2021, https://www.entrepreneur.com/article/244042.

*Differences between Java EE and Java SE - Your First Cup: An Introduction to the Java EE Platform* 2021. viewed 11 April 2021, <https://docs.oracle.com/javaee/6/firstcup/doc/gkhoy.html#:~:text=The%20Java%20EE%20platform%20is,reliable%2C%20and%20secure%20network%20applications>..

*Difference between Open source Software and Proprietary Software - GeeksforGeeks* 2021. viewed 14 April 2021, <https://www.geeksforgeeks.org/difference-between-open-source-software-and-proprietary-software/>.

Eaton, C 2012, *Software Portability - an overview | ScienceDirect Topics*. viewed 14 April 2021, https://www.sciencedirect.com/topics/computer-science/software-portability.

*Enterprise JavaBeans (EJBs) (WebLogic Server Components)* 2021. viewed 14 April 2021, <https://docs.oracle.com/cd/E19509-01/820-5892/ref_ejb/index.html>.

*EntityManager (Java(TM) EE 7 Specification APIs)* 2021. viewed 14 April 2021, <https://docs.oracle.com/javaee/7/api/javax/persistence/EntityManager.html>.

*GlassFish Server Specific Security Features (Oracle GlassFish Server 3.0.1 Application Development Guide)* 2021. viewed 14 April 2021, <https://docs.oracle.com/cd/E19798-01/821-1752/beabi/index.html>.

LAGUNA, R 2021, *4 Myths About Open Source We Should Put to Rest*. viewed 14 April 2021, <https://www.wired.com/insights/2013/03/4-myths-about-open-source-to-put-to-rest/>.

Marinina, M 2017, *Java vs .NET: Factors to Consider*. viewed 14 April 2021, <https://codeburst.io/java-vs-net-factors-to-consider-cde1d22b06a7>.

*Overview of Assembly and Deployment (Sun GlassFish Enterprise Server 2.1 Application Deployment Guide)* 2021. viewed 14 April 2021, <https://docs.oracle.com/cd/E19316-01/820-4337/6nfqe0rj7/index.html>.

Richards, M 2015, *Software architecture patterns*, O'Reilly Media, Inc.

*The Java Persistence API - A Simpler Programming Model for Entity Persistence* 2021. viewed 14 April 2021, <https://www.oracle.com/technical-resources/articles/java/jpa.html#:~:text=The%20Java%20Persistence%20API%20deals%20with%20the%20way,even%20after%20the%20application%20that%20uses%20it%20ends>..

*What Is a Message-Driven Bean? - The Java EE 6 Tutorial* 2021. viewed 14 April 2021, <https://docs.oracle.com/javaee/6/tutorial/doc/gipko.html>.