

**Your text here**

**Assignment -1**

**Topic: Glassfish as Emerging Technology in Application-server Middleware**

**Your text here**

**Ass**

**Sumanth Mohan**

**Student ID: C0752365**

**Abstract:**

**In this survey paper, a latest trend in the Enterprise middleware management domain ins extensively discussed. One of the emerging technologies in Enterprise Middleware is Glassfish. It is a latest technology which is primarily designed for the Enterprise Application Middleware architecture. In this survey paper, the various use cases of Glassfish are discussed and how Glassfish is powering the application middleware is discussed in detail. The Features and the architecture of Glassfish is also discussed. The survey paper includes a scenario-based example which demonstrates the working, integration and deployment of Glassfish in the application middleware architecture.**

**Keywords:** *Enterprise Middleware, Applications, web services, interoperability, Java Enterprise Edition, multi-objective optimization*

1. **Introduction**

With the Invent of webservices, a plethora of information is collected, processed and integrated in different ways. The Information that is collected can be varying data types, sizes and can represent different sets of information. Also, with the emergence of cloud computing a number of applications are being developed for the cloud. This is where the importance of middleware management plays such a vital role. The Middleware is designed keeping in mind a number of design patters to ensure the applications are able to interact with each other **[1]** (reference1, 2010). The web services middleware is usually optimised by throughput by design.

The Enterprise Middleware can be classified into different categories such as the database middleware, application server middleware, web middleware, message-oriented middleware and the transaction processing monitor. Each program normally gives informing administrations with the goal that various applications can convey utilizing informing systems like straightforward item get to convention (SOAP), web administrations, illustrative state move (REST), and JavaScript object documentation (JSON). While all middleware performs correspondence works, the sort an organization decides to utilize will rely upon what administration is being utilized and what kind of data should be imparted. This can incorporate security confirmation, exchange the board, message lines, applications servers, web servers, and indexes.

Glassfish is a latest technology which is primarily designed for the Enterprise Application Middleware architecture. The Glassfish is used in the application middleware architecture, that is glassfish can be used as a web server (Http Server).Glassfish promotes Interportability. Interporability can be defined as the ability of a particular service to work together with a client**[2]** (Reference2, 2017). A web server usually consists of handling the HTTP requests, a servlet container like Tomcat.

1. **Salient Features of Glassfish**

**2(1) Constraints**

The most Important feature of Glassfish is the application server removes the constraints which enables easy Interaction and management among applications. The Glassfish removes both the global and the local constraint in its composition. Software architecture allows early documentation of high-level software design solutions, providing a comprehensive and structural software view **[5]**. It belongs to the post stage of lifting requirements. According to thanks to the architectural specifications, software engineers are better supported to deal with the increasing complexity of software systems today. Therefore, this is one of the most important activities within software development.

**2(2) Lifting Requirements**

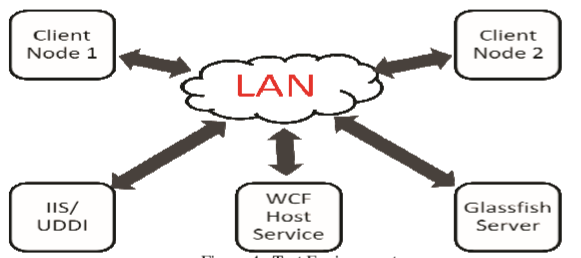
Another Important feature of the Glassfish is the lifting up of requirements. Glassfish works as the application server manager and helps the application to match up to the requirements of the application server. By doing this, interaction among the applications would be a easier choice. The engineering styles decide a lot of configuration decides that distinguish sorts of segments and connectors that can be utilized to construct a framework or programming subsystem with nearby and worldwide imperatives of its structure**[5]** (Reference5, 2016). Different creators, for example, engineering style isn't a product design, yet a procedure that is a piece of the arrangement of solid issues

**2(3) Interportability**

Interoperability can be characterized as the capacity of a specific assistance to cooperate with a customer. Therefore, this is one of the most important activities within software development.The engineering styles decide a lot of configuration decides that distinguish sorts of segments and connectors that can be utilized to construct a framework or programming subsystem with nearby and worldwide imperatives of its structure. In this paper we center around syntactic interoperability, which at last alludes to address correspondence and trading of information between the two frameworks included, and in general alludes to the entire between activity process, as portrayed straightaway

1. **Architecture of Glassfish**

Below is a figure the demonstrates the working of Glassfish Application server. The Glassfish server along with the Client nodes are connected to the LAN which is based out of the cloud. The WCF host service, IIS/UD**[4]** (Reference4, 2012)D1 , WCF host service and other application servers are all Interconnected to form a bus like structure which is demonstrated in the figure below.



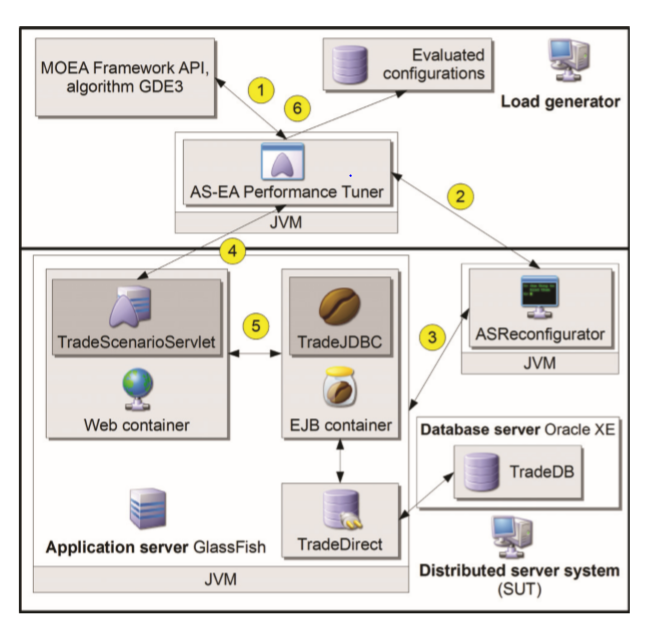
***Figure (a): Glassfish server connected to the LAN***

For web administrations usage, IIS server, UDDI library, application servers (Glassfish 3.1 and Windows Communication Foundation Application Server) and two customer hubs with significant customer applications were arrangement **[4]**.

The customer hubs have customer applications which make a solicitation to web server/UDDI library to get the necessary pages/benefits individually. We facilitated web administrations on the Glassfish Server rendition 3.1 for Java web administrations applications. The site pages conjuring these administrations were additionally made and refreshed on IIS for clients to buy in to. Likewise, we facilitated the Windows Communication Foundation Host Service on a hub where web administrations were facilitated utilizing C#. These web administrations were then distributed on UDDI Registry on a similar hub where IIS web server was begun. The customer applications were utilized to call these administrations made in VB for collaborating with UDDI and speak with the application server facilitating the web administration

**4.Use case to demonstrate the working of Glassfish in a database Environment: Glassfish as emerging trend in the Application server Middleware architecture**

The Glassfish is the database application server which facilitates the Interaction and management of the middleware architecture.



**Figure (b): Glassfish server**

The Middleware architecture consists of the Load Generator, Application server and the Database server. The Glassfish as tool came across as the application server which helps in the middleware architecture. Glassfish also demonstrates the working capabilities among the middleware system architecture. execution tuning results rely upon the equipment and software conﬁguration used in the process**[3]** (reference 3, 2013). We assume that our equipment conﬁguration is able enough of dealing with the given outstanding task at hand and that the main execution bottleneck dwells in the AS itself. For example, many open source Java projects (e.g. JEdit, ArgoUML and Glassﬁsh) assume that their developers should use the Java coding conventions**[6]** (Reference6, 2015). Just a single equipment set is available to us, forestalling equal tuning. Additionally, we limit our methodology here to only each application in turn despite the fact that it tends to be stretched out to numerous applications. Furthermore, the method of sending the solicitations to the AS issues also**[3]**. We have attempted two of them: sending all the solicitations to the as simultaneously as could be expected under the circumstances and artiﬁcially deferring them so as not to over-burden the AS. For the time being just a solitary enhancement run is performed

1. **Conclusion**

In this survey paper, the Glassfish as an emerging tool in the application middleware segment is widely discussed. Glassfish is open source, easily available and also has high interpretability features. This survey paper demonstrates the working example of glassfish and suggests two salient features which are negation of all the constraints that might exist and also lifting all the requirements that exists between the applications.

Thus, the survey paper concludes the Glassfish helps multiple the interfaces and also demonstrates the working of middleware architecture. Figure(a) and Figure(b) demonstrates the working of the middleware architecture and specifically using Glassfish as the application server management tool. The effect of standards on the achievement/disappointment of a product task and building an express standard "database". These subjects would intrigue both sociology analysts and PC researchers.

**References:**

**[1]** Vidura Gamini Abhaya, RMIT University, Melbourne, Australia, Building Web Services Middleware with Predictable Service Execution <https://link.springer.com/chapter/10.1007/978-3-642-17616-6_5>

**[2]** Bruno Martins, University of Coimbra, Portugal, INTENSE: Interoperability Testing as a Service - <https://ieeexplore.ieee.org/document/8029837>

**[3]** Marko Leˇsnik, Faculty of Electrical Engineering and Computer Science, University of Maribor, Performance Tuning of Java EE Application Servers with Multi-Objective Differential Evolution - <https://ieeexplore.ieee.org/document/6601444>

**[4]** Abhijit M Kulkarni, Dept of Computer Engineering, Defence Institute of Advance Technology (DU, Comparative Study of Middleware for C4I Systems - <https://ieeexplore.ieee.org/document/6212685\>

**[5]** Daniel Guamán, Universidad Tecnica Particular de Loja, Ecuador, Implementation of techniques, standards and safety recommendations to prevent XSS and SQL injection attacks in Java EE RESTful applications -<https://link.springer.com/chapter/10.1007/978-3-319-31232-3_65>

**[6]** Hoa Khanh Dam, University of Wollongong, Mining Software Repositories for Social Norms <https://ieeexplore.ieee.org/document/7203029>