Systems Integration Framework

Name

Course

Tutor

Date

Systems Integration framework

**The architecture of the components of the proposed system integration**

*A description of the proposed architecture*

 The integration framework that was adopted for this Google project mixed all the three models of system integration. It has the conceptual model, the logical model, and the physical model. The Conceptual model indicates the relationship that exists among the various concepts. It utilizes arrows to show the existence of relationships between the different entities that form part of the framework. On the other hand, the logical model explains the mechanics or the logics that dictate the operations of the system. Finally, the physical model indicates how the interfaces, the protocols, and the data are arranged in the system so as to arrive at the desired system integration.

 The architecture of the mixed model usually involves a significant number of layers. In most cases, the number of the layers is determined by the desired result of the system and the various functions that the system is expected to perform. In that regard, the layers are set up by their duties. For this purpose, the architecture of the project will contain the following channels; the end-user device, the front end user (business process and user interface) the integration layer consisting of the integration services and the back end provider which entail the business logic and the data that would be involved in the activities (Schuldt, 2009).

*Diagrammatic representation of the architecture*

 The following is a diagram below is an image of the multi-tier architectural design that has been adopted to integrate the delivery of services to the customers of Google Inc. The main aim of this integration is to improve the services that the clients of the company enjoy and reduce some of the costs that the institution incurs especially on maintenance.



*The components and interfaces that could be acquired and those that can be generated internally*

 The application server is one of the key elements in this arrangement. It sits strategically between the enterprise server and the end-user. The company does not need to acquire such servers from outside of the facility. Instead, it can use its internal engineers to build new ones for the institution or use the existing ones as long as they are in working conditions. Besides the application servers, the company can also make its enterprise resource data contain that would facilitate the transfer of information to the clients (Lim et al, 2009). From the design, it is necessary to note that it would not be prudent for the company to outsource any interfaces or components since most of them are already in existence in the other models that the company utilizes. They would only need to be reorganized to be in line with the new design.

**Implementation framework**

*The structure that has been used to define and implement the project*

 The structure that has been utilized in the system integration is the multiple-tier distribution process. In earlier computer applications, especially in circumstances that involved client-server technology, there always arose problems that are associated with a low-performance level, demand for frequent maintenance and rigidity. In the multi-tier design, it is possible for Google to serve billions of customers simultaneously while doing away with the inflexibility, slothfulness and the demands for frequent maintenance. The application can execute these functions by staging the databases. Besides, it is in a position to carry out queuing and perfect database execution.

*Definition of the project regarding the framework*

 In earlier research activities, it emerged that a substantial number of clients of Google Inc were not happy with the services of the company. Google provides a platform through which people can access information across the globe. They have databases in which they store the information that they have collected. However, access to these pieces of information affected by slow response and provision of wrong or unwanted information. In that regard, it is necessary to integrate the systems of the institution uses make access to such information very easy.

*Advantages of the selected framework*

 The major benefit of a multiple-tier integration framework is that it guarantees improved performance. Due to the various components that are involved, the structure will make users access information at a faster rate than the current mechanism. Additionally, the system is secure since it uses security applications that will prevent individuals from carrying out malicious actions that may compromise the quality of service that is being offered. Finally, the Applications in this design are relatively easy to develop since due to their simplicity (Moody et al, 2010).

*The integration challenges*

 Some problems were faced while implementing this project. Firstly, the mix of technologies that have been used in the process presented a daunting task. Since the layers are numerous, they need to be executed in such a manner that they do not pose any elements of disagreement. As a result, it requires that the company must be extra vigilant in monitoring and determining any possibility of the system disagreeing. The result is that it took a lot of time to develop the system. Another feature that brought challenges to the system is that the developers appear to have tiny control over the participating applications. That can make the system uncontrollable, and that may lead to undesirable qualities in the final system.

References

Lim, S. H., Sharma, B., Nam, G., Kim, E. K., & Das, C. R. (2009, August). MDCSim: A multi-tier data center simulation, platform. In Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on (pp. 1-9). IEEE.

Moody, A., Bronevetsky, G., Mohror, K., & Supinski, B. R. D. (2010, November). Design, modeling, and evaluation of a scalable multi-level check pointing system. In Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-11). IEEE Computer Society.

Schuldt, H. (2009). Multi-tier architecture. In Encyclopedia of database systems (pp. 1862-1865). Springer US.