

**Enterprise Computing Architecture – Java, NET Beans,
and Cloud Computing**

CONFIDENTIAL

Table of Content

1. Introduction.....	1
1.1 Enterprise computing and available technologies.....	1
1.2 Report structure.....	1
2. Layered model.....	1
2.1 Benefits of a layered design.....	2
2.2 Components of each layer and its functions.....	2
3. Service Orientation.....	4
3.1 IaaS.....	5
3.2 PaaS.....	5
3.3 SaaS.....	6
3.4 Which service provides Java EE development and runtime environment.....	6
4. Runtime Framework.....	6
4.1 Transaction.....	7
4.2 Concurrency control.....	7
4.3 Security.....	7
4.4 Component packaging and deployment.....	7
4.5 Benefits of the runtime framework.....	8
5. Pros, Cons, and misconceptions.....	8
5.1 Open source vs proprietary software.....	8
5.2 Software portability.....	9
5.3 Misconceptions and clarification about cloud computing.....	9
6. Conclusion.....	10
6.1 Suitability.....	10
6.2 Potential impact.....	10
References.....	11

Table of Figures

Figure 1- Layered architecture.....2
Figure 2- Java EE environment.....3
Figure 3 - .NET environment.....4
Figure 4- SOA of Cloud Computing.....5

CONFIDENTIAL

1. Introduction

1.1 Enterprise computing and available technologies

An enterprise refers to the business. It further means "enterprise computing" or business logic for supporting business operations. Technically, enterprise computing describes a set of information technology tools, software, and business organizations' techniques. It is used to implement business functions, execute production operations, and other back-office support. Different categories of integrated IT systems and hardware tools help enterprises perform numerous functions such as managing invoices, marketing, customer relationships, and so forth (Bestcomputersciencedegrees.com, 2019). The competitions are increasing complexity in businesses that need IT systems that are leveraged with the latest technologies. Dot net [.NET] framework, Java Enterprise Edition or Java EE, Cloud Computing, etc. are common technologies for building enterprise software. These technologies have features which are reducing the complexity of enterprise operations.

1.2 Report structure

The report's major aim is to explore the concepts of enterprise computing and available technologies building an enterprise architecture. The report is discussing three technologies that include Java EE, .NET, and Cloud Computing. Each technology is following the layered architecture. The report would discuss this architecture, its advantages, and the most popular technology that is, "Cloud Computing," and its services. The report would also discuss the runtime framework for building enterprise applications and how it provides benefits to enterprises. The advantages and disadvantages of the three technologies would also differ in the report. A conclusion would be drawn based on discussed concepts related to enterprise architecture.

2. Layered model

A business model or architecture that uses multiple layers to design, develop, and make an application is called a layered architecture or layer model. This model is used as a standard model for the implementation of complex applications to support enterprises further. The classic layered model consists of four layers, including "presentation", "application", "domain", and "infrastructure" like Figure 1 (Richards, 2015). Various objects reside on each layer for implementing a specific function.

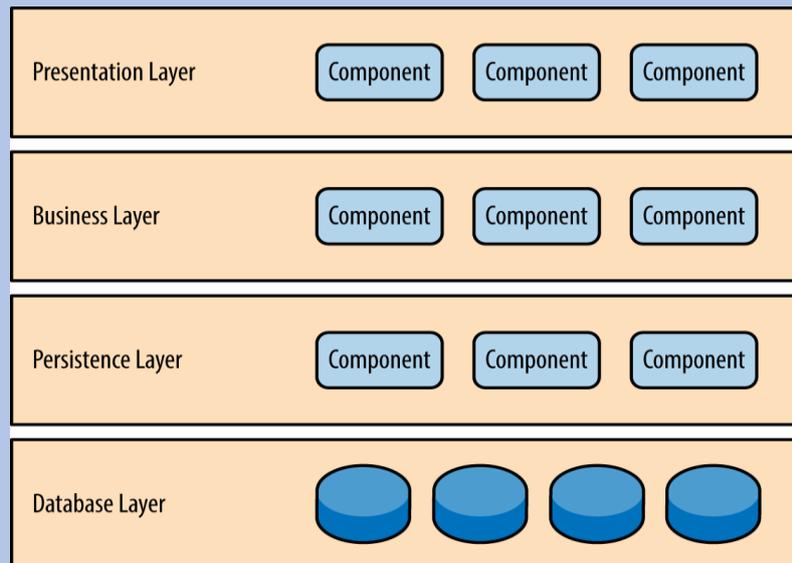


Figure 1- Layered architecture

2.1 Benefits of a layered design

Enterprises are getting benefits through the implementation of the layered design. Some major benefits are as follow:

- The layered design gives developers the freedom to use the latest technologies and update each layer's technology stack without changing in other layers of the application (Izenda, 2017).
- Modularity, portability, scalability, simplicity, flexibility, and stability are layered design features to develops a better enterprise application (Izenda, 2017).
- Multiple development teams can work independently according to their expertise area. Working on a single area makes robust application instead of working on a complete stack of architecture by one team (Izenda, 2017).
- Scalability permits developers to implement the different categories of databases and technologies instead of using one particular technology. In this way, an enterprise can scaleup multiple web servers.

2.2 Components of each layer and its functions

Whether it is Java EE, .NET, or others, each technology framework follows some kind of layered architecture. The framework contains components, and each component performs a separate function. The brief explanation of each framework is as follow:

Java EE layered architecture

Elements of Java EE layered architecture are J2EE application parts that are installed on the different client machines to respective application tiers in the multitiered environment. The "Client tier", "Web tier", "Business tier", and "Enterprise Information System [EIS] tier" are tiers of architecture, which is also in Figure 2 (Gerardnico.com, 2018).

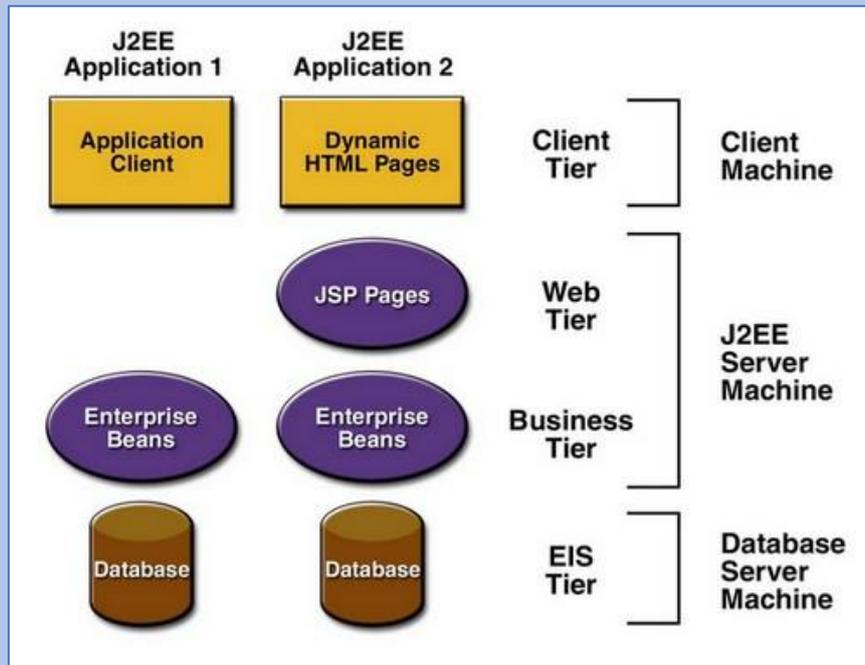


Figure 2- Java EE environment

- **The client tier** includes application clients [web browsers, standalone applications that access server, and located separately from the server. The main function of the clients is to make requests to the server. The server has the responsibility to complete the request process and return a response (Goncalves, 2009).
- **Web tier** includes components [Java applets, Java Server Pages [JSP], etc.] that establish interaction between the client and the business tier. The dynamic generation of content in the different formats for various clients, controlling pages, collecting input from the users, and maintaining the static data for a session are the components' common functions (Docs.oracle.com, 2018).
- **The business tier** includes components that support business logic for the required application. Business logic refers to codes that add functionality to a particular business domain (Gerardnico.com, 2018).
- **EIS tier** includes components like database systems, Enterprise Resource Planning [ERP] systems, mainframe systems, and other legacy systems (Gerardnico.com, 2018).

It stores and maintains the databases of the enterprises by making a database connection.

.NET framework layered architecture

The layered architecture of .NET consists of "presentation", "business", and "data" layer, which is also shown in Figure 3 (Patel, 2014).

- **Presentation layer** – This layer performs user interaction between the users and applications. Web forms and User Controls are the basic components of the layer representing the required information to the users (Patel, 2014).
- **Business layer** – It works to bridge between the data layer and the presentation layer. It consists of the business logic layer, value object, data access layer, and other components used to add business logic (Patel, 2014). All kinds of data are passed via this layer before presenting it to the presentation layer.
- **Data layer** – The data layer stores and maintain all types of application's data. XML files, database tables, and other components that store application data are common.

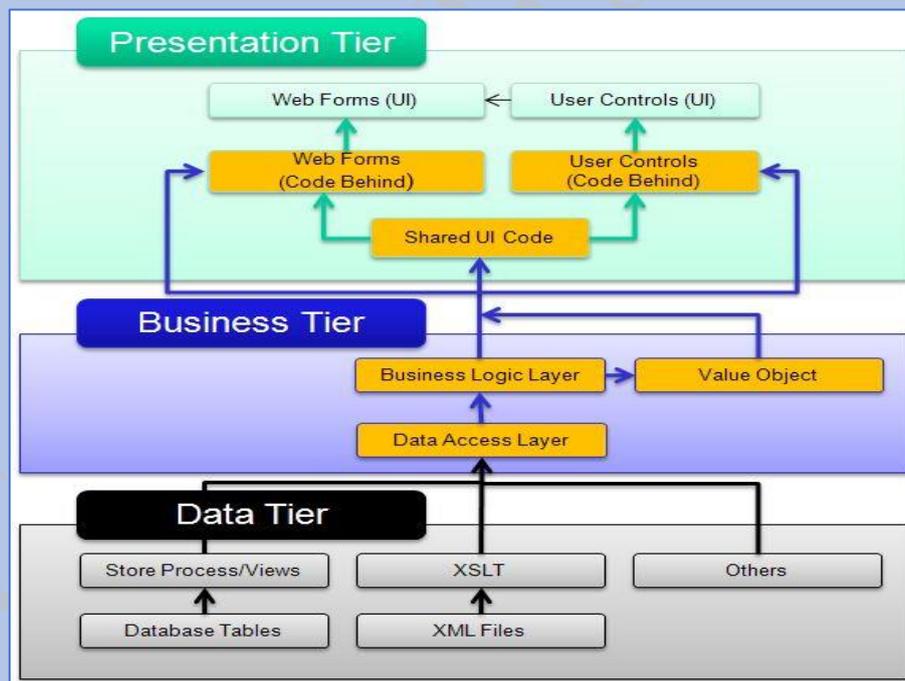


Figure 3 - .NET environment

3. Service Orientation

An evaluation of distributed computing based on request/reply design concepts for synchronous and asynchronous applications is called "Service- Oriented Architecture" or SOA. An application's business logic or other individual functions is modularised into modules and

presented in services for client applications. Cloud Computing also followed the SOA. It provides various forms of assistance, including Infrastructure as a Service [IaaS], Platform as a Service [PaaS], and Software as a Service [SaaS], which is shown in Figure 4 (Dhillon, 2015).

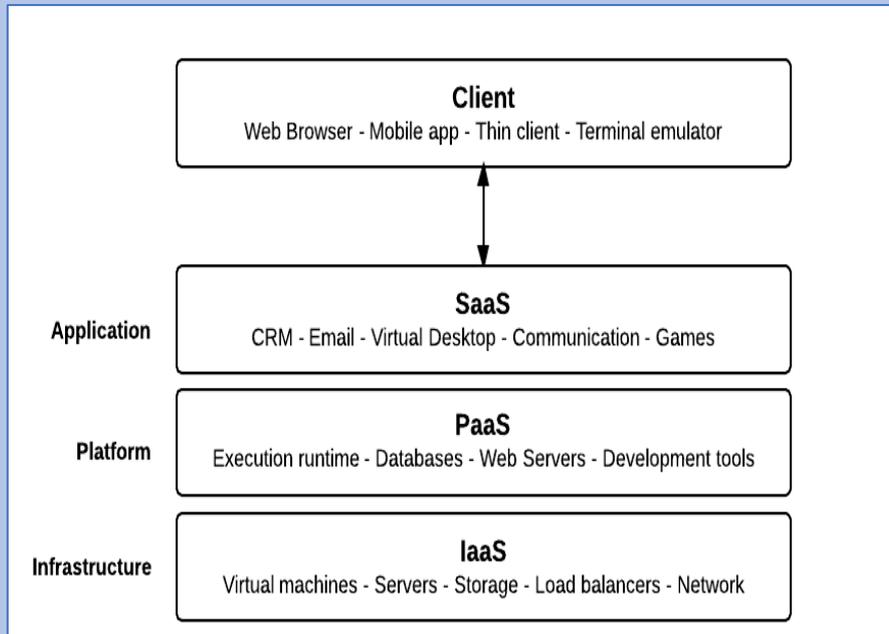


Figure 4- SOA of Cloud Computing

3.1 IaaS

A cloud computing service that provides virtualized hardware computing resources across the internet refers to the IaaS. A third-party service provider offers hosting services means hosts required hardware, software, storage, services, and other infrastructure components to the users. Hosing applications, system maintenance backup, resiliency planning, dynamic scaling, automation of administrative tasks, desktop virtualization, etc. are other services of IaaS (Rittinghouse and Ransome, 2016). IaaS offers scalable resources which could be adjusted according to demand and workload. Due to the low barrier of entry for providing this service, many companies are offering IaaS. Amazon Web Services, Joyent, Rackspace cloud, etc. are top IaaS providers.

3.2 PaaS

A form of cloud computing service which delivers applications across the internet is called PaaS. The cloud service providers provide hardware and software which are required for application development on their Infrastructure. It reduces the overhead of the process of installation of in-house resources for developing an application. Scalability, security, redundancy, auto-provisioning, logging, reporting, integration with other infrastructure

components, etc. are the PaaS's major features (Rittinghouse and Ransome, 2016). PaaS mostly focuses on the protection of data that are stored on their Infrastructure. Users only need to pay for the taken services from the host providers. Google Cloud, Windows Azure, and force.com are top PaaS providers.

3.3 SaaS

SaaS is a kind of software distribution model which hosts various applications. The cloud service provides host applications to clients over a network or internet. It is related to on-demand computing software delivery models and Application Service Providers that offer on-demand applications and charges according to use. The SaaS focuses on managing application access. For example, a client took a CRM application service from the cloud service provider. After taking the service, clients can access and download detailed information. The easy administration, collaboration, global accessibility, compatibility, and automatic updates are SaaS features. Salesforce.com, Google Apps, Zoho, etc. are the top PaaS providers.

3.4 Which service provides Java EE development and runtime environment

PaaS cloud service is a service that supports runtime environment and Java EE development. At present, different companies like Amazon Web Service [AWS], Microsoft Azure, Google App Engine, IBM Cloud Platform, Oracle Cloud Platform, etc. are providing PaaS services. It hosts all kinds of hardware and software across the internet by providing a runtime environment that efficiently develops applications.

4. Runtime Framework

A runtime framework refers to a framework that provides an environment for building and executing applications. This environment includes executable variables, executable files, libraries, machine files, etc. and other required things to implement processes. It is also called an application server that offers facilities for creating web applications and a server environment for running it. Apache Geronimo, Oracle GlassFish, Apache TomEE, JBoss WildFly, etc. are a Java application server (Sapkota, 2013). Sun Microsystem started the open-source application server that is "GlassFish" for the Java EE platform. The transaction, concurrency control, security, and component packaging are features of this server described in the following subsections.

4.1 Transaction

Generally, a transaction refers to the management of the operation. Concerning the database, the transaction of the database manages the database operations within a database system. The application server or framework has the responsibility to provide the surety of consistent database operations. In this way, it prevents databases from going into inconsistent states. For example, the GlassFish server provides Java Database Connectivity [JDBC] drivers to implement the transaction operations. It also offers a feature, "Message Queue" ensuring the sender's messages are received by receivers (Goncalves, 2009). The framework also provides facilities for database recovery.

4.2 Concurrency control

When two or more processes, programs, functions, or methods are executed simultaneously, this state is called concurrency. For any kind of application or database, it is possible that more than one user accessing the same database or requesting the same pages from the server. If the concurrency of applications would not be controlled, then it may cause issues. Due to this, the runtime framework provides features for managing the concurrency (Goncalves, 2009). For example. Managed Executor Service is used as a concurrency control utility in a Java runtime environment. It monitors all processes to avoid concurrency issues.

4.3 Security

Any kind of framework or server requires a hundred percent security to the applications built on it. Without security features, no framework would be useful. Thus, the framework defines security features for creating a secured application. For example. GlassFish defines security mechanisms and tokens, verifying and validating the enterprise's applications developed in the Java EE platform. It also explains a security feature which authenticates the web services (Goncalves, 2009). It is also used for transmitting security information via a security message from one sender to another receiver across the internet.

4.4 Component packaging and deployment

Any kind of framework includes various components such as server files, server instances, administrative domains, message queue, etc. for deploying and packaging applications. These components are packaged flexibly. For example, GlassFish is specially designed for the Java EE platform. It contains all these kinds of parts for creating robust enterprise applications. During the development process, each part has its responsibility to do separate functions

(Goncalves, 2009). It ensures that the previous version of the application would be the same in the next versions.

-4.5 Benefits of the runtime framework

Application server's framework provides an environment where different applications could run no matter what those applications do or what they are. Here are some benefits (Sapkota, 2013):

- It provides an integrated environment where developers could integrate data and code in one place. It supports a centralized methodology for developing applications and updates to it.
- It provides enhanced security through the centralization of data access. The authentication process is also implemented to increase security.
- It limits the network traffic, which automatically improves the performance during the heavy use of the application. It also results in a lower cost of ownership.
- It is easy to install the application in the environment without investing a huge amount. It provides effective memory management, centralization, language interoperability, and reduced cost of development.

5. Pros, Cons, and misconceptions

Each technology, software, hardware has its advantages and disadvantages. Java EE and .NET frameworks are open source frameworks, and anyone can download it from the internet. Open-Source framework or software permits computer codes to be freely available, sharable, and modifiable by others. It is distributed under a license agreement, and users can change the actual design. It means developers can change the Java EE and .NET framework according to the applications. Proprietary software is just the opposite of open source and requires permission for accessing it (Singh, Bansal, and Jha, 2015). The source code is not shared with the public, and no one can make changes.

5.1 Open source vs proprietary software

There are some pros and cons of each kind of software. Three factors [cost, service, usability] are used for describing the pros and cons.

- **Cost:** The pros of open source are the cost as users do not need to pay any amount for using it. The cost of proprietary software depends on the complexity of the system. Users need to pay a small amount for using the standard version, but the cost increases

when the system gets customized (Saltis, 2018). This customization increases the security, functionality, and innovation level, which are the pros of proprietary software.

- **Service:** Open-source software fails for delivering a high level of services due to a lack of experts, but reliable services and support are the greatest advantages of proprietary software. The proprietary software community offers skilled technical staff and other ongoing support to end-users (Saltis, 2018).
- **Usability:** Usability is criticized for open-source as technologies are not updated regularly by the experts or developers. For proprietary software, usability is another pro or high selling point (Saltis, 2018). It is because users get various resources like user manuals for maximizing the use of the software.

5.2 Software portability

Software portability refers to the usability of the same software in a different environment. For example, developers use a single runtime environment to develop various applications using other programming languages. It provides a benefit that developers could do programming in any language and run it in a different climate as per need. This feature reduces the installation process's overhead, increases the flexibility of application development, etc. of the complex systems (Dan, 2012). Software portability has a disadvantage in that there is a chance of security issues. When developers transfer files from one computer to another computer, there is a possibility of virus transfer.

5.3 Misconceptions and clarification about cloud computing

For many enterprises, the transition to cloud computing is the right decision. However, developers are making mistakes due to surrounded myths or misconceptions. It is necessary to understand the benefits of cloud computing without falling towards the myths. Some common myths are as follow (Nordquist, 2018):

- Cloud saves huge money by cutting the upfront expenses. But, it increases the cost of other expenditure items and unable to keep a fortune.
- Cloud computing is not secured as on-premise Infrastructure. The fact is that the cloud environment is quite secured as compared to open internet storage. Cloud service providers are using security features for implementing security within the cloud.
- Every enterprise must have to move to the cloud. But, all enterprises do not need to adopt and implement a cloud environment. Enterprises must have to use the cloud according to their business and needs.

6. Conclusion

Cloud computing, Java EE, and .NET, other technologies are discussed in the report. The report found that these technologies are popular for developing enterprise applications. Here are two important conclusions have been drawn based on the analysis of these technologies.

6.1 Suitability

It is a fact that "cloud computing" is one of the latest technology for application development and other operations. The report concludes that adopting this technology would be a good decision for complex business operations. Enterprise can get several cloud service providers that could support the implementation of the cloud environment.

6.2 Potential impact

Implementation of a cloud environment has a greater impact on business. It would help the organization to goes high into the current business market. It increases competition for other companies. The cloud environment can bring innovation in business operations, scale-up resources, and bring products for the end-users. Thus, the report concludes that cloud environment implementation would provide benefits to enterprises.

References

- Bestcomputersciencedegrees.com. (2019). *What is Enterprise Computing?*. [online] Available at: <https://www.bestcomputersciencedegrees.com/faq/what-is-enterprise-computing/>[Accessed 6 Apr. 2019].
- Docs.oracle.com. (2018). *Overview of Enterprise Applications*. [online] Available at: <https://docs.oracle.com/javaee/6/firstcup/doc/gcrky.html>[Accessed 6 Apr. 2019].
- Dan (2012). What is a portable application, and why should you use them?. [Blog] Available at: <https://www.rlvision.com/blog/what-is-a-portable-application-and-why-should-you-use-them/>[Accessed 7 Apr. 2019].
- Dhillon, B. (2015). *Different Types of Cloud Computing Service Models | BluePi Blogs*. [online] BluePi Blogging. Available at: <https://www.bluepiit.com/blog/different-types-of-cloud-computing-service-models/>[Accessed 7 Apr. 2019].
- Gerardnico.com. (2018). *Java EE - (J2EE/JEE) Platform [Gerardnico]*. [online] Available at: <https://gerardnico.com/lang/java/j2ee>[Accessed 6 Apr. 2019].
- Goncalves, A., (2009). *Beginning Java EE 6 Platform with GlassFish 3: from novice to professional*. Apress.
- Izenda. (2017). *5 Benefits of a 3-Tier Architecture - Izenda*. [online] Available at: <https://www.izenda.com/5-benefits-3-tier-architecture/>[Accessed 6 Apr. 2019].
- Nordquist, B. (2018). *The 8 Top Myths and Realities of Cloud Computing | StorageCraft*. [online] StorageCraft Technology Corporation. Available at: <https://blog.storagecraft.com/top-myths-realities-cloud-computing/>[Accessed 7 Apr. 2019].
- Patel, P. (2014). *Three Layer Architecture in C# .NET*. [online] Codeproject.com. Available at: <https://www.codeproject.com/Articles/36847/Three-Layer-Architecture-in-C-NET-2>[Accessed 6 Apr. 2019].
- Richards, M. (2015). *Software Architecture Patterns*. 1st ed. United States of America: O'Reilly Media.
- Rittinghouse, J.W., and Ransome, J.F., (2016). *Cloud computing: implementation, management, and security*. CRC press.
- Sapkota, D., (2013). *Back End Programming in .NET Framework*.

Schniederjans, D.G. and Hales, D.N., (2016). Cloud computing and its impact on economic and environmental performance: A transaction cost economics perspective. *Decision Support Systems*, 86, pp.73-82.

Singh, A., Bansal, R.K. and Jha, N., (2015). Open source software vs proprietary software. *International Journal of Computer Applications*, 114(18).

Saltis, S. (2018). Comparing Open Source Software vs Closed Source Software. [Blog] *Content Management*. Available at: <https://www.coredna.com/blogs/comparing-open-closed-source-software> [Accessed 7 Apr. 2019].

CONFIDENTIAL